

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2012

X. Fu
M. Tao
ZTE
October 24, 2011

Associate PW label with PTP application
draft-fuxh-tictoc-associate-pw-with-ptp-00.txt

Abstract

[1588overMPLS] defines two methods for transporting PTP messages (PDUs) over an MPLS network. The second method is to transport PTP messages inside a PW via Ethernet encapsulation. When PHP is applied to PTP LSP or the PW is established between two routers directly and no PTP LSP is needed, PW label must be associated with PTP application at the PW termination point. This document introduces a mechanism to associate PW label with PTP application.

Requirements Language

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].

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 26, 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. PTP-Aware Capability Advertisement 3
 - 2.1. LDP Extension 3
 - 2.2. BGP Extension 3
- 3. PTP Application Association 4
 - 3.1. LDP Extension 4
 - 3.2. BGP Extension 4
- 4. IANA Considerations 5
- 5. Security Considerations 5
- 6. Acknowledgements 5
- 7. References 5
 - 7.1. Normative References 5
 - 7.2. Informative References 5
- Authors' Addresses 5

1. Introduction

[1588overMPLS] defines two methods for transporting PTP messages (PDUs) over an MPLS network. The second method is to transport PTP messages inside a PW via Ethernet encapsulation. When PHP is applied to PTP LSP or the PW is established between two routers directly and no PTP LSP is needed, PW label must be associated with PTP application at the PW termination point. This document extend LDP and BGP to associate PW label with PTP application.

2. PTP-Aware Capability Advertisement

It is useful for PW switching point to announce its capabilities, such as the capability to be PTP-aware. So both PW switching points could know each other of the PTP-aware capability. If both of them could support PTP-aware, PTP PW label could be coordinated during the label mapping.

2.1. LDP Extension

[RFC5561] defines a mechanism for advertising LDP enhancements at session initialization time. So LDP capability advertisement provides means for an LDP speaker to announce what it can receive and process. This document introduces a new Capability Parameter TLV, the PTP-Aware Capability. Following is the format of the PTP-Aware Capability Parameter.

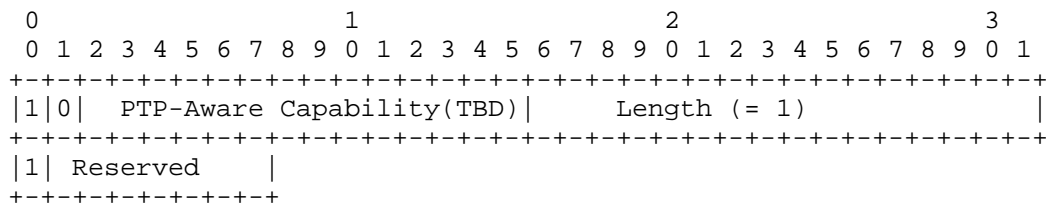


Figure 1: PTP-Aware Capability TLV

The PTP-Aware Capability TLV MUST be supported in the LDP Initialization Message([RFC5561]). Advertisement of the PTP-Aware Capability indicates that the PW switching point supports PTP message processing and PTP application association

2.2. BGP Extension

TBD

3. PTP Application Association

When PTP LSP isn't be present, PW switching point must associate the top label (aka PW Label) with PTP application so that it can identify PTP traffic carried in the PW.

This PTP application association relationship could be configured by management system. It could also be configure by dynamic control plane. This document introduces LDP/BGP extension to signal that this PW segment is a PTP PW.

3.1. LDP Extension

[RFC3036] defines the Label Distribution Protocol (LDP) for distributing labels. This document defines a new TLV, PTP Association TLV which can be used to indicate a PW is associated with PTP traffic. This TLV is carried in the Label Mapping message.

The PTP Association TLV, is defined as follows (TLV type needs to be assigned by IANA):

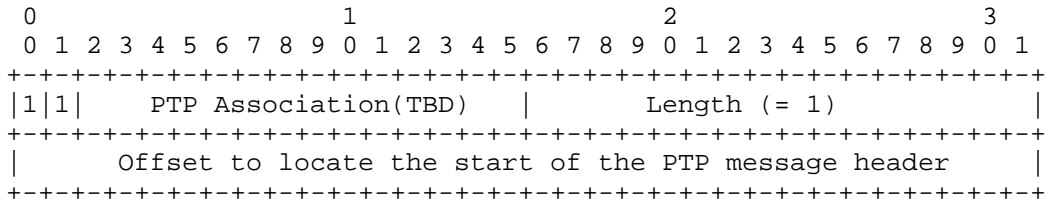


Figure 2: PTP Association TLV

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 T-PE or S-PE must include this object in the LDP Mapping Message when it want to request a PTP label or advertise a PTP label to a peer.

3.2. BGP Extension

TBD

4. IANA Considerations

TBD.

5. Security Considerations

TBD.

6. Acknowledgements

TBD.

7. References

7.1. Normative References

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

7.2. Informative References

[1588overMPLS]
S. Davari, "Transporting PTP messages (1588) over MPLS Networks", draft-ietf-tictoc-1588overmpls-02 .

Authors' Addresses

Xihua Fu
ZTE

Email: fu.xihua@zte.com.cn

Muliu Tao
ZTE

Email: tao.muliu@zte.com.cn

TICTOC Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 9, 2012

S. Davari
A. Oren
Broadcom Corp.
M. Bhatia
P. Roberts
Alcatel-Lucent
L. Montini
Cisco Systems
October 7, 2011

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.

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	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/LEs 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 a 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 minimum 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 require 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.

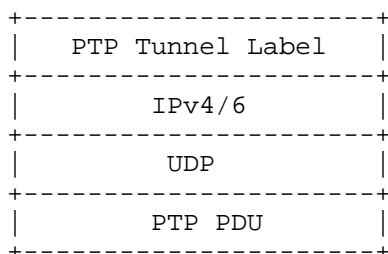


Figure 1 - 1588 over LSP Encapsulation

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.

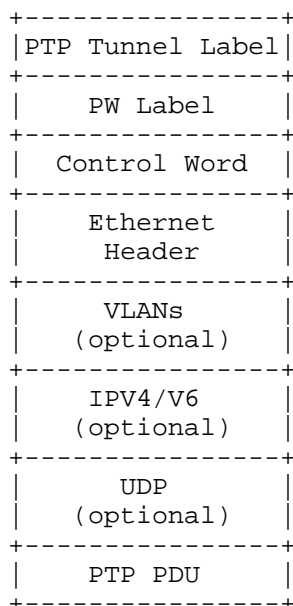


Figure 2 - 1588 over PW Encapsulation

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 in which the PTP LSP label may not be present:

- o When PHP is applied to the PTP LSP, and the packet is received without PTP LSP label at PW termination point .
- o 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:

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

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

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

SYNC and DELAY_REQ are exchanged between 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/LEs 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 1588aware 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. 1588aware 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:

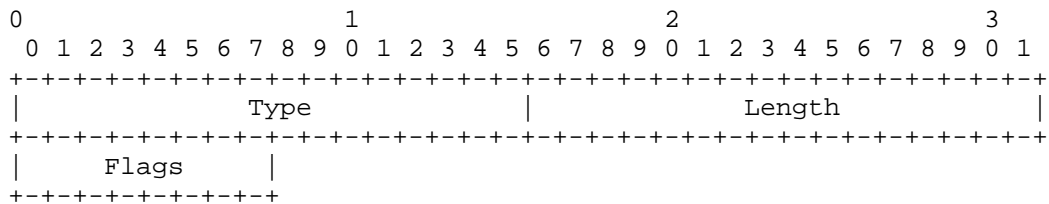


Figure 3: 1588-aware Capability TLV

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.

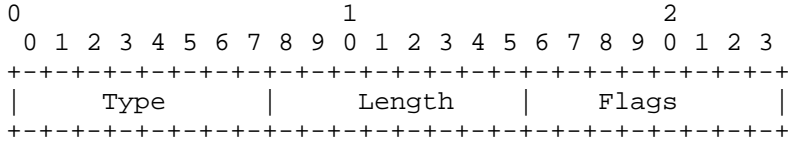


Figure 5: 1588-aware Capability sub-TLV

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.

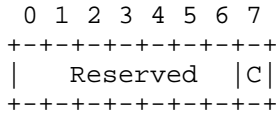


Figure 6: Flags Format

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:

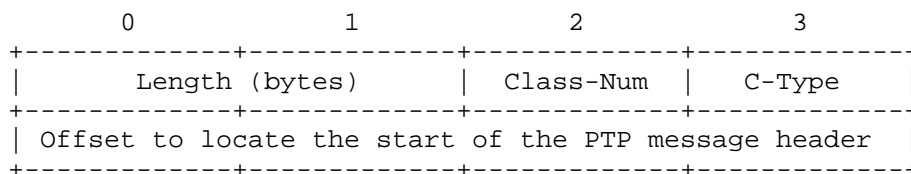


Figure 7: RSVP 1588_PTP_LSP object

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.

Value	Sub-TLV	References
TBD	1588aware node sub-TLV	(this document)

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.

Value	Sub-TLV	References
TBD	1588aware node sub-TLV	(this document)

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 11bbbbbb

Suggested value TBD

Defined CType: 1 (1588 PTP LSP)

21. References

21.1. Normative References

- [IEEE] IEEE 1588-2008, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.
- [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.
- [RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.
- [RFC4448] Martini, L., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, April 2006.
- [RFC4720] Malis, A., Allan, D., and N. Del Regno, "Pseudowire Emulation Edge-to-Edge (PWE3) Frame Check Sequence Retention", RFC 4720, November 2006.
- [RFC5085] Nadeau, T. and C. Pignataro, "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

21.2. Informative References

- [I-D.ietf-pwe3-fat-pw] Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow Aware Transport of Pseudowires over an MPLS Packet Switched Network", draft-ietf-pwe3-fat-pw-07 (work in progress), July 2011.

- [ISO] ISO/IEC 10589:1992, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)".
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC3784] Smit, H. and T. Li, "Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)", RFC 3784, June 2004.
- [RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.
- [RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, September 2008.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

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

TICTOC Working Group
INTERNET DRAFT
Intended status: Standards Track

Vinay Shankarkumar
Laurent Montini
Cisco Systems

Tim Frost
Greg Dowd
Symmetricom

Expires: January 4, 2012

July 4, 2011

Precision Time Protocol Version 2 (PTPv2)
Management Information Base
draft-ietf-tictoc-ntp-mib-00.txt

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-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 January 4, 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.

Abstract

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

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

Table of Contents

1. Introduction.....	2
1.1. Change Log.....	2
2. The SNMP Management Framework.....	3
3. Overview.....	4
4. IETF PTP MIB Definition.....	4
5. Security Considerations.....	64
6. IANA Considerations.....	64
7. References.....	64
7.1. Normative References.....	64
7.2. Informative References.....	64
8. Acknowledgements.....	66
9. Author's Addresses.....	66

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. It is envisioned this MIB will complement other managed objects defined to monitor, measure the performance of the PTP devices and telecom clocks. Those objects are considered out of scope for the current draft.

1.1. 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-ntp-mib

-00 Mar 11 Initial version; showed structure of MIB

draft-ietf-tictoc-ntp-mib

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

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in [RFC 3411].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16 [RFC 1155], STD16 [RFC 1212] and [RFC 1215]. The second version, called SMIV2, is described in STD 58: [RFC 2578], [RFC 2579] and [RFC 2580]
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15 [RFC 1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in [RFC 1901] and [RFC 1906]. The third version of the message protocol is called SNMPv3 and described in STD62: [RFC 3417], [RFC 3412] and [RFC 3414].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15 [RFC 1157]. A second set of protocol operations and associated PDU formats is described in STD 62 [RFC 3416].
- o A set of fundamental applications described in STD 62 [RFC 3413] and the view-based access control mechanism described in STD 62 [RFC 3415].

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

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (e.g., use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3. Overview

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

4. IETF PTP MIB Definition

```
IETF-PTP-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Integer32,
    Gauge32,
    Unsigned32,
    Counter32,
    Counter64,
    transmission
        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;
```

```
ietfPtpMIB MODULE-IDENTITY
```

```
    LAST-UPDATED      "201105060000Z"
    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 PTPv2, IEEE Std. 1588(TM) - 2008

Overview of PTPv2 (IEEE Std. 1588(TM) - 2008)

[IEEE Std. 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 standard IEEE Std. 1588(TM) - 2002 and 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 Std. 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 Std. 1588-2008] also defines conformance and management capability.

MIB description

This MIB is to support the Precision Time Protocol version 2 (PTPv2, hereafter designated as PTP) features of network element system devices.

Acronyms:

ARB	arbitrary
BMC	Best Master Clock
CAN	Controller Area Network
CP	Communication Profile [according to IEC 61784-1:200710]
CPF	Communication Profile Family [according to IEC 61784-1:2007]
DS	Differentiated Service
E2E	End-to-End
E2ETC	End-to-End Transparent Clock
EUI	Extended Unique Identifier.
FFO	Fractional Frequency Offset
GPS	Global Positioning System
IANA	Internet Assigned Numbers Authority
ICV	Integrity Check Value
ID	Identification
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
JD	Julian Date
JDN	Julian Day Number
MAC	Media Access Control (according to [IEEE Std 802.3-2008])
MJD	Modified Julian Day
NIST	National Institute of Standards and Technology (see http://www.nist.gov)
NTP	Network Time Protocol, see IETF [RFC 5905]
OUI	Organizational Unique Identifier (allocated by the IEEE)
P2P	Peer-to-Peer
P2PTC	Peer-To-Peer Transparent Clock
PHY	physical layer (according to [IEEE Std 802.3-2008])
POSIX	Portable Operating System Interface (see ISO/IEC 9945:2003)
PPS	Pulse per Second
PTP	Precision Time Protocol
SA	Security Associations
SNTTP	Simple Network Time Protocol
SOF	Start of Frame
TAI	International Atomic Time
TC	Traffic Class
TC	Transparent Clock
TLV	Type, Length, Value (according to [IEEE Std 802.1AB-2009])
ToD	Time of Day Synchronization
ToS	Type of Service
UCMM	UnConnect Message Manager

UDP/IP User Datagram Protocol
UTC Coordinated Universal Time

References:

[IEEE Std. 1588-2008] Precision clock synchronization protocol for networked measurement and control systems - IEC 61588
IEEE 1588(tm) Edition 2.0 2009-02

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 the boundary clock in that the clock roles can change.

As defined in [IEEE Std. 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.

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.

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-path support:

Indicates that each node in the synchronization chain from master to slave can support [IEEE Std. 1588-2008].

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 Global Positioning System, NTP, and National Institute of Standards and Technology (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 Std 1588-2008/IEC 61588:2009, or different Precision Time Protocol profiles.

transparent clock:

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

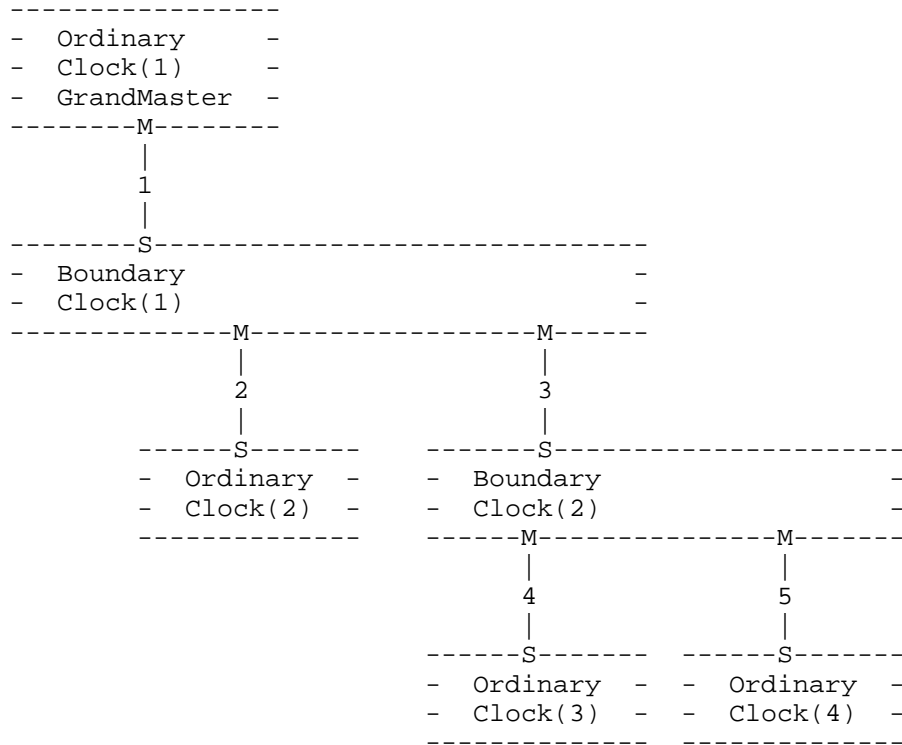
Two-step clock:

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

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

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

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



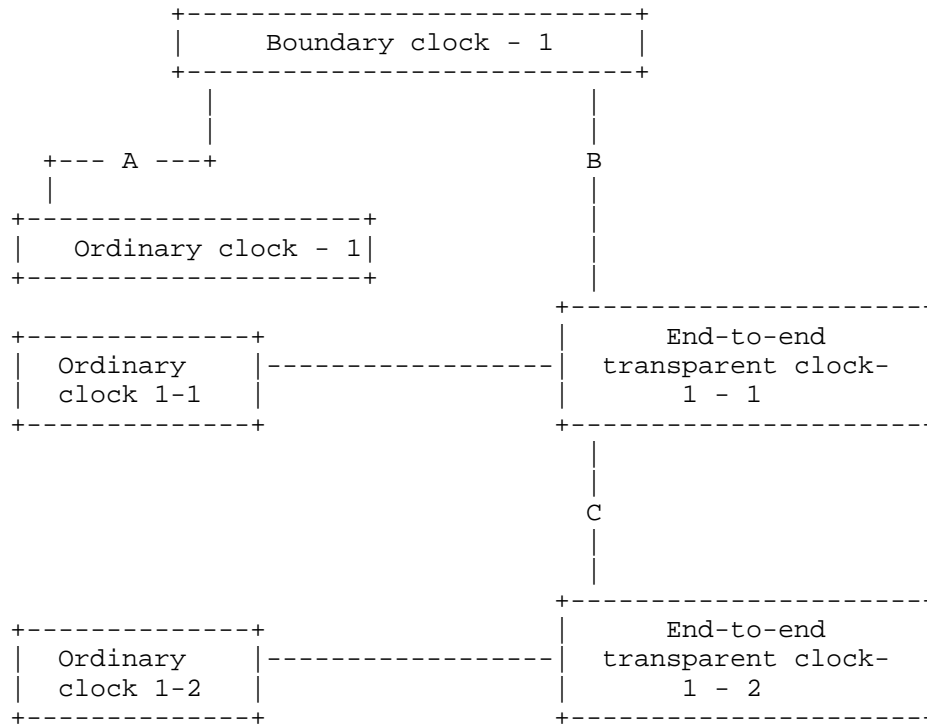
Grandmaster

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

Relationship cardinality

PTP system 1 : N PTP Clock
 PTP Clock 1 : 1 Domain
 PTP Clock 1 : N PTP Ports
 PTP Port N : N Physical Port (interface in IF-MIB)

Transparent clock diagram from section 6.7.1.3 of [IEEE Std. 1588-2008]



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

-- revision log

```

REVISION "201105060000Z" -- 5 May 2011
DESCRIPTION
  "Initial Version"
  ::= { transmission 95 }
  
```

```

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. [IEEE Std. 1588-2008] Section 7.1,
    Domains Table 2
  
```

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

REFERENCE "Section 7.1 Domains and Table 2 of
[IEEE Std. 1588-2008]"

SYNTAX Unsigned32 (0..255)

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 Std. 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 of [IEEE Std. 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 from [IEEE Std. 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 Std. 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 Std. 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 Std. 1588-2008].

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

REFERENCE "Sections 8.2.5.4.4, 6.6.4 and 7.4.2 of [IEEE Std. 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 "Section 7.5.2.3 Port Number and 5.3.5 of
[IEEE Std. 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 Std. 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. From [IEEE Std. 1588-2008] section 3.1.30, 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 Std. 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 Std. 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 [RFC 2578] 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 Std. 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 Std. 1588-2008].

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

- to an application-specific source of time. The timescale distributed shall be ARB. A clockClass 13 clock shall not be a slave to another clock in the domain.
- 14 Shall designate a clock that has previously been designated as clockClass 13 but that has lost the ability to synchronize to an application-specific source of time and is in holdover mode and within holdover specifications. The timescale distributed shall be ARB. A clockClass 14 clock shall not be a slave to another clock in the domain.
- 15-51 Reserved.
- 52 Degradation alternative A for a clock of clockClass 7 that is not within holdover specification. A clock of clockClass 52 shall not be a slave to another clock in the domain.
- 53-57 Reserved.
- 58 Degradation alternative A for a clock of clockClass 14 that is not within holdover specification. A clock of clockClass 58 shall not be a slave to another clock in the domain.
- 59-67 Reserved.
- 68-122 For use by alternate PTP profiles.
- 123-127 Reserved.
- 128-132 Reserved.
- 133-170 For use by alternate PTP profiles.
- 171-186 Reserved.
- 187 Degradation alternative B for a clock of clockClass 7 that is not within holdover specification. A clock of clockClass 187 may be a slave to another clock in the domain.
- 188-192 Reserved.
- 193 Degradation alternative B for a clock of clockClass 14 that is not within holdover specification. A clock of clockClass 193 may be a slave to another clock in the domain.
- 194-215 Reserved.
- 216-232 For use by alternate PTP profiles.
- 233-247 Reserved.
- 248 Default. This clockClass shall be used if none of the other clockClass definitions apply.
- 249-250 Reserved.
- 251 Reserved for version 1 compatibility; see Clause 18.
- 252-254 Reserved.
- 255 Shall be the clockClass of a slave-only clock; see 9.2.2."

REFERENCE "section 5.3.7, 7.6.2.4 and Table 5 of
[IEEE Std. 1588-2008]."

SYNTAX Unsigned32 (0..255)

ClockRoleType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The Clock Role. The protocol generates a Master Slave relationship among the clocks in the system.

Clock Role	Value	Description
Master clock	1	A clock that is the source of time to which all other clocks on that path synchronize.
Slave clock	2	A clock which synchronizes to another clock (master)."

SYNTAX INTEGER {
 master(1),
 slave(2)
}

ClockStateType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The clock state returned by PTP engine.

Clock State	Value	Description
Freerun state	1	Applies to a slave device that is not locked to a master. This is the initial state a slave starts out with when it is not getting any PTP packets from the master or because of some other input error (erroneous packets, etc).
Holdover state	2	In this state the slave device is locked to a master but communication with the master 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 Locked 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 Std. 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 [RFC 2578] 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 Std. 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
                    internalOillator(160) -- 0xA0
                  }
```

ClockTimeInterval ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention corresponds to the TimeInterval structure indicated in section 5.3.2 of [IEEE Std. 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 Std. 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 Std. 1588-2008]."  
SYNTAX        INTEGER {  
                ordinaryClock(1),  
                boundaryClock(2),  
                transparentClock(3),  
                boundaryNode(4)  
            }  
ietfPtpMIBNotifs OBJECT IDENTIFIER  
    ::= { ietfPtpMIB 0 }  
  
ietfPtpMIBObjects OBJECT IDENTIFIER  
    ::= { ietfPtpMIB 1 }  
  
ietfPtpMIBConformance OBJECT IDENTIFIER  
    ::= { ietfPtpMIB 2 }  
  
ietfPtpMIBSystemInfo OBJECT IDENTIFIER  
    ::= { ietfPtpMIBObjects 1 }  
  
-- Conformance Information Definition  
  
ietfPtpMIBCompliances OBJECT IDENTIFIER  
    ::= { ietfPtpMIBConformance 1 }  
  
ietfPtpMIBGroups OBJECT IDENTIFIER  
    ::= { ietfPtpMIBConformance 2 }  
  
ietfPtpMIBCompliances1 MODULE-COMPLIANCE  
    STATUS      current  
    DESCRIPTION  
        "Compliance statement for agents that provide read-only support  
        for IETF-PTP-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 { ietfPtpMIBSystemInfoGroup }  
    ::= { ietfPtpMIBCompliances 1 }  
  
ietfPtpMIBCompliances2 MODULE-COMPLIANCE  
    STATUS      current  
    DESCRIPTION  
        "Compliance statement for agents that provide read-only support  
        for IETF-PTP-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 {
    ietfPtpMIBClockCurrentDSGroup,
    ietfPtpMIBClockParentDSGroup,
    ietfPtpMIBClockDefaultDSGroup,
    ietfPtpMIBClockRunningGroup,
    ietfPtpMIBClockTimepropertiesGroup
}
 ::= { ietfPtpMIBCompliances 2 }

ietfPtpMIBCompliances3 MODULE-COMPLIANCE
STATUS          current
DESCRIPTION
    "Compliance statement for agents that provide read-only support
    for IETF-PTP-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 {
    ietfPtpMIBClockPortGroup,
    ietfPtpMIBClockPortDSGroup,
    ietfPtpMIBClockPortRunningGroup,
    ietfPtpMIBClockPortAssociateGroup
}
 ::= { ietfPtpMIBCompliances 3 }

ietfPtpMIBCompliances4 MODULE-COMPLIANCE
STATUS          current
DESCRIPTION
    "Compliance statement for agents that provide read-only support
    for IETF-PTP-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 {
    ietfPtpMIBClockTranparentDSGroup,
    ietfPtpMIBClockPortTransDSGroup
}
 ::= { ietfPtpMIBCompliances 4 }
```

```
ietfPtpMIBSystemInfoGroup OBJECT-GROUP
  OBJECTS      {
                ptpIetfSystemDomainTotals,
                ptpDomainClockPortsTotal,
                ptpIetfSystemProfile
              }
  STATUS       current
  DESCRIPTION  "Group which aggregates objects describing system-wide
               information"
  ::= { ietfPtpMIBGroups 1 }

ietfPtpMIBClockCurrentDSGroup OBJECT-GROUP
  OBJECTS      {
                ptpIetfClockCurrentDSStepsRemoved,
                ptpIetfClockCurrentDSOffsetFromMaster,
                ptpIetfClockCurrentDSMeanPathDelay
              }
  STATUS       current
  DESCRIPTION  "Group which aggregates objects describing PTP Current Dataset
               information"
  ::= { ietfPtpMIBGroups 2 }

ietfPtpMIBClockParentDSGroup OBJECT-GROUP
  OBJECTS      {
                ptpIetfClockParentDSParentPortIdentity,
                ptpIetfClockParentDSParentStats,
                ptpIetfClockParentDSOffset,
                ptpIetfClockParentDSClockPhChRate,
                ptpIetfClockParentDSGMClockIdentity,
                ptpIetfClockParentDSGMClockPriority1,
                ptpIetfClockParentDSGMClockPriority2,
                ptpIetfClockParentDSGMClockQualityClass,
                ptpIetfClockParentDSGMClockQualityAccuracy,
                ptpIetfClockParentDSGMClockQualityOffset
              }
  STATUS       current
  DESCRIPTION  "Group which aggregates objects describing PTP Parent Dataset
               information"
  ::= { ietfPtpMIBGroups 3 }

ietfPtpMIBClockDefaultDSGroup OBJECT-GROUP
  OBJECTS      {
                ptpIetfClockDefaultDSTwoStepFlag,
                ptpIetfClockDefaultDSClockIdentity,
                ptpIetfClockDefaultDSPriority1,

```

```

        ptpIetfClockDefaultDSPriority2,
        ptpIetfClockDefaultDSSlaveOnly,
        ptpIetfClockDefaultDSQualityClass,
        ptpIetfClockDefaultDSQualityAccuracy,
        ptpIetfClockDefaultDSQualityOffset
    }
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing PTP Default Dataset
    information"
 ::= { ietfPtpMIBGroups 4 }

ietfPtpMIBClockRunningGroup OBJECT-GROUP
OBJECTS         {
    ptpIetfClockRunningState,
    ptpIetfClockRunningPacketsSent,
    ptpIetfClockRunningPacketsReceived
}
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing PTP running state
    information"
 ::= { ietfPtpMIBGroups 5 }

ietfPtpMIBClockTimepropertiesGroup OBJECT-GROUP
OBJECTS         {
    ptpIetfClockTimePropertiesDSCurrentUTCOffsetValid,
    ptpIetfClockTimePropertiesDSCurrentUTCOffset,
    ptpIetfClockTimePropertiesDSLeap59,
    ptpIetfClockTimePropertiesDSLeap61,
    ptpIetfClockTimePropertiesDSTimeTraceable,
    ptpIetfClockTimePropertiesDSFreqTraceable,
    ptpIetfClockTimePropertiesDSPTPTimescale,
    ptpIetfClockTimePropertiesDSSource
}
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing PTP Time Properties
    information"
 ::= { ietfPtpMIBGroups 6 }

ietfPtpMIBClockTranparentDSGroup OBJECT-GROUP
OBJECTS         {
    ptpIetfClockTransDefaultDSClockIdentity,
    ptpIetfClockTransDefaultDSNumOfPorts,
    ptpIetfClockTransDefaultDSDelay,
    ptpIetfClockTransDefaultDSPrimaryDomain
}
STATUS          current
```

DESCRIPTION

"Group which aggregates objects describing PTP Transparent Dataset information"

::= { ietfPtpMIBGroups 7 }

ietfPtpMIBClockPortGroup OBJECT-GROUP

OBJECTS {
 ptpIetfClockPortName,
 ptpIetfClockPortSyncOneStep,
 ptpIetfClockPortCurrentPeerAddress,
 ptpIetfClockPortNumOfAssociatedPorts,
 ptpIetfClockPortCurrentPeerAddressType,
 ptpIetfClockPortRole
}

STATUS current

DESCRIPTION

"Group which aggregates objects describing information for a given PTP Port."

::= { ietfPtpMIBGroups 8 }

ietfPtpMIBClockPortDSGroup OBJECT-GROUP

OBJECTS {
 ptpIetfClockPortDSName,
 ptpIetfClockPortDSPortIdentity,
 ptpIetfClockPortDSAnnouncementInterval,
 ptpIetfClockPortDSAnnounceRctTimeout,
 ptpIetfClockPortDSSyncInterval,
 ptpIetfClockPortDSMinDelayReqInterval,
 ptpIetfClockPortDSPeerDelayReqInterval,
 ptpIetfClockPortDSDelayMech,
 ptpIetfClockPortDSPeerMeanPathDelay,
 ptpIetfClockPortDSGrantDuration,
 ptpIetfClockPortDSPTPVersion
}

STATUS current

DESCRIPTION

"Group which aggregates objects describing PTP Port Dataset information"

::= { ietfPtpMIBGroups 9 }

ietfPtpMIBClockPortRunningGroup OBJECT-GROUP

OBJECTS {
 ptpIetfClockPortRunningName,
 ptpIetfClockPortRunningState,
 ptpIetfClockPortRunningRole,
 ptpIetfClockPortRunningInterfaceIndex,
 ptpIetfClockPortRunningIPversion,
 ptpIetfClockPortRunningEncapsulationType,
}

```

        ptpIetfClockPortRunningTxMode,
        ptpIetfClockPortRunningRxMode,
        ptpIetfClockPortRunningPacketsReceived,
        ptpIetfClockPortRunningPacketsSent
    }
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing PTP running interface
    information"
 ::= { ietfPtpMIBGroups 10 }

ietfPtpMIBClockPortTransDSGroup OBJECT-GROUP
OBJECTS         {
    ptpIetfClockPortTransDSPortIdentity,
    ptpIetfClockPortTransDSlogMinPdelayReqInt,
    ptpIetfClockPortTransDSFaultyFlag,
    ptpIetfClockPortTransDSPeerMeanPathDelay
}
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing PTP TransparentDS
    Dataset
    information"
 ::= { ietfPtpMIBGroups 11 }

ietfPtpMIBClockPortAssociateGroup OBJECT-GROUP
OBJECTS         {
    ptpIetfClockPortAssociatePacketsSent,
    ptpIetfClockPortAssociatePacketsReceived,
    ptpIetfClockPortAssociateAddress,
    ptpIetfClockPortAssociateAddressType,
    ptpIetfClockPortAssociateInErrors,
    ptpIetfClockPortAssociateOutErrors
}
STATUS          current
DESCRIPTION
    "Group which aggregates objects describing information on peer
    PTP ports for a given PTP clock-port."
 ::= { ietfPtpMIBGroups 12 }
ietfPtpMIBClockInfo OBJECT IDENTIFIER
 ::= { ietfPtpMIBObjects 2 }

ptpIetfSystemTable OBJECT-TYPE
SYNTAX          SEQUENCE OF PtpIetfSystemEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "Table of count information about the PTP system for all
```

```

        domains."
 ::= { ietfPtpMIBSystemInfo 1 }

ptpIetfSystemEntry OBJECT-TYPE
    SYNTAX      PtpIetfSystemEntry
    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
            }
 ::= { ptpIetfSystemTable 1 }

PtpIetfSystemEntry ::= 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"
 ::= { ptpIetfSystemEntry 1 }

ptpInstanceIndex OBJECT-TYPE
    SYNTAX      ClockInstanceType (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the instance of the Clock for this

```

```
    domain."
 ::= { ptpIetfSystemEntry 2 }

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

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

ptpIetfSystemDomainEntry OBJECT-TYPE
    SYNTAX          PtpIetfSystemDomainEntry
    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           { ptpIetfSystemDomainClockTypeIndex }
 ::= { ptpIetfSystemDomainTable 1 }

PtpIetfSystemDomainEntry ::= SEQUENCE {
    ptpIetfSystemDomainClockTypeIndex ClockType,
    ptpIetfSystemDomainTotals         Gauge32
}

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

```

ptpIetfSystemDomainTotals OBJECT-TYPE
    SYNTAX      Gauge32
    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."
    ::= { ptpIetfSystemDomainEntry 2 }

ptpIetfSystemProfile 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 Std. 1588-2008]"
    ::= { ietfPtpMIBSystemInfo 3 }

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

ptpIetfClockCurrentDSEntry OBJECT-TYPE
    SYNTAX      PtpIetfClockCurrentDSEntry
    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 Std. 1588-2008]"
    INDEX       {
        ptpIetfClockCurrentDSDomainIndex,
        ptpIetfClockCurrentDSClockTypeIndex,
        ptpIetfClockCurrentDSInstanceIndex
    }
    ::= { ptpIetfClockCurrentDSTable 1 }

PtpIetfClockCurrentDSEntry ::= SEQUENCE {
    ptpIetfClockCurrentDSDomainIndex  ClockDomainType,
    ptpIetfClockCurrentDSClockTypeIndex  ClockType,

```



```
    ptpIetfClockCurrentDSInstanceIndex      ClockInstanceType,
    ptpIetfClockCurrentDSStepsRemoved       Counter32,
    ptpIetfClockCurrentDSOffsetFromMaster   ClockTimeInterval,
    ptpIetfClockCurrentDSMeanPathDelay     ClockTimeInterval
}

ptpIetfClockCurrentDSDomainIndex 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."
    ::= { ptpIetfClockCurrentDSEntry 1 }

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

ptpIetfClockCurrentDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType (0..255)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockCurrentDSEntry 3 }

ptpIetfClockCurrentDSStepsRemoved OBJECT-TYPE
    SYNTAX          Counter32
    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"
    ::= { ptpIetfClockCurrentDSEntry 4 }

ptpIetfClockCurrentDSOffsetFromMaster 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 "Section 8.2.2.3 of [IEEE Std. 1588-2008]"
::= { ptpIetfClockCurrentDSEntry 5 }

ptpIetfClockCurrentDSMeanPathDelay OBJECT-TYPE

SYNTAX ClockTimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the current clock dataset
MeanPathDelay value.

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

ptpIetfClockParentDSTable OBJECT-TYPE

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

ptpIetfClockParentDSEntry OBJECT-TYPE

SYNTAX PtpIetfClockParentDSEntry
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 Std. 1588-2008]"
INDEX {
 ptpIetfClockParentDSDomainIndex,
 ptpIetfClockParentDSClockTypeIndex,
 ptpIetfClockParentDSInstanceIndex
}
::= { ptpIetfClockParentDSTable 1 }

```

PtpIetfClockParentDSEntry ::= SEQUENCE {
    ptpIetfClockParentDSDomainIndex          ClockDomainType,
    ptpIetfClockParentDSClockTypeIndex      ClockType,
    ptpIetfClockParentDSInstanceIndex       ClockInstanceType,
    ptpIetfClockParentDSParentPortIdentity  OCTET STRING,
    ptpIetfClockParentDSParentStats         TruthValue,
    ptpIetfClockParentDSOffset              ClockIntervalBase2,
    ptpIetfClockParentDSClockPhChRate       Integer32,
    ptpIetfClockParentDSGMClockIdentity     ClockIdentity,
    ptpIetfClockParentDSGMClockPriority1    Integer32,
    ptpIetfClockParentDSGMClockPriority2    Integer32,
    ptpIetfClockParentDSGMClockQualityClass ClockQualityClassType,
    ptpIetfClockParentDSGMClockQualityAccuracy ClockQualityAccuracyType,
    ptpIetfClockParentDSGMClockQualityOffset Unsigned32
}

```

```

ptpIetfClockParentDSDomainIndex 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."
    ::= { ptpIetfClockParentDSEntry 1 }

```

```

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

```

```

ptpIetfClockParentDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType (0..255)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockParentDSEntry 3 }

```

```

ptpIetfClockParentDSParentPortIdentity OBJECT-TYPE
    SYNTAX          OCTET STRING
    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 Std. 1588-2008]"

::= { ptpIetfClockParentDSEntry 4 }

ptpIetfClockParentDSParentStats 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 Std. 1588-2008]"

::= { ptpIetfClockParentDSEntry 5 }

ptpIetfClockParentDSOffset 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 Std. 1588-2008]"

::= { ptpIetfClockParentDSEntry 6 }

ptpIetfClockParentDSClockPhChRate 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 Std. 1588-2008]"
 ::= { ptpIetfClockParentDSEntry 7 }
```

ptpIetfClockParentDSGMClockIdentity OBJECT-TYPE

SYNTAX ClockIdentity

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the parent dataset Grandmaster clock identity."

REFERENCE

"section 8.2.3.6 parentDS.grandmasterIdentity of [IEEE Std. 1588-2008]"

```
 ::= { ptpIetfClockParentDSEntry 8 }
```

ptpIetfClockParentDSGMClockPriority1 OBJECT-TYPE

SYNTAX Integer32

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 Std. 1588-2008]"

```
 ::= { ptpIetfClockParentDSEntry 9 }
```

ptpIetfClockParentDSGMClockPriority2 OBJECT-TYPE

SYNTAX Integer32

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 Std. 1588-2008]"

```
 ::= { ptpIetfClockParentDSEntry 10 }
```

ptpIetfClockParentDSGMClockQualityClass 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 Std. 1588-2008]"
 ::= { ptpIetfClockParentDSEntry 11 }

ptpIetfClockParentDSGMClockQualityAccuracy 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 Std. 1588-2008]"
 ::= { ptpIetfClockParentDSEntry 12 }

ptpIetfClockParentDSGMClockQualityOffset 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 Std. 1588-2008]"
 ::= { ptpIetfClockParentDSEntry 13 }

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

ptpIetfClockDefaultDSEntry OBJECT-TYPE
SYNTAX          PtpIetfClockDefaultDSEntry
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          {
                ptpIetfClockDefaultDSDomainIndex,
                ptpIetfClockDefaultDSClockTypeIndex,
                ptpIetfClockDefaultDSInstanceIndex
            }
```

```

    }
    ::= { ptpIetfClockDefaultDSTable 1 }

PtpIetfClockDefaultDSEntry ::= SEQUENCE {
    ptpIetfClockDefaultDSDomainIndex      ClockDomainType,
    ptpIetfClockDefaultDSClockTypeIndex   ClockType,
    ptpIetfClockDefaultDSInstanceIndex    ClockInstanceType,
    ptpIetfClockDefaultDSTwoStepFlag      TruthValue,
    ptpIetfClockDefaultDSClockIdentity    ClockIdentity,
    ptpIetfClockDefaultDSPriority1        Integer32,
    ptpIetfClockDefaultDSPriority2        Integer32,
    ptpIetfClockDefaultDSSlaveOnly        TruthValue,
    ptpIetfClockDefaultDSQualityClass     ClockQualityClassType,
    ptpIetfClockDefaultDSQualityAccuracy  ClockQualityAccuracyType,
    ptpIetfClockDefaultDSQualityOffset    Integer32
}

ptpIetfClockDefaultDSDomainIndex 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."
    ::= { ptpIetfClockDefaultDSEntry 1 }

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

ptpIetfClockDefaultDSInstanceIndex OBJECT-TYPE
    SYNTAX      ClockInstanceType (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockDefaultDSEntry 3 }

ptpIetfClockDefaultDSTwoStepFlag OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

```
    "This object specifies whether the Two Step process is used."  
 ::= { ptpIetfClockDefaultDSEntry 4 }
```

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

```
ptpIetfClockDefaultDSPriority1 OBJECT-TYPE  
SYNTAX      Integer32  
MAX-ACCESS  read-only  
STATUS      current  
DESCRIPTION  
    "This object specifies the default Datasets clock Priority1."  
 ::= { ptpIetfClockDefaultDSEntry 6 }
```

```
ptpIetfClockDefaultDSPriority2 OBJECT-TYPE  
SYNTAX      Integer32  
MAX-ACCESS  read-only  
STATUS      current  
DESCRIPTION  
    "This object specifies the default Datasets clock Priority2."  
 ::= { ptpIetfClockDefaultDSEntry 7 }
```

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

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

```
ptpIetfClockDefaultDSQualityAccuracy OBJECT-TYPE  
SYNTAX      ClockQualityAccuracyType  
MAX-ACCESS  read-only  
STATUS      current  
DESCRIPTION  
    "This object specifies the default dataset Quality Accuracy."
```



```
 ::= { ptpIetfClockDefaultDSEntry 10 }
```

```
ptpIetfClockDefaultDSQualityOffset OBJECT-TYPE
```

```
SYNTAX Integer32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"This object specifies the default dataset Quality offset."
```

```
 ::= { ptpIetfClockDefaultDSEntry 11 }
```

```
ptpIetfClockRunningTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF PtpIetfClockRunningEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Table of information about the PTP clock Running Datasets for all domains."
```

```
 ::= { ietfPtpMIBClockInfo 4 }
```

```
ptpIetfClockRunningEntry OBJECT-TYPE
```

```
SYNTAX PtpIetfClockRunningEntry
```

```
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 {
    ptpIetfClockRunningDomainIndex,
    ptpIetfClockRunningClockTypeIndex,
    ptpIetfClockRunningInstanceIndex
}
```

```
 ::= { ptpIetfClockRunningTable 1 }
```

```
PtpIetfClockRunningEntry ::= SEQUENCE {
```

```
    ptpIetfClockRunningDomainIndex ClockDomainType,
```

```
    ptpIetfClockRunningClockTypeIndex ClockType,
```

```
    ptpIetfClockRunningInstanceIndex ClockInstanceType,
```

```
    ptpIetfClockRunningState ClockStateType,
```

```
    ptpIetfClockRunningPacketsSent Counter64,
```

```
    ptpIetfClockRunningPacketsReceived Counter64
```

```
}
```

```
ptpIetfClockRunningDomainIndex 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."
 ::= { ptpIetfClockRunningEntry 1 }

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

ptpIetfClockRunningInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType (0..255)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
 ::= { ptpIetfClockRunningEntry 3 }

ptpIetfClockRunningState 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."

::= { ptpIetfClockRunningEntry 4 }

ptpIetfClockRunningPacketsSent OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the total number of all packet Unicast and multicast that have been sent out for this clock in this domain for this type."

::= { ptpIetfClockRunningEntry 5 }

ptpIetfClockRunningPacketsReceived 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."

::= { ptpIetfClockRunningEntry 6 }

ptpIetfClockTimePropertiesDSTable OBJECT-TYPE

SYNTAX SEQUENCE OF PtpIetfClockTimePropertiesDSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of information about the PTP clock Timeproperties Datasets for all domains."

::= { ietfPtpMIBClockInfo 5 }

ptpIetfClockTimePropertiesDSEntry OBJECT-TYPE

SYNTAX PtpIetfClockTimePropertiesDSEntry

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 Std. 1588-2008]"

INDEX

{
 ptpIetfClockTimePropertiesDSDomainIndex,
 ptpIetfClockTimePropertiesDSClockTypeIndex,
 ptpIetfClockTimePropertiesDSInstanceIndex

```

    }
    ::= { ptpIetfClockTimePropertiesDSTable 1 }

PtpIetfClockTimePropertiesDSEntry ::= SEQUENCE {
    ptpIetfClockTimePropertiesDSDomainIndex      ClockDomainType,
    ptpIetfClockTimePropertiesDSClockTypeIndex   ClockType,
    ptpIetfClockTimePropertiesDSInstanceIndex    ClockInstanceType,
    ptpIetfClockTimePropertiesDSCurrentUTCOffsetValid TruthValue,
    ptpIetfClockTimePropertiesDSCurrentUTCOffset Integer32,
    ptpIetfClockTimePropertiesDSLeap59          TruthValue,
    ptpIetfClockTimePropertiesDSLeap61         TruthValue,
    ptpIetfClockTimePropertiesDSTimeTraceable   TruthValue,
    ptpIetfClockTimePropertiesDSFreqTraceable   TruthValue,
    ptpIetfClockTimePropertiesDSPTPTimescale    TruthValue,
    ptpIetfClockTimePropertiesDSSource          ClockTimeSourceType
}

ptpIetfClockTimePropertiesDSDomainIndex 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."
    ::= { ptpIetfClockTimePropertiesDSEntry 1 }

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

ptpIetfClockTimePropertiesDSInstanceIndex OBJECT-TYPE
    SYNTAX      ClockInstanceType (0..255)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockTimePropertiesDSEntry 3 }

ptpIetfClockTimePropertiesDSCurrentUTCOffsetValid 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 Std. 1588-2008]"

::= { ptpIetfClockTimePropertiesDSEntry 4 }

ptpIetfClockTimePropertiesDSCurrentUTCOffset 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 Std. 1588-2008]"

::= { ptpIetfClockTimePropertiesDSEntry 5 }

ptpIetfClockTimePropertiesDSLeap59 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 Std. 1588-2008]"

::= { ptpIetfClockTimePropertiesDSEntry 6 }

ptpIetfClockTimePropertiesDSLeap61 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 Std. 1588-2008]"

::= { ptpIetfClockTimePropertiesDSEntry 7 }

ptpIetfClockTimePropertiesDSTimeTraceable 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 Std. 1588-2008]"
 ::= { ptpIetfClockTimePropertiesDSEntry 8 }

ptpIetfClockTimePropertiesDSFreqTraceable OBJECT-TYPE

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

ptpIetfClockTimePropertiesDSPTPTimescale 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 Std. 1588-2008]"
 ::= { ptpIetfClockTimePropertiesDSEntry 10 }

ptpIetfClockTimePropertiesDSSource 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 Std. 1588-2008]"
 ::= { ptpIetfClockTimePropertiesDSEntry 11 }

ptpIetfClockTransDefaultDSTable OBJECT-TYPE

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

```
ptpIetfClockTransDefaultDSEntry OBJECT-TYPE
    SYNTAX          PtpIetfClockTransDefaultDSEntry
    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 Std. 1588-2008]"
    INDEX          {
                    ptpIetfClockTransDefaultDSDomainIndex,
                    ptpIetfClockTransDefaultDSInstanceIndex
                }
    ::= { ptpIetfClockTransDefaultDSTable 1 }

PtpIetfClockTransDefaultDSEntry ::= SEQUENCE {
    ptpIetfClockTransDefaultDSDomainIndex    ClockDomainType,
    ptpIetfClockTransDefaultDSInstanceIndex  ClockInstanceType,
    ptpIetfClockTransDefaultDSClockIdentity  ClockIdentity,
    ptpIetfClockTransDefaultDSNumOfPorts    Counter32,
    ptpIetfClockTransDefaultDSDelay         ClockMechanismType,
    ptpIetfClockTransDefaultDSPrimaryDomain Integer32
}

ptpIetfClockTransDefaultDSDomainIndex 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."
    ::= { ptpIetfClockTransDefaultDSEntry 1 }

ptpIetfClockTransDefaultDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType (0..255)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockTransDefaultDSEntry 2 }

ptpIetfClockTransDefaultDSClockIdentity 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 Std. 1588-2008]"
```

```
::= { ptpIetfClockTransDefaultDSEntry 3 }
```

ptpIetfClockTransDefaultDSNumOfPorts 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 Std. 1588-2008]"

```
::= { ptpIetfClockTransDefaultDSEntry 4 }
```

ptpIetfClockTransDefaultDSDelay OBJECT-TYPE

SYNTAX ClockMechanismType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object, if the transparent clock is an end-to-end transparent clock, has the value shall be E2E; If the transparent clock is a peer-to-peer transparent clock, the value shall be P2P."

REFERENCE "Section 8.3.2.3.1 of [IEEE Std. 1588-2008]"

```
::= { ptpIetfClockTransDefaultDSEntry 5 }
```

ptpIetfClockTransDefaultDSPrimaryDomain OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the value of the primary synchronization domain. The initialization value shall be 0."

REFERENCE "Section 8.3.2.3.2 of [IEEE Std. 1588-2008]"

```
::= { ptpIetfClockTransDefaultDSEntry 6 }
```

ptpIetfClockPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF PtpIetfClockPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of information about the clock ports for a particular domain."

```
::= { ietfPtpMIBClockInfo 7 }
```

ptpIetfClockPortEntry OBJECT-TYPE

SYNTAX PtpIetfClockPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a single clock port."

```
INDEX          {
                ptpIetfClockPortDomainIndex,
                ptpIetfClockPortClockTypeIndex,
                ptpIetfClockPortClockInstanceIndex,
                ptpIetfClockPortTablePortNumberIndex
            }
 ::= { ptpIetfClockPortTable 1 }
```

```
PtpIetfClockPortEntry ::= SEQUENCE {
    ptpIetfClockPortDomainIndex      ClockDomainType,
    ptpIetfClockPortClockTypeIndex   ClockType,
    ptpIetfClockPortClockInstanceIndex ClockInstanceType,
    ptpIetfClockPortTablePortNumberIndex ClockPortNumber,
    ptpIetfClockPortName              DisplayString,
    ptpIetfClockPortRole              ClockRoleType,
    ptpIetfClockPortSyncOneStep      TruthValue,
    ptpIetfClockPortCurrentPeerAddressType InetAddressType,
    ptpIetfClockPortCurrentPeerAddress InetAddress,
    ptpIetfClockPortNumOfAssociatedPorts Gauge32
}
```

ptpIetfClockPortDomainIndex 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."

```
::= { ptpIetfClockPortEntry 1 }
```

ptpIetfClockPortClockTypeIndex OBJECT-TYPE

```
SYNTAX          ClockType
MAX-ACCESS      not-accessible
STATUS          current
```

DESCRIPTION

"This object specifies the clock type as defined in the Textual convention description."

```
::= { ptpIetfClockPortEntry 2 }
```

ptpIetfClockPortClockInstanceIndex OBJECT-TYPE

```
SYNTAX          ClockInstanceType (0..255)
MAX-ACCESS      not-accessible
STATUS          current
```

DESCRIPTION

"This object specifies the instance of the clock for this clock type in the given domain."

```
 ::= { ptpIetfClockPortEntry 3 }
```

```
ptpIetfClockPortTablePortNumberIndex OBJECT-TYPE
```

```
SYNTAX          ClockPortNumber (1..65535)
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

```
DESCRIPTION
```

```
"This object specifies the PTP Portnumber for this port."
```

```
 ::= { ptpIetfClockPortEntry 4 }
```

```
ptpIetfClockPortName OBJECT-TYPE
```

```
SYNTAX          DisplayString (SIZE (1..64))
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
"This object specifies the PTP clock port name configured on the router."
```

```
 ::= { ptpIetfClockPortEntry 5 }
```

```
ptpIetfClockPortRole OBJECT-TYPE
```

```
SYNTAX          ClockRoleType
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
"This object describes the current role (slave/master) of the port."
```

```
 ::= { ptpIetfClockPortEntry 6 }
```

```
ptpIetfClockPortSyncOneStep 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."
```

```
 ::= { ptpIetfClockPortEntry 7 }
```

```
ptpIetfClockPortCurrentPeerAddressType 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)"

::= { ptpIetfClockPortEntry 8 }

ptpIetfClockPortCurrentPeerAddress OBJECT-TYPE

SYNTAX InetAddress

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)"

::= { ptpIetfClockPortEntry 9 }

ptpIetfClockPortNumOfAssociatedPorts 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)."

::= { ptpIetfClockPortEntry 10 }

ptpIetfClockPortDSTable OBJECT-TYPE

SYNTAX SEQUENCE OF PtpIetfClockPortDSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of information about the clock ports dataset for a particular domain."

```
 ::= { ietfPtpMIBClockInfo 8 }
```

```
ptpIetfClockPortDSEntry OBJECT-TYPE
```

```
SYNTAX          PtpIetfClockPortDSEntry
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "An entry in the table, containing port dataset information for  
    a single clock port."
```

```
INDEX           {
                ptpIetfClockPortDSDomainIndex,
                ptpIetfClockPortDSClockTypeIndex,
                ptpIetfClockPortDSClockInstanceIndex,
                ptpIetfClockPortDSPortNumberIndex
            }
```

```
 ::= { ptpIetfClockPortDSTable 1 }
```

```
PtpIetfClockPortDSEntry ::= SEQUENCE {
```

```
    ptpIetfClockPortDSDomainIndex          ClockDomainType,
```

```
    ptpIetfClockPortDSClockTypeIndex       ClockType,
```

```
    ptpIetfClockPortDSClockInstanceIndex   ClockInstanceType,
```

```
    ptpIetfClockPortDSPortNumberIndex     ClockPortNumber,
```

```
    ptpIetfClockPortDSName                 DisplayString,
```

```
    ptpIetfClockPortDSPortIdentity         OCTET STRING,
```

```
    ptpIetfClockPortDSAnnouncementInterval Integer32,
```

```
    ptpIetfClockPortDSAnnounceRctTimeout  Integer32,
```

```
    ptpIetfClockPortDSSyncInterval        Integer32,
```

```
    ptpIetfClockPortDSMinDelayReqInterval Integer32,
```

```
    ptpIetfClockPortDSPeerDelayReqInterval Integer32,
```

```
    ptpIetfClockPortDSDelayMech           ClockMechanismType,
```

```
    ptpIetfClockPortDSPeerMeanPathDelay   ClockTimeInterval,
```

```
    ptpIetfClockPortDSGrantDuration       Unsigned32,
```

```
    ptpIetfClockPortDSPTPVersion         Integer32
```

```
}
```

```
ptpIetfClockPortDSDomainIndex 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."
```

```
 ::= { ptpIetfClockPortDSEntry 1 }
```

```
ptpIetfClockPortDSClockTypeIndex OBJECT-TYPE
```

```
SYNTAX          ClockType
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

```
DESCRIPTION
```

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

ptpIetfClockPortDSClockInstanceIndex OBJECT-TYPE
SYNTAX ClockInstanceType (0..255)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "This object specifies the instance of the clock for this clock
 type in the given domain."
 ::= { ptpIetfClockPortDSEntry 3 }

ptpIetfClockPortDSPortNumberIndex OBJECT-TYPE
SYNTAX ClockPortNumber (1..65535)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "This object specifies the PTP portnumber associated with this
 PTP port."
 ::= { ptpIetfClockPortDSEntry 4 }

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

ptpIetfClockPortDSPortIdentity OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "This object specifies the PTP clock port Identity."
 ::= { ptpIetfClockPortDSEntry 6 }

ptpIetfClockPortDSAnnouncementInterval OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "This object specifies the Announce message transmission
 interval associated with this clock port."
 ::= { ptpIetfClockPortDSEntry 7 }

ptpIetfClockPortDSAnnounceRctTimeout OBJECT-TYPE
SYNTAX Integer32

MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the Announce receipt timeout associated
with this clock port."
 ::= { ptpIetfClockPortDSEntry 8 }

ptpIetfClockPortDSSyncInterval OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the Sync message transmission interval."
 ::= { ptpIetfClockPortDSEntry 9 }

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

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

ptpIetfClockPortDSDelayMech 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."
 ::= { ptpIetfClockPortDSEntry 12 }

ptpIetfClockPortDSPeerMeanPathDelay OBJECT-TYPE
SYNTAX ClockTimeInterval
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the peer meanPathDelay."

```
 ::= { ptpIetfClockPortDSEntry 13 }
```

```
ptpIetfClockPortDSGrantDuration OBJECT-TYPE
```

```
SYNTAX          Unsigned32
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "This object specifies the grant duration allocated by the
    master."
```

```
 ::= { ptpIetfClockPortDSEntry 14 }
```

```
ptpIetfClockPortDSPTPVersion OBJECT-TYPE
```

```
SYNTAX          Integer32
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "This object specifies the PTP version being used."
```

```
 ::= { ptpIetfClockPortDSEntry 15 }
```

```
ptpIetfClockPortRunningTable OBJECT-TYPE
```

```
SYNTAX          SEQUENCE OF PtpIetfClockPortRunningEntry
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "Table of information about the clock ports running dataset for
    a particular domain."
```

```
 ::= { ietfPtpMIBClockInfo 9 }
```

```
ptpIetfClockPortRunningEntry OBJECT-TYPE
```

```
SYNTAX          PtpIetfClockPortRunningEntry
```

```
MAX-ACCESS      not-accessible
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "An entry in the table, containing running dataset information
    about a single clock port."
```

```
INDEX
```

```
    {
        ptpIetfClockPortRunningDomainIndex,
        ptpIetfClockPortRunningClockTypeIndex,
        ptpIetfClockPortRunningClockInstanceIndex,
        ptpIetfClockPortRunningPortNumberIndex
    }
```

```
 ::= { ptpIetfClockPortRunningTable 1 }
```

```
PtpIetfClockPortRunningEntry ::= SEQUENCE {
```

```
    ptpIetfClockPortRunningDomainIndex      ClockDomainType,
```

```
    ptpIetfClockPortRunningClockTypeIndex   ClockType,
```

```
    ptpIetfClockPortRunningClockInstanceIndex ClockInstanceType,
```

```

    ptpIetfClockPortRunningPortNumberIndex    ClockPortNumber,
    ptpIetfClockPortRunningName                DisplayString,
    ptpIetfClockPortRunningState              ClockPortState,
    ptpIetfClockPortRunningRole               ClockRoleType,
    ptpIetfClockPortRunningInterfaceIndex     InterfaceIndexOrZero,
    ptpIetfClockPortRunningIPversion         Integer32,
    ptpIetfClockPortRunningEncapsulationType Integer32,
    ptpIetfClockPortRunningTxMode            ClockTxModeType,
    ptpIetfClockPortRunningRxMode            ClockTxModeType,
    ptpIetfClockPortRunningPacketsReceived    Counter64,
    ptpIetfClockPortRunningPacketsSent       Counter64
}

```

ptpIetfClockPortRunningDomainIndex 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."

```
 ::= { ptpIetfClockPortRunningEntry 1 }
```

ptpIetfClockPortRunningClockTypeIndex OBJECT-TYPE

```

SYNTAX          ClockType
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION

```

"This object specifies the clock type as defined in the Textual convention description."

```
 ::= { ptpIetfClockPortRunningEntry 2 }
```

ptpIetfClockPortRunningClockInstanceIndex OBJECT-TYPE

```

SYNTAX          ClockInstanceType (0..255)
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION

```

"This object specifies the instance of the clock for this clock type in the given domain."

```
 ::= { ptpIetfClockPortRunningEntry 3 }
```

ptpIetfClockPortRunningPortNumberIndex OBJECT-TYPE

```

SYNTAX          ClockPortNumber (1..65535)
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION

```

"This object specifies the PTP portnumber associated with this clock port."

```
 ::= { ptpIetfClockPortRunningEntry 4 }
```


ptpIetfClockPortRunningName OBJECT-TYPE

SYNTAX DisplayString (SIZE (1..64))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the PTP clock port name."

::= { ptpIetfClockPortRunningEntry 5 }

ptpIetfClockPortRunningState OBJECT-TYPE

SYNTAX ClockPortState

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

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

::= { ptpIetfClockPortRunningEntry 6 }

ptpIetfClockPortRunningRole OBJECT-TYPE

SYNTAX ClockRoleType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the Clock Role."

::= { ptpIetfClockPortRunningEntry 7 }

ptpIetfClockPortRunningInterfaceIndex 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."

::= { ptpIetfClockPortRunningEntry 8 }

ptpIetfClockPortRunningIPversion 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)."

::= { ptpIetfClockPortRunningEntry 9 }

ptpIetfClockPortRunningEncapsulationType 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 (eg. VLAN, Pseudowire encapsulation...) for the PTP messages."

::= { ptpIetfClockPortRunningEntry 10 }

ptpIetfClockPortRunningTxMode 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"

::= { ptpIetfClockPortRunningEntry 11 }

ptpIetfClockPortRunningRxMode 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"

::= { ptpIetfClockPortRunningEntry 12 }

ptpIetfClockPortRunningPacketsReceived OBJECT-TYPE

SYNTAX Counter64

UNITS "packets"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the packets received on the clock port (cumulative)."

::= { ptpIetfClockPortRunningEntry 13 }

ptpIetfClockPortRunningPacketsSent OBJECT-TYPE

SYNTAX Counter64

UNITS "packets"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the packets sent on the clock port (cumulative)."

::= { ptpIetfClockPortRunningEntry 14 }

ptpIetfClockPortTransDSTable OBJECT-TYPE

SYNTAX SEQUENCE OF PtpIetfClockPortTransDSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of information about the Transparent clock ports running dataset for a particular domain."

::= { ietfPtpMIBClockInfo 10 }

ptpIetfClockPortTransDSEntry OBJECT-TYPE

SYNTAX PtpIetfClockPortTransDSEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing clock port Transparent dataset information about a single clock port"

INDEX {
 ptpIetfClockPortTransDSDomainIndex,
 ptpIetfClockPortTransDSInstanceIndex,

```

        ptpIetfClockPortTransDSPortNumberIndex
    }
    ::= { ptpIetfClockPortTransDSTable 1 }

PtpIetfClockPortTransDSEntry ::= SEQUENCE {
    ptpIetfClockPortTransDSDomainIndex      ClockDomainType,
    ptpIetfClockPortTransDSInstanceIndex    ClockInstanceType,
    ptpIetfClockPortTransDSPortNumberIndex  ClockPortNumber,
    ptpIetfClockPortTransDSPortIdentity     ClockIdentity,
    ptpIetfClockPortTransDSlogMinPdelayReqInt Integer32,
    ptpIetfClockPortTransDSFaultyFlag      TruthValue,
    ptpIetfClockPortTransDSPeerMeanPathDelay ClockTimeInterval
}

ptpIetfClockPortTransDSDomainIndex 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."
    ::= { ptpIetfClockPortTransDSEntry 1 }

ptpIetfClockPortTransDSInstanceIndex OBJECT-TYPE
    SYNTAX      ClockInstanceType (0..255)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpIetfClockPortTransDSEntry 2 }

ptpIetfClockPortTransDSPortNumberIndex 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 of
                [IEEE Std. 1588-2008]"
    ::= { ptpIetfClockPortTransDSEntry 3 }

ptpIetfClockPortTransDSPortIdentity 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 Std. 1588-2008]"
::= { ptpIetfClockPortTransDSEntry 4 }

ptpIetfClockPortTransDSlogMinPdelayReqInt 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 Std. 1588-2008]"
::= { ptpIetfClockPortTransDSEntry 5 }

ptpIetfClockPortTransDSFaultyFlag OBJECT-TYPE

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

ptpIetfClockPortTransDSPeerMeanPathDelay 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 8.3.3.3.3 of [IEEE Std. 1588-2008]"
::= { ptpIetfClockPortTransDSEntry 7 }

ptpIetfClockPortAssociateTable OBJECT-TYPE

SYNTAX SEQUENCE OF PtpIetfClockPortAssociateEntry
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."

```
::= { ietfPtpMIBClockInfo 11 }
```

ptpIetfClockPortAssociateEntry OBJECT-TYPE

SYNTAX PtpIetfClockPortAssociateEntry

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,
    ptpIetfClockPortAssociatePortIndex
}
```

```
::= { ptpIetfClockPortAssociateTable 1 }
```

```
PtpIetfClockPortAssociateEntry ::= SEQUENCE {
    ptpClockPortCurrentDomainIndex      ClockDomainType,
    ptpClockPortCurrentClockTypeIndex   ClockType,
    ptpClockPortCurrentClockInstanceIndex ClockInstanceType,
    ptpClockPortCurrentPortNumberIndex  ClockPortNumber,
    ptpIetfClockPortAssociatePortIndex  Unsigned32,
    ptpIetfClockPortAssociateAddressType InetAddressType,
    ptpIetfClockPortAssociateAddress     InetAddress,
    ptpIetfClockPortAssociatePacketsSent Counter64,
    ptpIetfClockPortAssociatePacketsReceived Counter64,
    ptpIetfClockPortAssociateInErrors    Counter64,
    ptpIetfClockPortAssociateOutErrors   Counter64
}
```

ptpClockPortCurrentDomainIndex OBJECT-TYPE

SYNTAX ClockDomainType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object specifies the given port's domain number."

```
::= { ptpIetfClockPortAssociateEntry 1 }
```

ptpClockPortCurrentClockTypeIndex OBJECT-TYPE

SYNTAX ClockType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object specifies the given port's clock type."

::= { ptpIetfClockPortAssociateEntry 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."

::= { ptpIetfClockPortAssociateEntry 3 }

ptpClockPortCurrentPortNumberIndex OBJECT-TYPE

SYNTAX ClockPortNumber

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object specifies the PTP Port Number for the given port."

::= { ptpIetfClockPortAssociateEntry 4 }

ptpIetfClockPortAssociatePortIndex 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."

::= { ptpIetfClockPortAssociateEntry 5 }

ptpIetfClockPortAssociateAddressType 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."

::= { ptpIetfClockPortAssociateEntry 6 }

ptpIetfClockPortAssociateAddress OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object specifies the peer port's network address used for PTP communication."

::= { ptpIetfClockPortAssociateEntry 7 }

```
ptpIetfClockPortAssociatePacketsSent 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."
    ::= { ptpIetfClockPortAssociateEntry 8 }

ptpIetfClockPortAssociatePacketsReceived OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "packets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of packets received from this peer port by the
        current port."
    ::= { ptpIetfClockPortAssociateEntry 9 }

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

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

END

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

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

7.2. Informative References

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

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

[RFC 1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions",

STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991

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

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

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

[RFC 2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.

[RFC 2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.

[RFC 2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.

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

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

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

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

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

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

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

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

[IEEE Std. 802.1AB-2009] "IEEE Standard for Local and Metropolitan area networks - Station and Media Access Control Connectivity Discovery", IEEE Std. 802.1AB(TM)-2009, 17 September 2008

[IEEE Std. 802.3-2008] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and Metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", IEEE Std. 802.3 - 2008, 26 December 2008

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

TICTOC
Internet Draft
Intended status: Informational
Expires: April 28, 2012

D. Marlow,
S. Knickerbocker,
T. Plunkett
NSWC-DD
October 28, 2011

Network Time Mechanisms for Improving Computer Clock Accuracy
draft-marlow-tictoc-computer-clock-accuracy-01.txt

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), 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/lid-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 28, 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.

Abstract

This draft describes network time synchronization mechanisms that may enable increased accuracy, beyond that possible with the current Network Time Protocol version 4 standard, to the time of computer clocks. The mechanisms considered are those that will provide improved estimates as to when a packet is put on the network, transferred across a network, or taken from the network. Potential standardization actions will be considered for the mechanisms considered, though no such actions are recommended at this time.

Table of Contents

1. Introduction.....	2
1.1. Motivation for Increased Performance.....	3
1.2. NTP/PTP Commonality and Differences.....	4
1.3. Performance and Security Threat/Network Error Tradeoff....	4
2. Use Case Targeted.....	5
2.1. Emerging Need for NTP and PTP Commonality.....	5
3. Approach.....	6
4. Mechanisms Considered.....	6
4.1. NTP Interleaved Modes.....	6
4.1.1. Standardization Considerations.....	7
4.2. Use of IEEE 1588 PTP and 802.1AS Mechanisms in the Underlying Network Service (e.g., Network Interface Controller [NIC]).....	7
4.2.1. Standardization Considerations.....	7
4.3. Use of IEEE 1588 PTP to Synchronize Computer Clocks.....	7
4.3.1. Standardization Considerations.....	8
5. Initial Experimentation.....	8
5.1. Naval Surface Warfare Center, Dahlgren Division (NSWCDD), Experimentation.....	8
5.2. Future Metrics and Benchmarks Standard.....	9
6. Analysis of Results.....	10
6.1. Analysis of Initial NSWCDD Experimentation.....	10
6.2. Analysis of Published Results.....	10
7. Future Experimentation.....	11
8. Security Considerations.....	12
9. Internet Assigned Numbers Authority (IANA) Considerations.....	12
10. Conclusions.....	12
11. Informative References.....	13
12. Acknowledgments.....	13

1. Introduction

The IETF Timing over IP Connection and Transfer of Clock (TICTOC) Working Group was formed to investigate emerging needs to distribute highly accurate time and frequency information over Internet Protocol (IP) and Multiprotocol Label Switching (MPLS) Packet

Switched Networks (PSNs). In this draft, new mechanisms beyond those identified in the Network Time Protocol version 4 (NTPv4) standard (i.e., Request for Comment 5905) are considered to provide increased time synchronization accuracy for computer (i.e., operating system) clocks' time and frequency. The mechanisms considered are those that will provide improved estimates as to when a packet is put on the network, transferred across a network, or taken from the network. This draft identifies a set of mechanisms that are candidates for experimentation. Standardization considerations will be described for the mechanisms identified.

In this draft, the authors examine methods for improving NTPv4 time synchronization performance. The authors are requesting comments and contributions on the mechanisms described and on additional mechanisms that should be considered. It is hoped that discussions within the IETF TICTOC Working Group will motivate experimentation that will lead to standardization actions to enable better accuracy to those utilizing a future Network Time Protocol (NTP) specification.

1.1. Motivation for Increased Performance

There are two reasons to improve upon the time synchronization performance that is currently available from the NTP. Not only is the increased performance needed for existing product designs that would make use of the added performance if it were available, but it is expected that new uses will be identified that are not even possible until performance is improved. This is similar to how network speeds are increased every several years, and the uses for the increased network bandwidth soon follow.

The current methods for achieving an increase in time synchronization performance involve use of a technology separate from the existing computer network (e.g., Inter-Range Instrumentation Group technology) or use of a technology like Precision Time Protocol (PTP), which is defined in the Institute of Electrical and Electronics Engineers (IEEE) 1588, "Precision Clock Synchronization Protocol for Networked Measurement and Control Systems." With PTP, the computer applications must interface with the installed PTP hardware in order to read time from its oscillator. There is a lot of resiliency built into NTP, which does not exist in the PTP protocol. It is unknown what happens to the time provided by the PTP hardware when a network switch in the network path to the time source is temporarily unavailable (i.e., the network switch gets rebooted). It would be beneficial to have the resiliency of the NTP algorithms be paired with the highly accurate PTP hardware-based time distribution.

1.2. NTP/PTP Commonality and Differences

NTP and PTP are both packet-based protocols for exchanging time with a time server over a computer network. Both protocols are used to determine the offset between two independent clocks. Both use embedded algorithms to construct a shortest path spanning tree for obtaining time from a master time source through intermediary time sources to time clients. Both assume network paths are symmetric and both have their own methods for addressing network delays that are not symmetric. NTP uses its algorithms to determine which of several consecutive time measurements are most accurate and uses that measurement. PTP makes use of hardware means of measuring delays as packets traverse intermediate network devices and corrects its received time information based upon those measured delays. Both have similar authentication provisions based on cryptographic message digests. [1, page 306]

NTP is engineered to synchronize computer clocks in an extended network, while PTP is engineered to synchronize device clocks in a segmented LAN [Local Area Network]. [1] Because PTP has the ability to measure actual packet delays and to correct for them, PTP can provide the most accurate measurement of clock offset between two clocks. PTP does not define the method for synchronizing that clock once the highly accurate time measurements have been obtained. PTP is normally used to synchronize a relatively high-quality hardware clock located on an interface card and does not synchronize the operating system clock. NTP, on the other hand, possesses the ability to synchronize the commodity-quality system clock based on received clock offset measurements. NTP is normally utilized where relatively long update intervals are required to minimize network load, while PTP is normally utilized on a high-speed LAN with no such requirement and operates with update intervals on the order of 2 seconds. On a LAN with reduced phase noise and shorter update intervals, PTP can provide far better performance than NTP, even if using the same commodity oscillator. [1, page 306] The NTP algorithms have a lot of resiliency so that operating system clocks stay stable despite the conditions on the network.

1.3. Performance and Security Threat/Network Error Tradeoff

Through discussions in the TICTOC Working Group, it has been pointed out that one of the differences between NTP and PTP are security threats and network errors that each is designed to work around. While PTP has few capabilities to work in the presence of security threats and network errors, NTP has been designed to work "in the wild." The inherent resiliency built into the NTP protocol contributes directly to its security, by handling security threats via the detection of duplicate, unsynchronized, or bogus packets. Thus PTP needs private isolated network interconnections, while NTP can run on the Internet in the presence of major threats (e.g., man-

in-the-middle attacks). In an e-mail to the TICTOC working group on June 8, 2011, Dr. David L. Mills wrote "... a primary motivation for the NTP interleaved design was protection from network errors and intruder attack. The detailed analysis and simulation are designed to demonstrate resistance to common corruptions such as dropped or duplicate packets and possible bogus attacks. The NTP design includes a four-level security model, the lower two levels might be considered for a PTP application. This is one of the most important difference(s) between the PTP and NTP protocol designs; however, the NTP design might be considered overkill in a sheltered, isolated Ethernet network." [2]

2. Use Case Targeted

The use case considered in this Internet-Draft is a dense concentration of computing elements connected by a network. A satellite-based time source (e.g., Global Positioning System [GPS]) is used for synchronizing primary time servers. Secondary time servers and leaf computing elements are synchronized to the primary time servers over the network. In this use case, there are approximately 150 or so total computers where there are three to four levels of time servers. These time servers may have to communicate to each other through layer 2 and layer 3 network switches, which could be 10-20 different layer 2 subnetworks. All of the computers are connected together through gigabit or faster network connections. In this environment, there will be some groups of computers that will need to synchronize to each other to within a microsecond, while other groups of computers only have to be synchronized to each other to within a millisecond. In this use case, there is one interconnected time synchronization scheme where NTP, PTP, or a combination of both is used to meet all time synchronization needs. The use case presented here does not identify a defined set of security threats or network errors in which a network time synchronization mechanism is to be able to safely work; however, proposed network time synchronization solutions need to identify the tradeoff taken between the performance achievable and the security threats and network errors within which it is intended to work.

2.1. Emerging Need for NTP and PTP Commonality

Because of its accuracy capabilities, PTP is beginning to replace NTP as the base protocol for time clients in dense computing sites. This results in an implementation in which some hosts use PTP while others within the same building and sometimes within the same room use NTP. Over time, hosts are being changed from NTP to PTP. This leads to an emerging need to provide similar approaches for basic time service functions for operational ease of managing time distribution assets. Examples of functions where commonality is considered to be an emerging need include providing synchronization

of computer clock, providing management to time clients, and configuring timeservers. A standard means of synchronizing computer clocks for both protocols is of particular interest; there appears to be no value in using different methods when the hosts that both protocols are supporting are often working within the same system.

In addition, there are highly accurate PTP time clients that could serve as highly accurate secondary timeservers for NTP time clients if this capability were supported in vendor products.

3. Approach

The approach taken by the authors includes determining what the current accuracy capabilities are with NTPv4 and investigating additional mechanisms that may provide improvements in accuracy. Any degradation in security capabilities and/or the ability to work through network errors needs to be assessed for any mechanism for which a standards action is pursued. Through experiments of those additional mechanisms, estimations of improvements can be calculated. Depending on the standardization difficulty and potential benefits offered, more than one standardization action may be recommended in the future.

4. Mechanisms Considered

4.1. NTP Interleaved Modes

The NTP interleaved modes are an extension of the NTPv4 protocol, which is included in the current NTP distribution [1]. It utilizes Broadcast and Symmetric modes (client/server is not supported) and is designed to be backward compatible (i.e., not affecting NTP implementations that do not use the interleaved extensions). It also utilizes the same NTP packet format as the current standard NTPv4. Security and network robustness capabilities were a major design factor for NTP interleaved; thus, it is expected that an NTP interleaved configuration is at least as secure or resistant to network errors as any other NTP operational mode. NTP interleaved uses an IEEE 1588 PTP-like feature that provides a follow-up packet with a better estimate of when a previous NTP packet was sent on the network and a message exchange sequence to determine network mean path delay.

This mechanism could be used by some of the primary time servers for synchronizing secondary (i.e., lower stratum) time servers and leaf computing elements, which have very accurate time synchronization requirements.

Future experimentation may identify what gains are possible with this mechanism. Dr. David L. Mills pointed out in TICTOC Working Group discussions that the interleaved modes provide a major

performance benefit when large Protocol Data Units are used (e.g., when NTP is used with the Autokey Protocol for time server authentication). [3]

4.1.1. Standardization Considerations

There are additional reasons beyond time accuracy improvements to standardize NTP interleaved modes of operation. As noted earlier, its operation with large Protocol Data Units may be reason enough to standardize the NTP interleaved modes. NTP interleaved modes also provide additional measurement parameters not available with other NTP modes. Follow-on IETF TICTOC Working Group discussions are needed to decide whether to initiate a standardization action to add interleaved options to the NTP standard.

4.2. Use of IEEE 1588 PTP and 802.1AS Mechanisms in the Underlying Network Service (e.g., Network Interface Controller [NIC])

The purpose of investigating this mechanism is to determine if using special capabilities in the underlying network service can improve the timestamp estimates when NTP packets are put on the network, transferred across networks, or taken from the network. It is apparent there are many new network integrated circuit devices as well as general purpose processors that perform IEEE 1588 PTP hardware time stamping to support emerging interactive multimedia services. Such integrated circuits are identifying IEEE 1588 PTP; however, they appear to be programmable, and it is possible that key NTP operations or encryption algorithms could be supported as well. This is an area where research and experimentation is needed. Identifying the security threats and network errors that can be handled is an important part of this research and experimentation.

4.2.1. Standardization Considerations

If the investigation of these mechanisms generates promising results, this may initiate a standardization proposal for additions to the NTPv4 standard to make use of these capabilities. Alternatively, a modification to the NTPv4 standard may be proposed, which would enable hardware assists to be incorporated into future NICs.

4.3. Use of IEEE 1588 PTP to Synchronize Computer Clocks

The purpose of investigating this mechanism is to determine the viability of using PTP to synchronize computing elements, which require very accurate synchronization. This mechanism considers bringing the PTP synchronization all the way to the computer clock through a standardized clock discipline algorithm. Computing elements synchronized by PTP are candidates to be time servers (by the use of NTP) for computing elements not synchronized by PTP.

Based on their respective strengths, the natural way to merge NTP and PTP would be to use PTP as the means of obtaining extremely accurate time information from across the network and to let the NTP algorithms use that time to keep local clocks synchronized. Tradeoffs between performance and security or network error handling capabilities are needed to fit each deployment environment considered.

4.3.1. Standardization Considerations

If the investigation of this mechanism generates promising results, it may initiate a standardization proposal to specify a PTP profile for use by NTP. It may be possible to replace the current NTP clock coordination services without affecting the NTP time management services or the clock access mechanisms used by each operating system. A variety of studies will be needed if this approach is to be pursued, including a study to determine if there are any issues for secondary time servers to run both NTP and PTP. If such issues are identified, standards activities may be needed in the IETF or in IEEE 1588. A study would be needed to identify the security threats and/or network errors that can be handled.

5. Initial Experimentation

5.1. Naval Surface Warfare Center, Dahlgren Division (NSWCDD), Experimentation

Some preliminary experiments tested the new interleaved mode available in NTP v4.2.6. This mode mimics the operation of PTP defined by IEEE 1588 where an additional follow-up message is sent so that a more accurate transmission time can be used. In this experiment, seven workstations running Red Hat Enterprise Linux 5 were used. These workstations are 3 years old and make use of two dual core processors. Since the GPS-based (i.e., stratum 1) time server does not currently have NTP v4.2.6 available, which supports the interleaved modes, one of the seven workstations was synchronized to the stratum 1 time server, which then served as the stratum 2 time server to the other six stratum 3 workstations. The stratum 2 server and four of the six other workstations were upgraded to use NTP v4.2.6, while the remaining two workstations were left running NTP v4.2.2 that was included with Red Hat Enterprise Linux 5.

Interleaved mode was achieved by using the "xleave" option when either the broadcast mode or peer mode was used under NTP. The stratum 2 server was configured as a broadcast server, making use of the standard multicast address and using the xleave option. Two of the workstations running NTP v4.2.6 were configured as multicast clients so that the Interleaved Broadcast mode was utilized. The other two NTP v4.2.6 workstations were configured to synchronize to

the stratum 2 server using standard Client/Server (unicast) mode. The two workstations running NTP v4.2.2 were configured to be broadcast clients; however, they did not use interleaved mode since NTP v4.2.2 does not include interleaved support. All NTP polling intervals were configured to 16 seconds.

Offset measurements were obtained between the six clients and the stratum 2 server using the "ntpdate" command with the "-q" option. Measurements were taken every minute over approximately 4 days. These workstations were not running any other major tasks, and NTP ran over a network with no discernable network load. All of the workstations were connected through the same Virtual LAN on the same network switch (i.e., no routers involved). All of the network connections were 100 Mbit/sec Ethernet.

Some experiment results were obtained where the average and standard deviations of the absolute value of clock offset were measured. The worst-behaved NTP Interleaved Broadcast client was able to stay synchronized with an average clock offset of 9 microseconds with a standard deviation of 8 microseconds. The worst-behaved computer that synchronized using Client/Server mode was able to maintain an average clock offset of 11 microseconds with a standard deviation of 10 microseconds. The worst-behaved Broadcast (without NTP interleaved) client stayed synchronized with an average clock offset of 49 microseconds and with a standard deviation of 58 microseconds.

These results illustrate that NTP Interleaved Broadcast does provide results that are better than having every client poll the server via unicast. However, the result is not significantly better (e.g., not an order of magnitude better). Preliminary experiments with hardware-based PTP have been performed in the past where the average offsets between PTP NICs and the PTP Grandmaster clock are in the hundreds of nanoseconds with standard deviations in the tens of nanoseconds.

5.2. Future Metrics and Benchmarks Standard

One thing that would help to perform computer time synchronization experiments is a set of time-synchronization performance metrics; no standard or even a paper was found on this topic. Developing a standard to define time synchronization performance metrics would be beneficial by allowing different experimental efforts to be performed in a way that the results are comparable.

An Internet-Draft addressing Benchmark Methodology for network time synchronization devices may be a good path to provide the needed metrics and guidance on experiments to be run. This benchmark methodology would be similar to those that the IETF Benchmarking Methodology Working Group has standardized for other areas. This document would include methodology specific to benchmarking the

performance of devices used for synchronizing time over a network. This document could define a model for time offset measurement, describe test setups, metrics to be collected, and background test environments. Specific benchmarks may vary between testing a single device and an entire system under test. The methodology could identify requirements for identifying test accuracy and for ensuring that measurements are independent of the devices being measured (e.g., that the measurement techniques are not profoundly influenced by the delay of the network upon which the measurement is made).

6. Analysis of Results

6.1. Analysis of Initial NSWCDD Experimentation

In 2008, it was reported [4] that the synchronization of computer clocks using NTPv4 over a LAN can approach values on the order of $\sim 145\mu\text{s}$ (mean plus 3 sigma). This has been demonstrated in the laboratory and was achieved across a single stratum (e.g., stratum 1 to stratum 2). Recently, using newer hardware and a newer NTP version, time synchronization values were measured on the order of $\sim 40\mu\text{s}$ (mean plus 3 sigma). The variation in the time synchronization comprises the majority of the $40\mu\text{s}$ value. The results need to be analyzed in detail, but in a little over two years, the time synchronization of computer clocks over a LAN has improved; most likely due to hardware improvements and minor software enhancements.

If this $40\mu\text{s}$ synchronization can be maintained from stratum to stratum for all subsequent tiers in a well-engineered network with 3 to 5 stratums, a maximum theoretical time synchronization offset on the order of $120\text{--}200\mu\text{s}$ could be achieved. A 50 percent improvement in the variability of clock synchronization from stratum to stratum would reduce this number to less than $125\mu\text{s}$. This does not imply that stratum-to-stratum accuracy should not be improved. On the contrary, by working on the accuracy and variability together, all users of time synchronization will benefit. This needs to be accomplished without overloading the computer or the network.

A 50 percent improvement appears to be a reasonable goal; however, if the stratum-to-stratum synchronization variability could be improved by an order of magnitude, it is reasonable to anticipate maximum theoretical time synchronization offsets of $50\mu\text{s}$ or less in a well-behaved LAN and potentially in the hundreds of microseconds for a Wide Area Network.

6.2. Analysis of Published Results

In Dr. Mills' second edition of *Computer Network Time Synchronization* [1, pages 323-327], results are reported for two sample networks: a backroom LAN (a 100Mb/s switched Ethernet with very little traffic) and the campus LAN (a 100Mb/s switched Ethernet

with two very busy servers and NTP traffic volume of well over 1000 packets per second). For the backroom LAN, the measured offset for two identical machines from the GPS server was ~20us (operating in Client/Server mode) while the offset between machines operating in Interleaved Symmetric mode was less than 5us with a round-trip delay of roughly 100us. These machines were also operating in Interleaved Broadcast mode to support a third machine on the network with time synchronization offset averaging 40us or less and a round-trip delay of roughly 200us. In the case of Interleaved Broadcast, the experiments we performed showed roughly a 10 microsecond offset while the backroom LAN in Dr. Mills' book was ~3x greater. This difference in offset could be caused by differences in the size of the Protocol Data Units (PDUs) sent (since the backroom LAN used Autokey for all interleaved operations), the clients' processing power, and the client and server operating systems.

In the case of the campus LAN, one of dozens of subnets was used for testing purposes. This subnet contained two busy NTP servers and two test hosts dedicated to the experiment. The NTP servers were again synced with each other via Interleaved Symmetric mode while the test hosts were synced to one of the servers using the Client/Server mode and to each other using Interleaved Symmetric mode. The round-trip delay between the NTP servers was measured at roughly 600us and the offset for each was ~180us but of opposite sign, indicating an asymmetric path between machines. The test machines were able to maintain ~20us offset to the NTP server in Client/Server mode and less than 40us offset between each other with a round-trip delay on the order of 300us using Interleaved Symmetric mode. The heavy network traffic appears to have increased the round-trip delay by ~3x over the backroom LAN with roughly an 8x change in offset measurement. This emphasizes the need for standardized test methods that are necessary for estimating accurate improvements in network time synchronization.

7. Future Experimentation

While the results so far are good, further experiments are needed to draw conclusions. One concern is that the clock offsets are in the microsecond range. The use of ntpdate may not be a valid way to accurately measure clock offsets at this level since ntpdate makes measurements across the network and is susceptible to errors caused by variations in network delay. Further work is needed to ensure valid offset measurements. An out-of-band measurement technique, which is not affected by variations in network delay, needs to be investigated for use in future experiments.

With the results published by Dr. Mills on the campus LAN, we now have information on Interleaved Symmetric when put under load both from a processor and network perspective, and Interleaved Broadcast when put under a network load. Additional experiments should be

performed to measure and characterize the resiliency of the NTP interleaved modes while under network and processor load and then make comparisons to the other time synchronization methods. The experiments that have been run up to now had the test workstations connected to the same network switch. Future experiments should be conducted to determine how performance is affected when a more complex network configuration is used.

The experiments conducted so far have provided data concerning various interleaved modes of operation in a few configurations; however, there is no thorough, systematic set of results that would help in deciding whether to run one NTP mode versus another in real systems. Still needed is guidance on when to use the various NTP mode options and what the advantages and disadvantages are in using the various modes. For example, when a time server is available that is directly connected to a satellite time receiver, what are the relative tradeoffs in using either Client/Server or Interleaved Broadcast with other clients? In both of Dr. Mills' experiments, Interleaved Symmetric mode was used to obtain additional measurement data, but can this mode be used to gain higher time accuracy among peers? A set of best-use scenarios would also be very helpful.

8. Security Considerations

Security aspects of the mechanisms described is a major concern and will need to be considered in more detail.

9. Internet Assigned Numbers Authority (IANA) Considerations

No IANA actions are required as a result of the publication of this document.

10. Conclusions

Results from our experiments and those described in Dr. Mills' book [1] indicate that an interleaved capability can provide a modest improvement to the time accuracy achieved with NTPv4. A stronger reason to standardize NTP interleaved may be to achieve better performance when large PDUs are used (e.g., providing server authentication). Initial results also indicate that interleaved modes will not provide accuracies in the range that PTP with hardware assists can; thus, the other two mechanisms described in this paper should be pursued using experimentation in parallel with any action to standardize NTP interleaved.

It would be of benefit to the IETF TICTOC Working Group to standardize the performance metrics and/or benchmark methodology for use in describing the behavior and testing of devices for clock synchronization. Such a standard would enable better comparisons between considered mechanisms. In addition to the same definitions

of the metrics used, better agreement should be obtained in experiments being performed by different organizations.

The authors are interested in contributions on the mechanisms described in this draft as well as on additional mechanisms that may improve the accuracy of computer clocks synchronized over a network. The authors request that other experimental results on mechanisms that can improve NTPv4 accuracy be shared. It is hoped that discussions on this topic in the IETF TICTOC Working Group will lead to standardization actions to enable better accuracy to those utilizing a future NTP specification.

11. Informative References

- [1] Mills, David, L., 2011, "Network Time Synchronization-the Network Time Protocol on Earth and in Space, Second Edition", CRC Press
- [2] Mills, David, 8 June 2011, forwarded (by Karen O'Donoghue) email on the TICTOC (and NTP)Discussion Archive, <http://www.ietf.org/mail-archive/web/tictoc/current/msg00945.html>
- [3] Mills, David, 10 June 2011, email on the TICTOC (and NTP)Discussion Archive, <http://www.ietf.org/mail-archive/web/tictoc/current/msg00952.html>
- [4] O'Donoghue, K., Glass, M., and Plunkett, T. (2008). "Next Steps In Network Time Synchronization for Navy Shipboard Applications" [Electronic Version]. 40th Annual Precise Time and Time Interval (PTTI) Meeting, pp. 187-195.

12. Acknowledgments

The authors wish to thank Karen O'Donoghue (Internet Society) for her valuable comments.

Approved for public release. Distribution is unlimited.

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

Authors' Addresses

David Marlow
NSWCDD
17214 Avenue B Suite 126
Dahlgren, VA 22448-5147
USA

Email: david.marlow@navy.mil

Sterling Knickerbocker, PhD
NSWCDD
18372 Frontage Road, Suite 318
Dahlgren, VA 22448
USA
Email: sterling.knickerbocker@navy.mil

Timothy Plunkett
NSWCDD
17214 Avenue B Suite 126
Dahlgren, VA 22448-5147
USA
Email: timothy.plunkett@navy.mil

TICTOC Working Group
Internet Draft
Intended status: Informational
Expires: April 2012

Tal Mizrahi
Marvell
Karen O'Donoghue
ISOC
October 24, 2011

TICTOC Security Requirements
draft-mizrahi-tictoc-security-requirements-00.txt

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/lid-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 Proventionation 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

4.6. Performance	11
4.7. Confidentiality.....	11
4.8. Protection against packet delay attacks	12
5. Summary of Requirements	12
6. Additional security implications	13
7. Issues for Further Discussion	13
8. Security Considerations	14
9. IANA Considerations	14
10. Acknowledgments	14
11. References	14
11.1. Normative References	14
11.2. Informative References	15

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)

(4) 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

BC	Boundary Clock
MITM	Man In The Middle
NTP	Network Time Protocol
OC	Ordinary Clock
PTP	Precision Time Protocol
TC	Transparent Clock

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 Rouge 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:

- o Identification: verifying the identity of the peer clock.
- o 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 Proventionation of Masters

Requirement

The security mechanism MUST support a proventionation 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 proventionation.

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 than 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:

- o Verifies its integrity.
- o Modifies the packet, i.e., modifies the correctionField.
- o 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

Section	Requirement	Type
4.1.	Authentication of sender.	MUST
	Proventication.	MUST
	Authentication of slaves.	SHOULD
	PTP: Authentication of TCs.	SHOULD
	PTP: Authentication of Announce messages.	SHOULD
4.2.	Integrity protection.	MUST
	PTP: hop-by-hop integrity protection.	MUST
	PTP: end-to-end integrity protection.	SHOULD

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.

- o What external security practices impact the security and performance of time keeping? (and what can be done to mitigate these impacts?)
- o What are the security impacts of time synchronization protocol practices? (e.g. on-the-fly modification of timestamps)
- o 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

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [NTPv4] Mills, D., Delaware, U., Martin, J., Burbank, J., Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [AutoKey] Haberman, B., Mills, D., "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.
- [Traps] Treytl, A., Gaderer, G., Hirschler, B., Cohen, R., "Traps and pitfalls in secure clock synchronization" in Proceedings of 2007 International Symposium for Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2007, pp. 18-24, 2007.

11.2. Informative References

- [IEEE 1588] IEEE TC 9 Test and Measurement Society 2000, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

D. Mills
University of Delaware
K. O'Donoghue, Ed.
Internet Society
D. Hart

H. Stenn
Network Time Foundation, Inc.
October 31, 2011

Control Messages Protocol for Use with Network Time Protocol Version 4
draft-odonoghue-ntp4-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.

Table of Contents

- 1. Introduction 4
- 2. NTP Control Message Format 6
- 3. Status Words 8
 - 3.1. System Status Word 8
 - 3.2. Peer Status Word 10
 - 3.3. Clock Status Word 12
 - 3.4. Error Status Word 13
- 4. Commands 14
- 5. IANA Considerations 17
- 6. Security Considerations 17
- 7. Acknowledgements 17
- 8. References 17
 - 8.1. Normative References 17
 - 8.2. Informative References 17
- Authors' Addresses 18

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.

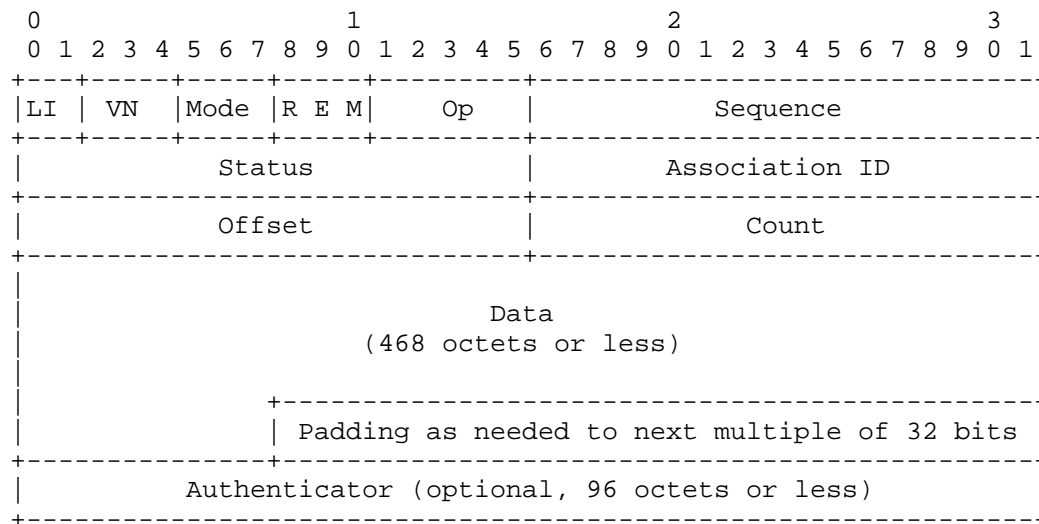
Some exception events, such as when a peer becomes reachable or unreachable, occur spontaneously and are not necessarily associated with a command. An implementation may elect to save the event information for later retrieval or to send an asynchronous response (called a trap) or both. In case of a trap 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.



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:

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

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.)

LI	Meaning
00	no warning
01	insert second after 23:59:59 of the current day
10	delete second 23:59:59 of the current day
11	unsynchronized

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

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

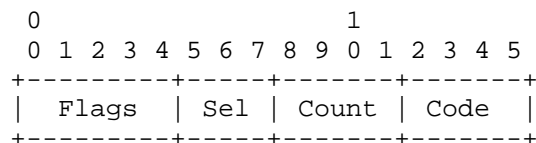
System Event Counter: This is a four-bit integer indicating the number of system events occurring since the last time the System Event Code changed. Upon reaching 15, subsequent events with the same code are not counted.

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:

Code	Meaning
0	unspecified
1	frequency correction (drift) file not available
2	frequency correction started (frequency stepped)
3	spike detected and ignored, starting stepout timer
4	frequency training started
5	clock synchronized
6	system restart
7	panic stop (required step greater than panic threshold)
8	no system peer
9	leap second insertion/deletion armed for end of current month
10	leap second disarmed
11	leap second inserted or deleted
12	clock stepped (stepout timer expired)
13	kernel loop discipline status changed
14	leapseconds table loaded from file
15	leapseconds table outdated, updated file needed

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:



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 Status Flag Bit	Value	Meaning
0	0x8000	configured (peer.config)
1	0x4000	authentication enabled (peer.authenable)
2	0x2000	authentication okay (peer.authentic)
3	0x1000	reachable (peer.reach != 0)
4	0x0800	broadcast association

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:

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

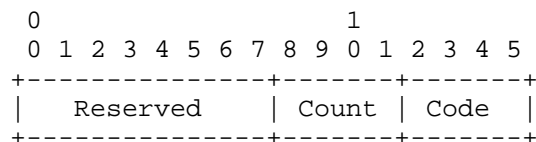
Peer Event Counter: This is a four-bit integer indicating the number of peer events that occurred since the last time the peer event code changed. Upon reaching 15, subsequent events with the same code are not counted.

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:

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

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:



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

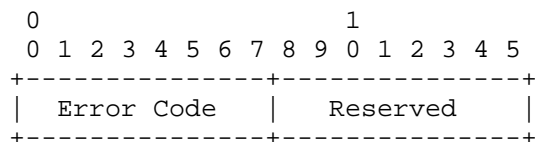
code changed. Upon reaching 15, subsequent events with the same code are not counted.

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

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

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:



Error code: an eight-bit integer coded as follows:

Error Status	Meaning
0	unspecified
1	authentication failure
2	invalid message length or format
3	invalid opcode
4	unknown association identifier
5	unknown variable name
6	invalid variable value
7	administratively prohibited
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

- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

8.2. Informative References

- [RFC5906] Haberman, B. and D. Mills, "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.
- [RFC5907] Gerstung, H., Elliott, C., and B. Haberman, "Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)", RFC 5907, June 2010.
- [RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, June 2010.

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

TICTOC
Internet-Draft
Intended status: Standards Track
Expires: March 19, 2012

Y. Xu
Huawei Technologies
September 16, 2011

IPsec security for packet based synchronization
draft-xu-tictoc-ipsec-security-for-synchronization-02.txt

Abstract

Cellular networks often use Internet standard technologies to handle synchronization. This document defines an extension based on WESP. Usually, several traffic flows are carried in one IPsec tunnel, for some applications, such as, 1588 or NTP, the packets need to be identified after IPsec encryption to handle specially. In order to achieve high scalability in implement, a separate IPsec tunnel will not be established for some special traffic. This document analyses the need for security methods for synchronization messages distributed over the Internet. This document also gives a solution on how to mark the synchronization message when IPsec is implemented in end to end frequency synchronization."

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 January 6, 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.

Table of Contents

- 1. Introduction 4
- 2. Terminology used in this document 6
- 3. Security requirements for synchronization 6
- 4. Security mechanism for synchronization 6
- 5. The extension of WESP 8
 - 5.1. Existing WESP format 8
 - 5.2. Extended WESP format 9
 - 5.3. Authentication field 11
- 6. Example 13
- 7. IPv4/v6 consideration for IPsec based sychronization 14
- 8. Security Considerations 14
- 9. IANA Considerations 14
- 10. Acknowledgments 15
- 11. References 15
 - 11.1. Normative References 15
 - 11.2. Informative References 15
- Author's Address 15

1. Introduction

When transferring timing in internet, a shared infrastructure is used, and hence the path is no longer physically deterministic. It leaves open the possibility to disrupt, corrupt or even spoof the timing flow, where a timing signal purports to come from a higher quality clock than it actually does. In the extreme, this may be used to attack the integrity of the network, to disrupt the synchronization flow, or cause authentication failures. On the other hand, it may be possible for unauthorized users to request service from a clock server. This may overload a clock server and compromise its ability to deliver timing to authorized users.

For the cellular backhaul applications, two kinds of synchronization are needed, one is the recovery of an accurate and stable frequency synchronization signal as a reference for the radio signal (e.g. GSM, UMTS FDD, LTE FDD). In addition to frequency synchronization, phase/time synchronization are also needed in Mobile technologies, This is the case for the TDD technologies such as UMTS TDD, LTE TDD.

Frequency synchronization is normally implemented in an end-to-end scenario where none of the intermediate nodes in the network have to recognize and process the synchronization packets. However In phase/time synchronization, a hop-by-hop scenario will request intermediate nodes to process the synchronization packets If very accurate phase/time is needed (e.g. sub-microsecond accuracy).

Femtocell is the typical cellular backhaul application that requires time synchronization. A Femtocell is defined as a wireless base station for deployment in residential environments and is typically connected to the mobile core network via a public broadband connection (eg., DSL modem, cable modem). Femtocell improves cellular network coverage and saves cost for operators. Just like a typical macrocell (larger base station), a Femtocell (residential base station) requires a certain level of synchronization (frequency or phase/time) on the air interface, predominantly frequency requirements.

The [3GPP.33.320] specification defines some of the high-level network architecture aspects of a Home NodeB (3G UMTS) and a Home eNodeB (4G LTE). In addition, the Femto Forum organization also provides a network reference model very similar to 3GPP. Both architectures have commonalities as illustrated in Figure 1.

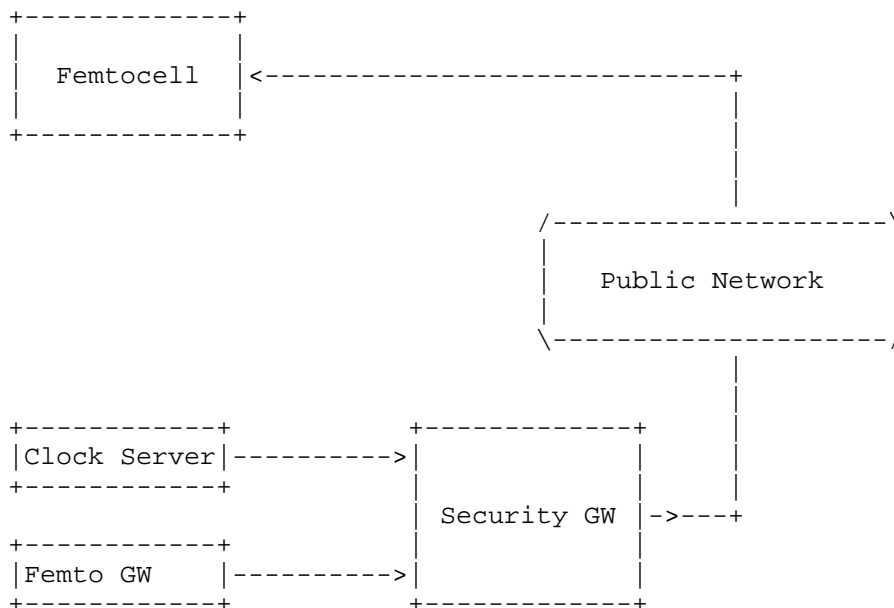


Figure 1. Typical Architecture of a Femtocell Network

The network architecture shows that a public network is used to establish connectivity between Femtocell and core network elements (e.g., Security Gateway, Femto Gateway, Clock server, etc.). With respect to synchronization process, Femtocell will therefore see synchronization messages exchanged over the public network (e.g, Internet). This presents a set of unique challenges for mobile operators.

One challenge involves the security aspects of such the Femto architecture. In both reference models, the communication between Femtocell and Femto Gateway is secured by a mandatory Security Gateway function. The Security Gateway is mandatory since the Femto Gateway and Clock server communicate to Femtocell via a public backhaul broadband connection (also known as the 3GPP iuh interface or Femto Forum Fa interface). The [3GPP.33.320] specification requires that the Femtocell SHALL support receiving time synchronization messages over the secure backhaul link between Femtocell and the Security Gateway, and Femtocell SHALL use IKEv2 protocol to set up at least one IPsec tunnel to protect the traffic with Security Gateway.

This document provides analysis on security requirements for packet-based synchronization and proposes IPsec security solution for end to end frequency synchronization.

2. Terminology 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 [RFC2119].

3. Security requirements for synchronization

The ITUT [G.8265] specification provides general consideration on synchronization security. Because packet-based timing streams may be observed at different points in the network, there may be cases where timing packets flow across multiple network domains which may introduce specific security requirements. There may also be aspects of security that may be related to both the network (e.g. authentication and/or authorization) and to the synchronization protocol itself. ITUT [G.8265] specification recommends to use existing, standards-based security techniques to help ensure the integrity of the synchronization. Examples may include encryption and/or authentication techniques, or network techniques for separating traffic, such as VLANs or LSPs. Specifically for the performance issue, it may not be possible to implement some security requirements without actually degrading the overall level of timing or system performance. From above analysis, following synchronizations requirements are listed:

1. synchronization client SHOULD be prevented from connecting to rogue clock servers
2. clock servers SHOULD be prevented from providing service to unauthorized synchronization client
3. Security mechanisms to achieve synchronization SHOULD minimize any degradation in performance and this side effect SHOULD be controlled to meet specific synchronization requirements(e.g., Femtocell synchronization)

4. Security mechanism for synchronization

There are mainly two kinds of security mechanism used in current synchronization: authentication-based and encryption-based.

For the authentication-based security mechanism, a shared secret key between the synchronization client and the clock servers is used to compute an authentication code (known as an "Integrity Check Value",

ICV) over the entire message datagram. [IEEE1588] contains an experimental security annex defining an authentication-based approach. This approach also implements a challenge-response mechanism to confirm the creation of any security association (SA) between a clock servers and a synchronization client. A limitation of the process is that no method of sharing the key is proposed in [IEEE1588]. This MUST be handled by other means.

For the encryption-based security mechanism, a shared-key approach is also used. Instead of creating an ICV, the shared key is used to encrypt the contents of the packet completely. The encryption might be performed in the synchronization device itself, or it might be performed in a separate device, e.g. a secure gateway. An example might be where the timing packets have to pass through an encrypted tunnel (e.g. an IPsec tunnel). Full encryption might be required for various reasons. The contents of the packet may be considered secret, such as might be the case where accuracy of the time distribution is being sold as a service. Alternatively, it may be because other traffic from a device is considered secret, and hence it is easier to encrypt all traffic.

IPsec, as a popular security mechanism, is being considered in some mobile applications, especially in case of unsecure backhaul links (e.g. Femtocells, [3GPP.33.320]) being involved. IPsec can provide data source authentication, confidentiality, integrity that is suitable to end to end synchronization without intermediate nodes. It provides security services by Authentication header (AH) and Encapsulating security payload (ESP). Authentication Header provides integrity protection and data origin authentication. Moreover, ESP can be used to provide confidentiality besides data origin authentication, connectionless integrity. For the time packet protection, the critical issue is the precision of the timestamps. That is the receiver must mark the time as soon as possible when taking over the time packet, and the time will be used for frequency synchronization. And in the implementation, an IPsec tunnel is created to carry all the traffic between the IPsec end points considering the cost of IPsec SA establishment, i.e., this IPsec tunnel will be used to protect both the service traffic packets and time packets. Therefore, for protect against active and passive attack, confidentiality and integrity will be configured when deploying IPsec processing policy. But nodes cannot recognize 1588 packets as defined in [IEEE1588] as the port is encrypted by IPsec. It becomes complicated when processing IPsec packets as the nodes will not be able to identify the 1588 packets that need to be time stamped any more. This document describes a method to resolve this problem. For time packets, some identifiers that can be used to recognize all such packet at the physical layer are defined in WESP, and all of these are provided with data integrity protection. For

example, if only frequency synchronization is needed, an end-to-end scenario where none of the intermediate nodes in the network have to recognise and process the synchronization packets might be suitable to use IPsec security mechanism. In this case, the synchronization packets will be encrypted if the packet is transported in the IPsec tunnel.

IPsec can meet synchronization requirement 1 and 2 in section 3. However IPsec still need some enhancement to meet requirement 3. Normally, device will decrypt IPsec message in IP layer, but in order to improve the synchronization accuracy, some synchronization protocol (e.g. [IEEE1588]) requests to process the synchronization message in hardware, therefore the synchronization device may need to identify synchronization messages in physical layer before the message is decrypted. How to identify the synchronization messages in IPsec becomes the most important issue to keep the synchronization accuracy in IPsec synchronization scenario.

5. The extension of WESP

As discussed above section, it has advantage to identify whether the tunnel packets received by synchronization client are the special timing packets or not. This section proposes a solution to identify the timing packets when using IPsec to protect the whole time synchronization message. The main thought is to use time packet identifier which is included in the WESP format to identify whether the received data packet is a timing packet or not.

5.1. Existing WESP format

[RFC5840] describes an encapsulating ESP, i.e., WESP, and affords an extension for ESP. This document applies WESP to provide a mechanism to identify time packet within an IPsec tunnel, the IPsec endpoints could distinguish the time packet and do the corresponding synchronization processing.

The WESP format is as follows:

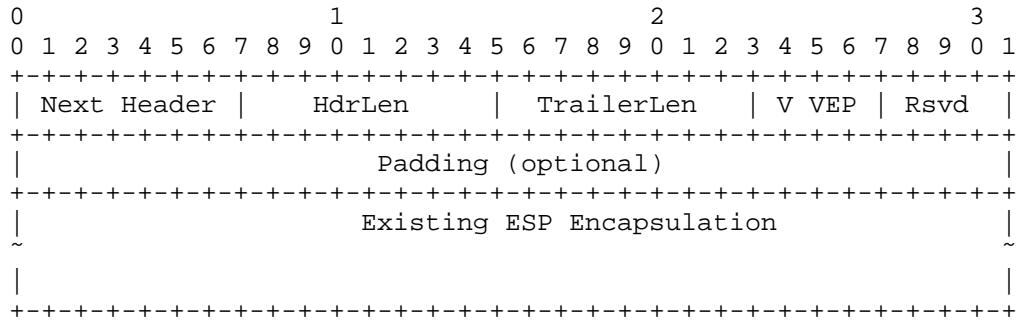


Figure 2. Format of an WESP Packet

These fields are introduced with the extended WESP format in next section.

5.2. Extended WESP format

This document describes the extension for the WESP for the additional application. It allows the ESP receiver or intermediate node not only distinguish encrypted and unencrypted traffic, but also identify whether the encrypted packets are the common packets or the time packets.

The extension format is depicted as follows:

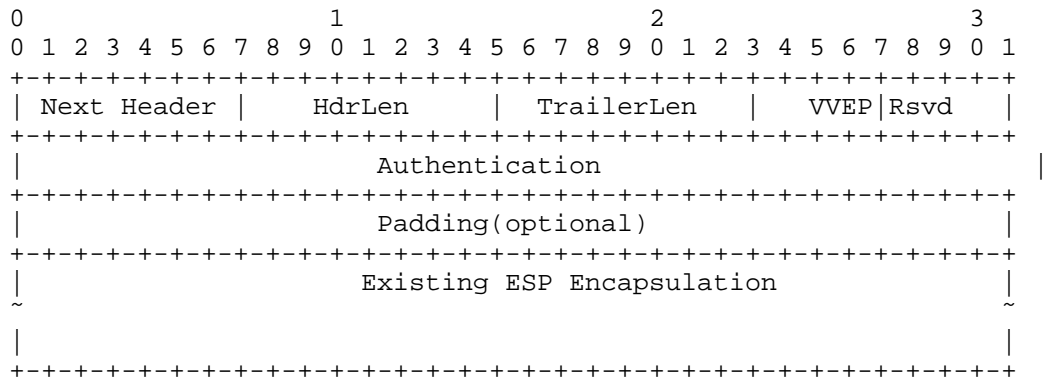


Figure 3. The extended WESP format

The definitions of these fields are as follows:

- o Next Header is identical with the definition in [RFC5840]. It MUST be the same as the Next Header field in the ESP trailer when using ESP in the Integrity-only mode. When using ESP with encryption, the "Next Header" field loses this name and semantics and becomes an empty field that MUST be initialized to all zeros. The receiver MUST ensure that the Next Header field in the WESP header is an empty field initialized to zero if using ESP with encryption.
- o HdrLen is identical with the definition in [RFC5840]. It is the offset from the beginning of the WESP header to the beginning of the Rest of Payload Data (i.e., past the IV, if present and any other WESP options defined in the future) within the encapsulated ESP header, in octets. HdrLen MUST be set to zero when using ESP with encryption.
- o TrailerLen contains the size of the Integrity Check Value (ICV) being used by the negotiated algorithms within the IPsec SA. TrailerLen MUST be set to zero when using ESP with encryption. One issue must be taken into account that if using ESP with encryption, TrailerLen has lost the significance of ICV, as any attacker could juggle the field definition above, Next Header, HdrLen, TrailerLen to zero, and forward the modified packet to the receiver. The receiver will deal with the dummy encrypted packet falsely.
- o Authentication contains extended data type, extended data length, the optional Algorithm ID field and extended data and ICV when using ESP with encryption. This part will be depicted in next section.
- o Flags: The bits are defined most-significant-bit (MSB) first, so bit 0 is the most significant bit of the flags octet. The four bits "Rsvd" are used for the future, the least significant bit of the four bit to indicate the some extended information is included when using ESP not only integrity but also with encryption, i.e., if the least significant bit is set to one, the corresponding extended information will be contained in Authentication payload.

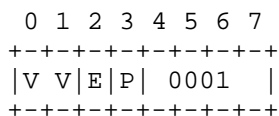


Figure 4: Flags Format

The definitions of each specific field in flags is as follows:

- o Version (V): It requires the new version number, and MUST be sent as 0 and checked by the receiver.

- o Encrypted Payload (E): Setting the Encrypted Payload bit to 1 indicates that the WESP (and therefore ESP) payload is protected with encryption. If this bit is set to 0, then the payload is using integrity-only ESP.
- o Padding header (P), 1 bit: If set (value 1), the 4-octet padding is present. If not set (value 0), the 4-octet padding is absent. The alignment requirement must be guarantee as defined in [RFC5840].
- o Rsvd, 4 bits: Reserved for future use. The reserved bits MUST checked whether the least significant bit is set as 0 or 1. If setting with 0, it will be ignored by the receiver. If setting with 1, the receiver will check the correction by ICV, either TrailerLen using ESP without encryption or Authentication when using ESP with encryption.

5.3. Authentication field

The Authentication field is comprised of extended data type, extended data length, the optional Algorithm ID field and extended data and ICV when using ESP with encryption. The extended data type indicates the packet type. When the type is time packets, it could identify whether the time packet is the event message or not. In addition, ICV parts offer the authentication of data integrity for the whole extended Data is provided.

The figure of the proposed flexible ESP format is as following:

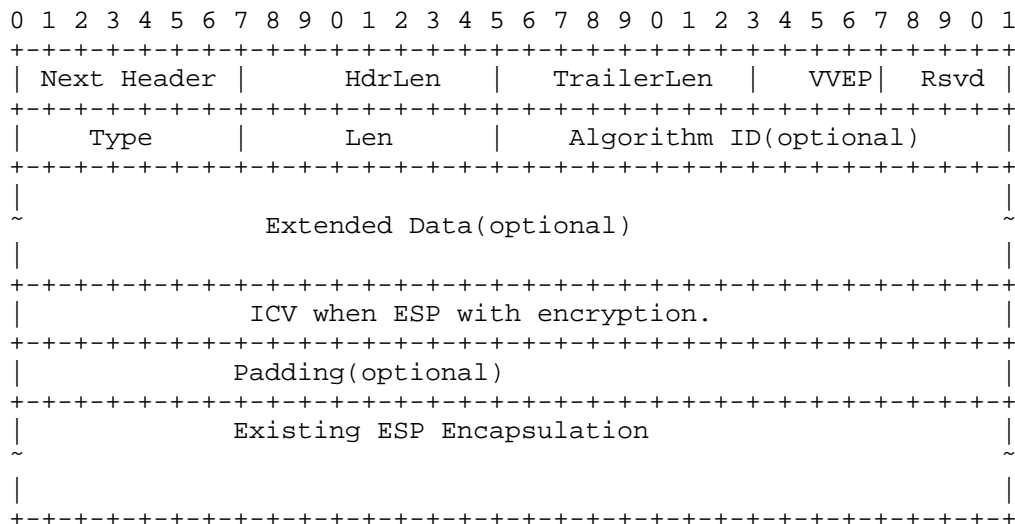


Figure 5. The detailed WESP format

In Femtocell scenario, as the link between Security Gateway and clock server is normally security path, the message transmitted between them are in plain text. When Security Gateway receives the message, it identifies the time packet at first, then put appropriate value to Data type field to identify the message type in Payload Data. After that, it could put more packet information into Extended Data Payload, such as UDP port number or timestamps, then Extended Data Length, Algorithm ID, Extended Data integrity Check value (Figure 4), could also be filled consequently. The following figure illustrates on how to use this new flexible ESP format to identify time packet.

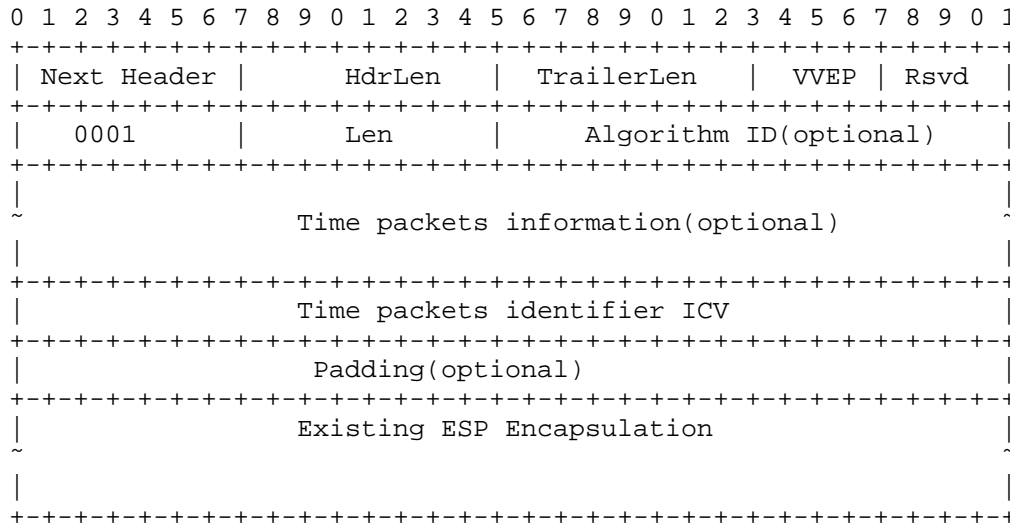


Figure 6. WESP format for time-packet

- o type (8-bit) - The value 0x1 here indicates that the extended context is time packet.
- o Length (16-bit)- The length of whole extended additional authentication data
- o Time packets information(variable)- the addintional message information, such as UDP port number or timestamps. It is a part of Authentication payload.
- o Algorithm ID- It indicates which algorithm could be used to generate the extended data ICV. It is a part of Authentication payload.The integrity algorithm negotiated during IKEv2 could be used, also Algorithm ID field in the extended additional

authentication data could be marked to indicate the integrity algorithm, such as HMAC-SHA1, HMAC-256, or others. It is a part of Authentication payload.

- o Time packets identifier integrity Check value (variable) - Time packets identifier integrity Check value, and used to guarantee the integrity of transmission.

Time packets information, Algorithm ID are the optional fields. As the integrity protection is only for the Extended Data when ESP with encryption but not for the whole ESP packet, the time delay of calculation can be decreased. In addition, if the integrity protection is not necessary, this part of security validation could be ignored.

6. Example

In this section, the procedure to identify time packet in Security Gateway scenario is depicted.

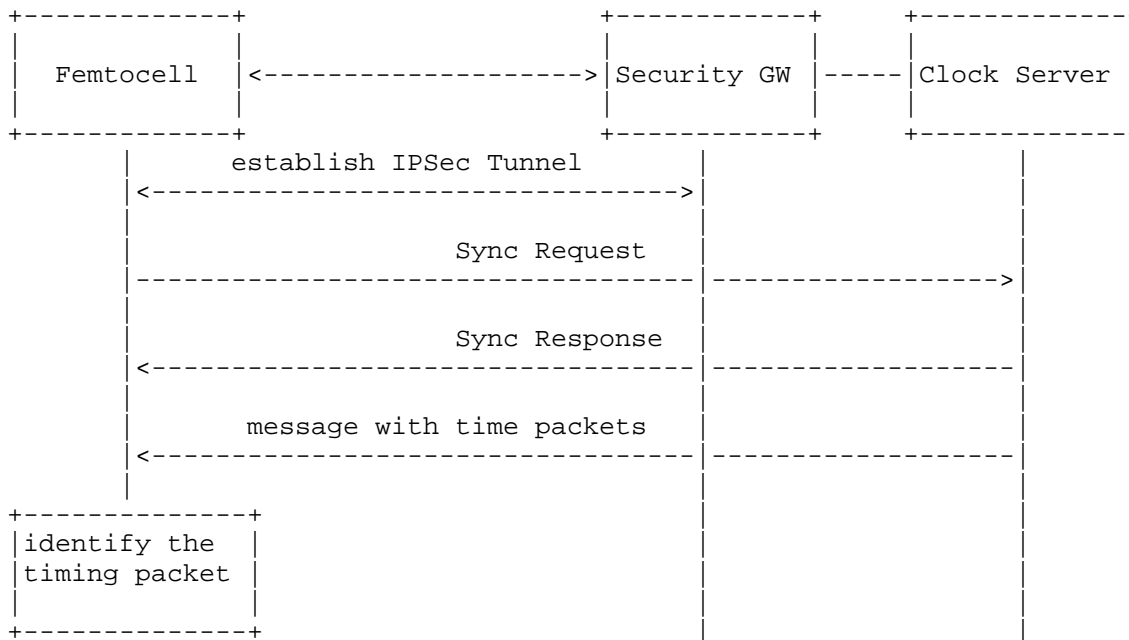


Figure 7. Example Procedure

In the Security Gateway scenario, The IPsec with tunnel mode is

established between Femtocell and Security Gateway. After Femtocell and Clock server exchange the Sync Request and Sync Response, the clock server will send the time packets to Femtocell to implement frequency synchronization with the protection of IPsec tunnel. When Femtocell receives the message, it can identify whether it is time packet, and can also identify whether the time packet is the event message by the time packet information in the unencrypted field as defined in the new ESP format. If the message is time packet and identifies that it is the event message, Femtocell will do special process for the event message, such as recording the message receiving time. On the server side, When Security Gateway receives the message, it identifies the time packet at first, then put appropriate value to Data type field to identify the message type in Payload Data, after that, it could put more packet information into Authentication Payload, such as UDP port number or timestamps, then Extended Data Length, Algorithm ID, Extended Data integrity Check value, could also be filled consequently.

7. IPv4/v6 consideration for IPsec based sychronization

IPsec is a security mechanism used both for IPv4 and IPv6, and WESP-based solution has no impact on the IPv4 header and makes the transition/migration from IPv4 to IPv6 seamless.

8. Security Considerations

This protocol variation inherits all the security properties of regular ESP as described in [RFC4303].

This document describes the modification or extension for the WESP for the additional application. The approach described in this document requires the ESP endpoints to be modified to support the new protocol. It allows the ESP receiver or intermediate node not only to distinguish encrypted and unencrypted traffic deterministically, but also identify whether the encrypted packets are the common packets or the time packets by a simpler implementation for the transport node.

Note that whether the time packets identified by the defined mark or tag are transparent or not, there is always a possibility for attackers to employ interception attacks to block transmission. How to prevent interception attack is out of scope of this draft.

9. IANA Considerations

There have been no IANA considerations so far in this document.

10. Acknowledgments

The authors appreciate the valuable work and contribution done to this document by Marcus Wong.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC5840] Grewal, K., Montenegro, G., and M. Bhatia, "Wrapped Encapsulating Security Payload (ESP) for Traffic Visibility", RFC 5840, April 2010.

11.2. Informative References

- [3GPP.33.320] 3GPP, "Security of Home Node B (HNB) / Home evolved Node B (HeNB)", 3GPP TS 33.320 10.3.0, June 2011.
- [G.8265] IEEE, "Architecture and requirements for packet based frequency delivery", V0.2 June 2010.
- [IEEE1588] IEEE, "Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008.

Author's Address

Yixian Xu
Huawei Technologies
Huawei Building, Xinxu Road No.3
Haidian District, Beijing 100085
P. R. China

Phone: +86-10-82836300
Email: xuyixian@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2012

J. Zhang
L. Xia
ZTE Corporation
Oct 20, 2011

PDV-based PTP LSP Setup, Reoptimization and Recovery
draft-zhang-tictoc-pdv-lsp-00

Abstract

This document defines a mechanism for the setup, reoptimization and recovery of PTP LSP based on the PDV metrics between the 1588 Master and the 1588 Slave.

When a PTP communication path goes through the third party networks (e.g. the MPLS networks), the PDV noise caused by the third party networks will have a significant impact on the synchronization performance. So, the PDV metrics should be considered in the setup of PTP LSP.

In addition, when the PDV noise exceeds to a certain degree, it is necessary to notify the head-end LSR (i.e. the 1588 Master) to switch to the backup PTP LSP and to reoptimize the primary PTP LSP in order to improve the PTP reliability.

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 22, 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
1.1.	Conventions Used in This Document	3
2.	Terminology	3
3.	Problem Statement	4
4.	PTP LSP setup and reoptimization	5
4.1.	Advertisement of the capability of reserving bandwidth	6
4.2.	PTP LSP setup	6
4.3.	Congestion detection	7
4.4.	PTP LSP reoptimization	7
5.	PTP LSP recovery mechanisms	8
5.1.	PDV measurement and PDV network limits	8
5.2.	PTP LSP recovery	8
5.3.	PTP Master recovery	9
6.	Protocal extensions	11
6.1.	IGP extensions	11
6.2.	RSVP-TE extensions	13
7.	Other considerations	13
8.	Security Considerations	14
9.	Acknowledgements	14
10.	IANA Considerations	14
10.1.	IANA Considerations for OSPF	14
10.2.	IANA Considerations for IS-IS	14
10.3.	IANA Considerations for RSVP	15
11.	References	15
11.1.	Normative References	15
11.2.	Informative References	15
	Authors' Addresses	16

1. Introduction

There are many applications that need frequency or phase/time synchronization, and there is an emerging need to distribute highly accurate time and frequency information over IP and over MPLS packet switched networks (PSNs), especially with the development of the telecom network. [IEEE] defines PTP for clock and time synchronization. PTP version 2 contains three clock type, they are Ordinary Clock(OC), Boundary Clock(BC) and Transparent Clock(TC). Transparent Clocks modify a "correction field" (CF) within the synchronization messages to compensate for residence and propagation delays. So, Transparent Clock can eliminate the impact of the PDV noise.

With the large-scale deployment of the MPLS networks and the 1588 networks, there is an increasing need to transport PTP messages over the MPLS networks. The MPLS networks could be a transit network between the 1588 Master and the 1588 Slave. But the PDV noise between the 1588 Master and the 1588 Slave may be excessive and therefore the 1588 Slave may not be able to properly recover the clock and time of day. Therefore, it is necessary to setup PTP LSP based on the PDV attributes, and when the PDV noise exceeds a certain degree, the 1588 Master will switch to the backup PTP LSP and reoptimize the primary PTP LSP.

1.1. 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 [RFC2119].

2. Terminology

PTN: Packet Transport Network;

1588: The timing and synchronization as defined by IEEE 1588;

PTP: The timing and synchronization protocol used by 1588;

Master: The Source of 1588 Timing and clock. This will be a port in master state on a Grandmaster Clock or on a Boundary Clock;

Slave: The Destination of 1588 Timing and clock that tries to follow the Master clock. This will be a port in slave state on a boundary clock or on a Slave-Only Ordinary Clock;

OC: Ordinary Clock - a device with a single PTP port;

TC: Transparent Clock, a time stamping method applied by intermediate nodes between Master and Slave;

BC: Boundary Clock, is a node that recovers the Master clock via a Slave function and uses that clock as the Master for other Slaves;

PDV: Packet Delay Variation;

PTP LSP: An LSP dedicated to carry PTP messages;

PDV PTP LSP: An PTP LSP based on the PDV attributes;

ACR: Adaptive Clock Recovery;

MBB: make-before-break;

LSR: Label Switch Router;

3. Problem Statement

With the development of telecom networks, there is an increasing need to transport PTP or CES over the third part networks(e.g. MPLS networks). Two main applications are addressed in ITU-T G.8261.1:

(1) the distribution of a synchronization network clock signal via packet based method (e.g. using PTP);

(2) the distribution of a service clock signal over a packet network according to the ACR method (e.g. clock recovery of CES using Adaptive Method). The packet networks are Ethernet, MPLS, T-MPLS or IP. For these applications, frequency synchronization information is carried via packets and is recovered according to adaptive clock recovery(ACR) method. But the third part networks(e.g. MPLS networks) may introduce the PDV noise which will have a significant impact on the ACR Methods and the synchronization performance.

The method for transporting PTP messages (PDUs) over an MPLS network is defined in [I-D.ietf-tictoc-1588overmpls]. This document defines a "1588-aware LSR" that is able to identify 1588 timing flows carried over MPLS. Transparent Clock (TC) function requires a 1588-aware LSR in the middle of an LSP to properly handle the PTP messages. However, this specification does not mandate that all LSRs in path of a PTP LSP be 1588-aware, Non-1588-aware LSRs don't perform any TC processing. Therefore, these LSRs may introduce additional PDV

noise, although the PTP messages are treated with the highest priority and Green for drop eligibility, because the other flows may use the same queue.

Just as MPLS-TE setup a TE LSP based on TE metrics(e.g. bandwidth), it is necessary to setup a PTP LSP based-on the PDV metrics, and if the PDV noise between the 1588 Master and the 1588 Slave has deteriorated into a certain degree, then the 1588 Master switches to the backup PTP LSP and reoptimizes the primary PTP LSP. So, it is useful for clock recovery algorithms to improve the performance of clock recovery.

4. PTP LSP setup and reoptimization

The PDV noise introduced by the MPLS networks is critical for the clock recovery algorithm and the synchronization performance. The main factor caused the PDV noise is congestion in the network nodes. In order to minimize the PDV noise between the 1588 Master and the 1588 Slave, the 1588 Master SHOULD discover a PTP LSP along which the number of the Non-1588-aware LSRs is minimum.

MPLS-TE support setting up TE-LSP based on the reserved bandwidth which can prevent TE-flow from congestion. But due to the complexity of implementation and the cost of HW, some of the network nodes don't support the capability of reserving bandwidth. In this case, congestion detection MUST be enabled on these nodes. If congestion occurred on one of these nodes, then the node MUST notify the 1588 Master(i.e. the head-end node), therefore the 1588 Master is able to reoptimize the TE-LSP and to avoid the congested nodes so that the PDV noise between the 1588 Master and the 1588 Slave can be minimized.

It is possible that after a PTP LSP has been established, a more efficient path for the PTP becomes available, perhaps because one or more new 1588aware links have been advertised or because some other services have been torn down causing resources in the network to be released. When a better path can be found for one of the previously computed paths, the node or component that originally requested the path can be notified. In order to take advantage of the new path, the ingress node must re-route the PTP onto the new path using the reoptimization technique(e.g. make-before-break).

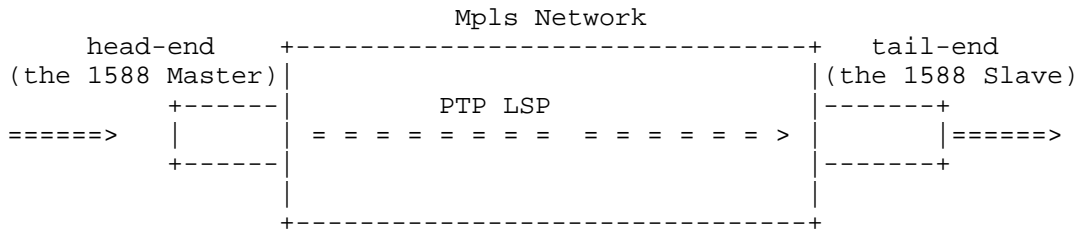


Figure 1. 1588 over MPLS Network

4.1. Advertisement of the capability of reserving bandwidth

Due to the complexity of implement and the cost of HW, some of the network nodes don't support the capability of reserving bandwidth based on the LSP , so that these network nodes may introduce jitter(i.e. PDV).

Just as the capability of 1588-aware which can compensate for residence time by updating the PTP packet Correction Field and eliminate the delay and jitter, it is useful to advertise data plane TE router link capabilities within the whole network, such as the capability of reserving bandwidth. This capability MUST then be taken into account during path computation to prefer links that advertise themselves as reserving bandwidth , so that the PTP LSPs can be properly handled.

For this purpose, the following sections specify extensions to OSPF and IS-IS in order to advertise the capability of reserving bandwidth.

4.2. PTP LSP setup

The Procedures for setting up LSP Tunnels are defined in [RFC3209]. To setup PTP LSP, more constraints information MUST be taken into account, for examples, 1588-aware, reserving bandwidth, and so on. .

After all TE information were advertised by link state protocol(e.g. OSPF or IS-IS), the TE database at the head-end node contains all the links and their characteristics or attributes and includes 1588-aware and reserving bandwidth. From this MPLS TE database, path calculation (PCALC) or constrained SPF (CSPF) calculates the shortest route that still adheres to all the constraints from the head-end LSR(i.e. the 1588 Master) to the tail-end LSR(i.e. the 1588 Slave).

4.3. Congestion detection

Quality of service (QoS) has become popular the past few years. Few networks have unlimited bandwidth, so congestion is always a possibility in the network. QoS is a means to prioritize important traffic over less important traffic and make sure it is delivered. Congestion may happen at different points within a network node. Each congestion point represents a potential source of delay, jitter, and loss for traffic streams.

If neither the capability of 1588-aware nor the capability of reserving bandwidth is supported at a node, then this node may introduce jitter(i.e. PDV), although the LSP tunnel is treated with the highest priority.

So, after PTP LSP has been set up, the head-end node(i.e. the 1588 Master) MUST request those network nodes which support neither 1588-aware nor reserving bandwidth to enable congestion detection. When congestion occurs or disappears on the node, the node MUST send a notification message to the head-end node, so that the head-end node can reoptimizes the PTP LSP and sets up another new PTP LSP which doesn't pass through these congested network nodes.

Because a PTP LSP is related to a special output port and a special priority, the data plane can detect congestion based on the output port and the corresponding queue. When congestion has occurred, the data plane will notify the control plane which will sends a notify message to the head-end.

4.4. PTP LSP reoptimization

As mentioned above, it is possible that a more efficient path for the PTP becomes available, for example, the deployment of new 1588-aware LSRs, and so on. In order to improve the synchronization performance, the head-end MUST re-route the PTP onto the new path. It is assumed that the head-end node has received the congestion notifications from the congested nodes along the PTP LSP, and the PDV noise between the 1588 Master and the 1588 Slave exceeds a certain degree. At this time, the tail-end node(e.g. the 1588 Slave) MUST send a reoptimization notification message to the head-end node, the receipt of such of message will then trigger a reoptimization on the head-end node for the affected PTP LSP. Because the head-end node has learned about which network nodes are 1588-aware and which network nodes has occurred congestion, so it can reoptimize the affected PTP LSP and avoid those congested nodes..

There are several reoptimization triggers, including timer-based reoptimization , event-driven reoptimization and operator-driven

reoptimization. Note that reoptimization or recovery may also introduce the PDV. Therefore, PTP LSP reoptimization MUST be triggered by the PDV metrics.

5. PTP LSP recovery mechanisms

One kind of network fault is packets arriving at a network node which the node has no forwarding information or incorrect forwarding information. This problem can be detected by the control information. Another kind of problem is the one in which the control plane information is correct but the data plane fails. In this case, LSP ping, LSP traceroute and Bidirectional Forwarding Detection (BFD) are tools that can detect problems in the MPLS control and data planes.

The above network problems are all due to connection failure. But for the synchronization application(e.g. PTP), it is tolerant of an occasional missed message, duplicated message, or message that arrived out of order, which means that an occasional connection failure is insignificant to the synchronization performance. However, the PDV by the packet timing signal as it traverses the network from the 1588 Master to the 1588 Slave is critical to the synchronization performance. So, it is reasonable to recover the synchronization application based on the PDV metrics.

5.1. PDV measurement and PDV network limits

ITU-T G.826x concerns frequency synchronization aspects in packet networks. In particular it specifies the Hypothetical Reference Model and the PDV network limits applicable when frequency synchronization is carried via packets and is recovered according to adaptive clock recovery method as defined in G.8261 and G.8260. It specifies the minimum equipment tolerance to packet delay variation in terms of the metrics defined in ITU-T G.8260 at boundary of these packet networks.

PDV measurement is at the input of the 1588 Slave. If the PDV exceed the network limits, then the 1588 Slave MUST send native indications over the PTP LSP to notify the 1588 Master of the PDV fault condition and to recover the synchronization application based on the PDV metrics.

5.2. PTP LSP recovery

The three main components are the 1588 Master, the 1588 Slave and the packet network. A packet timing signal generated by the 1588 Master is transported over the packet network so that the 1588 Slave can

generate a clock frequency traceable to the input timing signal available at the 1588 Master.

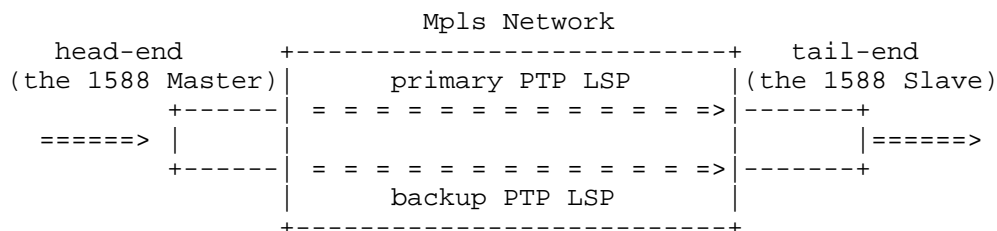


Figure 2. PTP LSP recovery

In this case, there is only a 1588 Master to which the 1588 Slaves are synchronized. It is REQUIRED to set up the primary and the backup PTP LSP, because the PDV metrics is critical to the synchronization performance. This document defines the following functions:

- 1) The head-end node sets up dedicated bidirectional 1+1 path protection which contain the primary and the backup PTP LSP;
- 2) The head-end node and the tail-end all send the Announce message to each other and to establish the synchronization hierarchy.
- 3) The 1588 Master initiates simultaneously the synchronization message exchange over the primary and the backup PTP LSPs. The 1588 Slave obtains the timestamp t1,t2,t3 and t4 which is used for PDV measurement.
- 4) The 1588 Slave compares the PDV performance of primary path with the PDV performance of backup path to determine which PTP LSP should be selected, and the 1588 Slave synchronizes to the PTP LSP which the PDV performance is better.
- 5) If the PDV performance of primary path is exceed the PDV network limits, then the PTP LSP will switch to the backup path and synchronize to it. At the same time, the 1588 Slave sends a notify message to the 1588 Master to reoptimize the primary PTP LSP.

5.3. PTP Master recovery

In traditional synchronization networks, timing availability is enhanced by the use of timing protection where by the timing to a

1588 Slave clock (e.g. SEC, or EEC) may be provided over one or more alternative network paths. In the case of the packet based timing architecture, the 1588 Slave may have visibility to two or more 1588 Masters. 1588 Master protection is stated in In ITU-T G.8265 section 7.2.1.

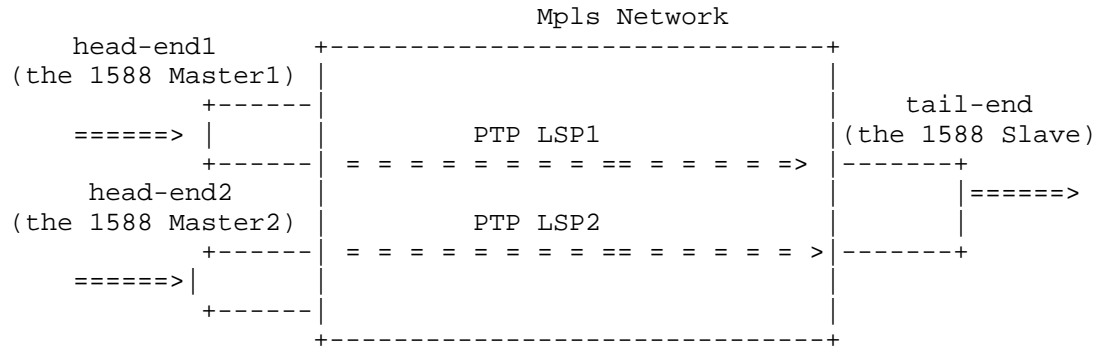


Figure 3. PTP Master recovery

In this case, there are two 1588 Masters to which the 1588 Slave is synchronized. The following details 1588 Master recovery process:

- 1) The primary Master and the backup Master are set up respectively a bidirectional PTP LSP to the 1588 Slave.
- 2) The 1588 Masters and the 1588 Slave all send the Announce message to each other and to establish the synchronization hierarchy.
- 3) The primary Master and the backup Master initiate respectively the synchronization message exchange over the PTP LSP. The 1588 Slave obtains the timestamp t1,t2,t3 and t4 which is used for PDV measurement from the primary Master and the backup Master.
- 4) The 1588 Slave compares the PDV performance of the primary Master with the PDV performance of the backup Master to determine which 1588 Master should be selected. According to ITU-T G.8265.1 6.7.3 "Master selection process", the following parameters contribute to the 1588 Master selection process: (1)Quality Level(QL), (2)Packet Timing Signal Fail(PTSF-lossSync,PTSF-lossAnnounce,PTSF-unusable) and Priority. PTSF-unusable means that the PTP packet timing signal is not usable for the 1588 Slave to achieve the performance target (e.g. violates the 1588 Slave input tolerance because of excessive PDV noise), then a PTSF-unusable associated to this 1588 Master must occur.

6. Protocal extensions

6.1. IGP extensions

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 reserving-bandwidth. This capability MUST then be taken into account during path computation to prefer links that advertise themselves as reserving- bandwidth, so that the PTP LSPs can be properly handled. For this purpose, the following sections specify extensions to OSPF and IS-IS in order to advertise reserving-bandwidth capabilities of a link.

1. reserving-bandwidth 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 reserving-bandwidth capability sub-TLV that can be carried as part of the Link TLV.

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

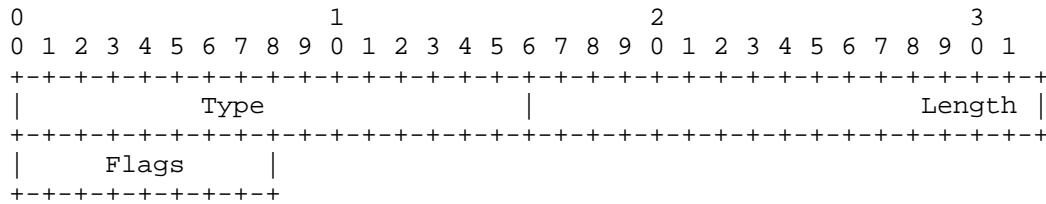


Figure 4: reserving-bandwidth Capability TLV

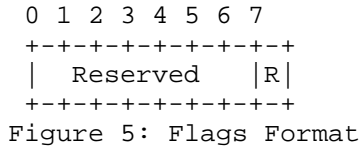
Where:

Type, 16 bits: reserving-bandwidth Capability TLV where the value is TBD;

Length, 16 bits: Gives the length of the flags field in octets, and

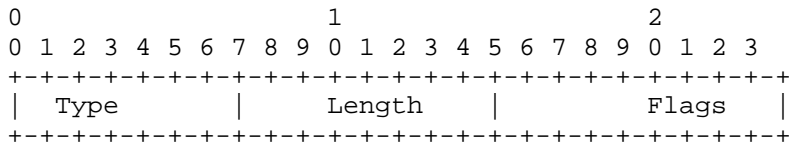
is currently set to 1;

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



Reserving bandwidth (R) field, 1 bit: Setting the R bit to 1 indicates that the node is capable of supporting reserving-bandwidth capability. When this is set to 0, it means that this node cannot reserve bandwidth for PTP LSP.

Reserved, 7 bits: Reserved for future use. The reserved bits must be ignored by the receiver. The reserving-bandwidth Capability sub-TLV is applicable to both OSPFv2 and OSPFv3. 2. reserving-bandwidth 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 reserving-bandwidth capability sub-TLV that can be carried as part of the Extended IS Reachability TLV. The reserving-bandwidth 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 reserving-bandwidth capability sub-TLV is present, the receiving system MUST only process the first instance of the sub-TLV. The format of the IS-IS reserving-bandwidth 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.



Where:

Type, 8 bits: reserving-bandwidth 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.

```

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

Figure 7: Flags Format

Reserving bandwidth (R) field, 1 bit: Setting the R bit to 1 indicates that the node is capable of supporting reserving-bandwidth capability. When this is set to 0, it means that this node cannot reserve bandwidth for PTP LSP.

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

6.2. RSVP-TE extensions

A new flag in the SESSION ATTRIBUTE object and new Error Value sub-codes in the ERROR SPEC object are proposed in this document.

1.PTP LSP Congestion Detection Request The following new flag of the SESSION_ATTRIBUTE object (C-Type 1 and 7) is defined: Path congestion detection request: 0x40 This flag indicates that a PTP LSP congestion detection (of the current PTP LSP in use) is requested.

2.New Error Value Sub-Codes As defined in [RFC3209], the Error Code 25 in the ERROR SPEC object corresponds to a Notify Error. This document adds a new Error Value sub-codes:

9,PDV degradation.

7. Other considerations

Network congestion may also led to PTP packets being dropped. So, in addition to the PDV, the statistics for packet loss rate SHOULD be collected by the 1588 Slave. When packet loss rate has going up a

certain threshold, the 1588 Slave send a notify message to the 1588 master that decides to reoptimize the PTP LSP or not.

8. Security Considerations

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.

9. Acknowledgements

The authors would like to thank Li He, Liang Xia, Lizhong Jin, Zhitao Fu, Weilian Jiang and other members for their suggestions and helpful comments during the discussions of this document.

10. IANA Considerations

10.1. 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 reserving-bandwidth capability sub-TLV carried within the Router Link TLV.

Value	Sub-TLV	References
-----	-----	-----
TBD	reserving-bandwidth node sub-TLV	(this document)

10.2. 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 reserving-bandwidth capability sub-TLV carried within the Extended IS Reacability TLV.

Value	Sub-TLV	References
-----	-----	-----
TBD	reserving-bandwidth node sub-TLV	(this document)

10.3. IANA Considerations for RSVP

IANA assigned three new error sub-code values for the RSVP PathErr Notify message (Error code=25): 9, PDV degradation.

11. References

11.1. Normative References

- [I-D.ietf-tictoc-1588overmpls]
S. Davari, A. Oren, M. Bhatia, P. Roberts, L. Montini,
"Transporting PTP messages (1588) over MPLS Networks",
draft-ietf-tictoc-1588overmpls-02 , October 2011.
- [IEEE] "IEEE 1588-2008, "IEEE Standard for a Precision Clock
Synchronization Protocol for Networked Measurement and
Control Systems", , 2008.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,
and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP
Tunnels", RFC 3209, December 2001.
- [RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-
Edge (PWE3) Architecture", RFC 3985, March 2005.
- [RFC4736] Vasseur, JP., Ikejiri, Y., and R. Zhang, "Reoptimization
of Multiprotocol Label Switching (MPLS) Traffic
Engineering (TE) Loosely Routed Label Switched Path
(LSP)", RFC 4736, November 2006.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
(BFD)", RFC 5880, June 2010.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow,
"Bidirectional Forwarding Detection (BFD) for MPLS Label
Switched Paths (LSPs)", RFC 5884, June 2010.

11.2. Informative References

- [ISO] "ISO/IEC 10589:1992, "Intermediate system to Intermediate
system routing information exchange protocol for use in
conjunction with the Protocol for providing the
Connectionless-mode Network Service (ISO 8473)", , 1992.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and
dual environments", RFC 1195, December 1990.

- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC3784] Smit, H. and T. Li, "Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)", RFC 3784, June 2004.
- [RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.
- [RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, September 2008.

Authors' Addresses

Junhui Zhang
ZTE Corporation
No.50 Ruanjian Ave,Yuhuatai District
Nanjing 210012
P.R.China

Email: zhang.junhuil@zte.com.cn
URI: <http://www.zte.com.cn/>

Liang Xia
ZTE Corporation
No.50 Ruanjian Ave, Yuhuatai District
Nanjing 210012
P.R.China

Email: xia.liang2@zte.com.cn
URI: <http://www.zte.com.cn/>

