NTP Working Group D. Sibold
Internet-Draft PTB
Intended status: Standards Track S. Roettger
Expires: August 18, 2014
K. Teichel
PTB
February 14, 2014

Network Time Security
draft-ietf-ntp-network-time-security-02.txt

Abstract

This document describes the Network Time Security (NTS) protocol that enables secure authentication of time servers using Network Time Protocol (NTP) or Precision Time Protocol (PTP). Its design considers the special requirements of precise timekeeping, which are described in Security Requirements of Time Protocols in Packet Switched Networks [I-D.ietf-tictoc-security-requirements].

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 [RFC2119].

Table of Contents

1. Introduction

Time synchronization protocols are utilized more and more to synchronize clocks in networked infrastructures. The reliable performance of such infrastructures can be degraded seriously by successful attacks against the time synchronization protocol. Therefore, time synchronization protocols applied in critical infrastructures have to provide security measures to defeat possible adversaries. Consequently, the widespread Network Time Protocol (NTP) [RFC5905] was supplemented by the autokey protocol [RFC5906] which shall ensure authenticity of the NTP server and integrity of the protocol packets. Unfortunately, the autokey protocol exhibits various severe security vulnerabilities as revealed in a thorough analysis of the protocol [Roettger]. For the Precision Time Protocol (PTP), Annex K of the standard document IEEE 1588 [IEEE1588] defines an informative security protocol that is still in experimental state.

Because of autokey’s security vulnerabilities and the absence of a standardized security protocol for PTP, these protocols cannot be applied in environments in which compliance requirements demand authenticity and integrity protection. This document specifies a security protocol which ensures authenticity of the time server via a Public Key Infrastructure (PKI) and integrity of the time synchronization protocol packets and which therefore enables the usage of NTP and PTP in such environments.

The protocol is specified with the prerequisite in mind that precise timekeeping can only be accomplished with stateless time synchronization communication, which excludes the utilization of standard security protocols like IPsec or TLS for time synchronization messages. This prerequisite corresponds with the requirement that a security mechanism for timekeeping must be designed in such a way that it does not degrade the quality of the time transfer [I-D.ietf-tictoc-security-requirements].

Note:

> The intent is to formulate the protocol to be applicable to NTP as well as PTP. In the current state the draft focuses on the application to NTP.

2. Security Threats

A profound analysis of security threats and requirements for NTP and PTP can be found in the I-D [I-D.ietf-tictoc-security-requirements].

3. Objectives

The objectives of the NTS specifications are as follows:

- Authenticity: NTS enables the client to authenticate its time server.
- Integrity: NTS protects the integrity of time synchronization protocol packets via a message authentication code (MAC).
- Confidentiality: NTS does not provide confidentiality protection of the time synchronization packets.
- Modes of operation: All operational modes of NTP are supported.
- Operational modes of PTP should be supported as far as possible.
- Hybrid mode: Both secure and insecure communication modes are possible for NTP servers and clients, respectively.
- Compatibility:
  - Unsecured NTP associations shall not be affected.
  - An NTP server that does not support NTS shall not be affected by NTS authentication requests.

4. Terms and Abbreviations

- TESLA: Timed Efficient Stream Loss-Tolerant Authentication

5. NTS Overview

5.1. Symmetric and Client/Server Mode

Authenticity of the time server is verified once by utilization of X.509 certificates. Authenticity and integrity of the NTP packets are then ensured by a Message Authentication Code (MAC), which is attached to the NTP packet. The calculation of the MAC includes the whole NTP packet and the cookie which is shared between client and server. It is calculated according to:

cookie = MSB_128 (HMAC(server seed, public key of client)),

with the server seed as key, and where the function MSB_128 cuts off the 128 most significant bits of the result of the HMAC function. The server seed is a 128 bit random value of the server, which has to be kept secret. The cookie thus never changes as long as the server seed stays the same. The server seed has to be refreshed periodically in order to provide key freshness as required in [I-D.ietf-tictoc-security-requirements]. The server does not keep a state of the client. Therefore it has to recalculate the cookie each time it receives a request from the client. To this end, the client has to attach the hash value of its public key to each request (see Section 6.4).

5.2. Broadcast Mode

Just as in the case of the client server mode and symmetric mode, authenticity and integrity of the NTP packets are ensured by a MAC, which is attached to the NTP packet by the sender. The verification of the authenticity is based on the TESLA protocol, in particular on its "Not Re-using Keys" scheme, see section 3.7.2 of [RFC4082]. TESLA is based on a one-way chain of keys, where each key is the output of a one-way function applied on the previous key in the chain. The last element of the chain is shared securely with all clients. The server splits time into intervals of uniform duration and assigns each key to an interval in reverse order, starting with the penultimate. At each time interval, the server sends an NTP broadcast packet appended by a MAC, calculated using the corresponding key, and the key of the previous disclosure interval. The client verifies the MAC by buffering the packet until the disclosure of the key in its associated disclosure interval. In order to be able to verify the validity of the key, the client has to be loosely time synchronized to the server. This has to be accomplished during the initial client server exchange between broadcast client and server. For a more detailed description of the TESLA protocol see Appendix D.

6. Protocol Messages

Note that this section currently describes realization of the message format of NTS only for its utilization for NTP, in which the NTS specific data are enclosed in extension fields on top of NTP packets. A specification of NTS messages for PTP would have to be developed accordingly.

The steps described in Section 6.1 - Section 6.4 belong to the unicast mode, while Section 6.5 and Section 6.6 explain the steps involved in the broadcast mode of NTS.

6.1. Association Messages

In this step, the hash and signature algorithms that are used for the rest of the protocol are negotiated.

6.1.1. Message type: "client_assoc"

The protocol sequence starts with the client sending an association message, called client_assoc. This message contains

- the version number of NTS that the client wants to use (this SHOULD be the highest version number that it supports),
- the hostname of the client,
- a selection of hash algorithms, and
- a selection of accepted algorithms for the signatures.

For NTP, this message is realized as a packet with an extension field of type "association", which contains all this data.

6.1.2. Message type: "server_assoc"

This message is sent by the server upon receipt of client_assoc. It contains

- the version number used for the rest of the protocol (which SHOULD be determined as the minimum over the client's suggestion in the client_assoc and the highest supported by the server),
- the hostname of the server, and
- the server's choice of algorithm for the signatures and cryptographic hash algorithm, both of which MUST be chosen from the client's proposals.

In the case of NTP, the data is enclosed in a packet's extension field, also of type "association".

6.2. Certificate Messages

In this step, the client receives the certification chain up to a trusted anchor. With the established certification chain the client is able to verify the server signatures and, hence, the authenticity of the server messages with extension fields is ensured.

Discussion:

Note that in this step the client validates the authenticity of its immediate NTP server only. It does not recursively validate the authenticity of each NTP server on the time synchronization chain. Recursive authentication (and authorization) as formulated in [I-D.ietf-tictoc-security-requirements] depends on the chosen trust anchor.

6.2.1. Message type: "client_cert"

This message is sent by the client, after it successfully verified the content of the received server_assoc message (see Section 7.1.1). It contains

- o the negotiated version number,
- o the client's hostname, and
- o the signature algorithm negotiated during the association messages.

It is realized as an NTP packet with extension field of type "certificate request" for the necessary data.

6.2.2. Message type: "server_cert"

This message is sent by the server, upon receipt of a client_cert message, if the version number and choice of methods communicated in that message are actually supported by the server. It contains

- o all the information necessary to authenticate the server to the client. This is a chain of certificates, which starts at the server and goes up to a trusted authority, where each certificate MUST be certified by the one directly following it.

This message is realized for NTP as a packet with extension field of type "certificate" which holds the certification data.

6.3. Cookie Messages

During this step, the server transmits a secret cookie to the client securely. The cookie will be used for integrity protection during unicast time synchronization.

6.3.1. Message type: "client_cook"

This message is sent by the client, upon successful authentication of the server. In this message, the client requests a cookie from the server. It contains

- o the negotiated version number,
- o the hash algorithm H negotiated between client and server during the association messages,
- o the client's public key.

For NTP, an extension field of type "cookie request" holds the listed data.

6.3.2. Message type: "server_cook"

This message is sent by the server, upon receipt of a client_cook message. The hash of the client's public key, as included in client_cook, is used by the server to calculate the cookie (see Section 5.1). This message contains

- o a concatenated pair, encrypted with the client's public key, where the pair consists of
    - * the cookie, and
    - * a signature of the cookie signed with the server's private key.

In the case of NTP, this is a packet with an extension field of type "cookie transmit".

6.4. Unicast Time Synchronisation Messages

In this step, the usual time synchronization process is executed, with the addition of integrity protection for all messages that the server sends. This step can be repeated as often as the client desires and as long as the integrity of the server's time responses is verified successfully. Secure time synchronization by repetition of this step is the goal of a unicast run.

6.4.1. Message type: "time_request"

This message is sent by the client when it requests time exchange. To send this message, the client MUST have received server_cook and successfully verified the cookie via the server's signature. It contains

- the negotiated version number,
- the time synchronization data that the client wants to transmit,
- a 128-bit nonce,
- the negotiated hash algorithm H,
- the hash of the client's public key under H.

It is realized as an NTP packet with the time synchronization data and an additional extension field of type "time request" for the rest of the information.

6.4.2. Message type: "time_response"

This message is sent by the server, after it received a time_request message. The server uses the hash of the client's public key and the transmitted hash algorithm to recalculate the cookie for the client. This message contains

- the server's time synchronization response data,
- the nonce transmitted in time_request,
- a MAC (generated with the cookie as key) for verification of the above.

It is realized as an NTP packet with the necessary time synchronization data and with a new extension field of type "time response". This packet has an appended MAC that is generated over the time synchronization data and the extension field, with the cookie as the key.

6.5. Broadcast Parameter Messages

In this step, the client receives the necessary information to execute the TESLA protocol in a secured broadcast association. The client can only initiate a secure broadcast association after a successful unicast run, see Section 7.1.2.

See Appendix D for more details on TESLA.

6.5.1. Message type: "client_bpar"

This message is sent by the client in order to establish a secured time broadcast association with the server. It contains

- the version number negotiated during association in unicast mode,
- the client's hostname, and
- the signature algorithm negotiated during unicast.

For NTP, this message is realized as a packet with an extension field of type "broadcast request".

6.5.2. Message type: "server_bpar"

This message is sent by the server upon receipt of a client_bpar message during the broadcast loop of the server. It contains

- the one-way function used for building the one-way key chain,
- the last key of the one-way key chain, and
- the disclosure schedule of the keys. This contains:
  * time interval duration,
  * the disclosure delay (number of intervals between use and disclosure of a key),
  * the time at which the next time interval will start, and
  * the next interval's associated index.
- The message also contains a signature signed by the server with its private key, verifying all the data listed above.

It is realized for NTP as a packet with an extension field of type "broadcast parameters", which contains all the given data.

6.6. Broadcast Message

In this step, the server keeps sending broadcast time synchronization messages to all participating clients.

6.6.1. Message type: "server_broad"

This message is sent by the server over the course of its broadcast schedule. It is part of any broadcast association. It contains

- time broadcast data,
- the index that belongs to the current interval (and therefore identifies the current, yet undisclosed key)
- the disclosed key of the previous disclosure interval (current time interval minus disclosure delay).
- a MAC, calculated with the key for the current time interval, verifying the time data

The message is realized as an NTP broadcast packet with the time broadcast data and with an extension field of type "broadcast message", which contains the rest of the listed data. The NTP packet is then appended by a MAC verifying the time data, but not the extension field.

7. Protocol Sequence

7.1. The client

7.1.1. The client in unicast mode

For a unicast run, the client performs the following steps:

1. It sends a client_assoc message to the server.

2. It waits for a reply in the form of a server_assoc message. After receipt of the message it performs the following checks:

   * The message MUST contain a conform version number.

   * The client has to verify that the server has chosen the signature and hash algorithms from its proposal sent in the client_assoc message.

   If one of the checks fails, the client MUST abort the run.

3. The client then sends a client_cert message to the server.

4. It awaits a reply in the form of a server_cert message and performs an authenticity check. If this check fails, the client MUST abort the run.

5. Next, it sends a client_cook message to the server.

6. It awaits a reply in the form of a server_cook message; upon receipt it executes the following actions:

   * It decrypts the message with its own private key.

   * It checks that the decrypted message has the format of a 128 bit Cookie concatenated with its own signature value, verifiable with the server’s public key.

   If the check fails, the client MAY abort the run.

7. The client sends a time_request message to the server.

8. It awaits a reply in the form of a time_response message. Upon receipt, it checks:

   * that the transmitted nonce belongs to the previous time_request message and .

   * that the appended MAC verifies the time data and the transmitted nonce.

   If the nonce is invalid, the client MUST ignore this time_response message. If the MAC is invalid, the client MUST do one of the following: abort the run or go back to step 5. If both checks are successful, the client SHOULD continue time synchronization by going back to step 7.

The client’s behaviour in unicast mode is also expressed in Figure 1.

7.1.2. The client in broadcast mode

To establish a secure broadcast association with a broadcast server, the client MUST initially authenticate the broadcast server and securely synchronize its time to it up to an upper bound for its time offset in unicast mode. After that, the client performs the following steps:

1. It sends a client_bpar message to the server.

2. It waits for a reply in the form of a server_bpar message after which it performs the following checks:

   * The message must contain all the necessary information for the TESLA protocol, as listed in Section 6.5.2.

* Verification of the message’s signature.

If any information is missing or cannot be verified as signed by the server, the client MUST abort the broadcast run.

3. The client awaits time synchronization data in the form of a server_broadcast message. Upon receipt, it performs the following checks:

   1. Proof that the MAC is based on a key that is not yet disclosed. This is achieved via a disclosure schedule, so this is where loose time synchronization is required. If verified the packet will be buffered for later authentication. Otherwise, the client MUST discard it. Note that the time information included in the packet will not be used for synchronization until its authenticity could be verified.

   2. The client checks whether it already knows the disclosed key. If so, the client SHOULD discard the packet to avoid a buffer overrun. If not, the client verifies that the disclosed key belongs to the one-way key chain by applying the one-way function until equality with a previous disclosed key is verified. If falsified, the client MUST discard the packet.

   3. If the disclosed key is legitimate the client verifies the authenticity of any packet that it received during the corresponding time interval. If authenticity of a packet is verified it is released from the buffer and the packet’s time information can be utilized. If the verification fails authenticity is no longer given. In this case the client MUST request authentic time from the server by means of a unicast time request message.

   See RFC 4082[RFC4082] for a detailed description of the packet verification process.

The client’s behaviour in broadcast mode can also be seen in Figure 2.

7.2. The server

The server’s behaviour is not as easy to express in sequential terms as the client’s, not even for a single association with one client. This is because the server does not keep state of any connection.

7.2.1. The server in unicast mode

A broadcast server MUST also support unicast mode, in order to provide the initial time synchronization is a precondition for any broadcast association. To support unicast mode, the server MUST be ready to perform the following actions:

- o Upon receipt of a client_assoc message, the server constructs and sends a reply in the form of a server_assoc message as described in Section 6.1.2.

- o Upon receipt of a client_cert message, the server checks whether it supports the given signature algorithm. If so, it constructs and sends a server_cert message as described in Section 6.2.2.

- o Upon receipt of a client_cook message, the server calculates the cookie according to the formula given in Section 5.1. With this, it constructs a server_cook message as described in Section 6.3.2.

- o Upon receipt of a time_request message, the server re-calculates the cookie, then computes the necessary time synchronization data and constructs a time_response message as given in Section 6.4.2.

Also, it must adhere to the rule of server seed refreshing, as given in [1]. More information on that can be found in Section 7.3.

7.2.2. The server in broadcast mode

To support NTS broadcast, the server MUST be ready to perform the following actions:

- o Upon receipt of a client_bpar message, the server constructs and sends a server_bpar message as described in Section 6.5.2.

- o The server follows the TESLA protocol in all other aspects, by regularly sending server_broad messages as described in Section 6.6.1, adhering to its own disclosure schedule.

It is also the server’s responsibility to watch for the expiration date of the one-way key chain and generate a new key chain accordingly.

7.3. Server Seed Refresh

According to the requirements in [I-D.ietf-tictoc-security-requirements] the server has to refresh its server seed periodically. As a consequence the cookie used in the time request messages becomes invalid. In this case the client cannot verify the attached MAC and has to respond accordingly by re-initiating the protocol with a cookie request (Section 6.3). This is true for the unicast and broadcast mode, respectively.

Additionally, in broadcast mode the client has to restart the broadcast sequence with a time request message if the one-way key chain expires.

During certificate message exchange the client reads the expiration date of the period of validity of the server certificate. The client MAY restart the protocol sequence with the association message before the server certificate expires.

8. Hash Algorithms and MAC Generation

8.1. Hash Algorithms

Hash algorithms are used at different points: calculation of the cookie and the MAC, and hashing of the public key. Client and server negotiate a hash algorithm H during the association message exchange (Section 6.1) at the beginning of a unicast run. The selected algorithm H is used for all hashing processes in that run.

In broadcast mode, hash algorithms are used as pseudo random functions to construct the one-way key chain. Here, the utilized hash algorithm is communicated by the server and non-negotiable.

The list of the hash algorithms supported by the server has to fulfill the following requirements:

- it MUST NOT include MD5 or weaker algorithms,

- it MUST include SHA-256 or stronger algorithms.

8.2. MAC Calculation

For the calculation of the MAC client and server are using a Keyed-Hash Message Authentication Code (HMAC) approach [RFC2104]. The HMAC is generated with the hash algorithm specified by the client (see Section 8.1).

9. Server Seed Considerations

The server has to calculate a random seed which has to be kept secret and which MUST be changed periodically. The server MUST generate a seed for each supported hash algorithm.

9.1. Server Seed Algorithm

9.2. Server Seed Live Time

10. IANA Considerations

This document makes no request of IANA.

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

11. Security Considerations

11.1. Initial Verification of the Server Certificates

The client has to verify the validity of the certificates during the certification message exchange (Section 6.2). Since it generally has no reliable time during this initial communication phase, it is impossible to verify the period of validity of the certificates. Therefore, the client MUST use one of the following approaches:

- The validity of the certificates is preconditioned. Usually this will be the case in corporate networks.

- The client ensures that the certificates are not revoked. To this end, the client uses the Online Certificate Status Protocol (OCSP) defined in [RFC6277].

- The client requests a different service to get an initial time stamp in order to be able to verify the certificates' periods of validity. To this end, it can, e.g., use a secure shell connection to a reliable host. Another alternative is to request a time stamp from a Time Stamping Authority (TSA) by means of the Time-Stamp Protocol (TSP) defined in [RFC3161].

11.2. Revocation of Server Certificates

According to Section 7.3, it is the client's responsibility to initiate a new association with the server after the server's certificate expires. To this end the client reads the expiration date of the certificate during the certificate message exchange (Section 6.2). Besides, certificates may also be revoked prior to the normal expiration date. To increase security the client MAY verify the state of the server's certificate via OCSP periodically.

11.3. Usage of NTP Pools

The certification based authentication scheme described in Section 6 is not applicable to the concept of NTP pools. Therefore, NTS is not able to provide secure usage of NTP pools.

11.4. Denial-of-Service in Broadcast Mode

TESLA authentication buffers packets for delayed authentication. This makes the protocol vulnerable to flooding attacks, causing the client to buffer excessive numbers of packets. To add stronger DoS protection to the protocol client and server SHALL use the "Not Re-using Keys" scheme of TESLA as pointed out in section 3.7.2 of RFC 4082 [RFC4082]. In this scheme the server never uses a key for the MAC generation more than once. Therefore the client can discard any packet that contains a disclosed key it knows already, thus preventing memory flooding attacks.

Note, an alternative approach to enhance TESLA’s resistance against DoS attacks involves the addition of a group MAC to each packet. This requires the exchange of an additional shared key common to the whole group. This adds additional complexity to the protocol and hence is currently not considered in this document.

12. Acknowledgements

The authors would like to thank David Mills and Kurt Roeckx for discussions and comments on the design of NTS. Also, thanks to Harlan Stenn for his technical review and specific text contributions to this document.

13. References

13.1. Normative References

[IEEE1588]
IEEE Instrumentation and Measurement Society. TC-9 Sensor Technology, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems", 2008.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

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

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC5906] Haberman, B. and D. Mills, "Network Time Protocol Version 4: Autokey Specification", RFC 5906, June 2010.

[RFC6277] Santesson, S. and P. Hallam-Baker, "Online Certificate Status Protocol Algorithm Agility", RFC 6277, June 2011.

13.2. Informative References

[I-D.ietf-tictoc-security-requirements]
Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", draft-ietf-tictoc-security-requirements-05 (work in progress), April 2013.

[Roettger]
Roettger, S., "Analysis of the NTP Autokey Procedures", February 2012.

Appendix A. Flow Diagrams of Client Behaviour

```
                        +---------------------+
                        |Association Messages |
                        +----------+----------+
                                   |
                                   v
                        +---------------------+
                        |Certificate Messages |
                        +----------+----------+
                                   |
   +------------------------------>o
   |                               |
   |                               v
   |                       +---------------+
   |                       |Cookie Messages|
   |                       +-------+-------+
   |                               |
```

```
   |                               o<------------------------------+
   |                               |                               |
   |                               v                               |
   |                     +-------------------+                     |
   |                     |Time Sync. Messages|                     |
   |                     +---------+---------+                     |
   |                               |                               |
   |                               v                               |
   |                            +-----+                            |
   |                            |Check|                            |
   |                            +--+--+                            |
   |                               |                               |
   |            /------------------+------------------\            |
   |           v                   v                   v           |
   |     .-----------.      .-------------.        .-------.       |
   |    ( MAC Failure )    ( Nonce Failure )      ( Success )      |
   |     '-----+-----'      '------+------'        '---+---'       |
   |           |                   |                   |           |
   |           v                   v                   v           |
   |    +-------------+     +-------------+     +--------------+   |
   |    |Discard Data |     |Discard Data |     |Sync. Process |   |
   |    +-------------+     +------+------+     +------+-------+   |
   |           |                   |                   |           |
   |           |                   |                   v           |
   +-----------+                   +------------------>o-----------+
```

Figure 1: The client's behaviour in NTS unicast mode.

```
                            +-----------------------------+
                            |Broadcast Parameter Messages |
                            +--------------+--------------+
                                           |
                                           o<--------------------------+
                                           |                           |
                                           v                           |
                            +-----------------------------+            |
                            |Broadcast Time Sync. Message |            |
                            +--------------+--------------+            |
                                           |                           |
   +-------------------------------------->o                           |
   |                                       |                           |
   |                                       v                           |
   |                             +-------------------+                 |
   |                             |Key and Auth. Check|                 |
   |                             +---------+---------+                 |
   |                                       |                           |
   |                      /----------------*----------------\          |
   |                     v                                   v         |
```

```
   |                .---------.                        .---------.    |
   |               ( Verified  )                      ( Falsified )   |
   |                '----+----'                        '----+----'    |
   |                     |                                  |         |
   |                     v                                  v         |
   |              +-------------+                        +-------+     |
   |              |Store Message|                        |Discard|     |
   |              +------+------+                        +---+---+     |
   |                     |                                  |         |
   |                     v                                  +---------o
   |             +---------------+                                     |
   |             |Check Previous |                                     |
   |             +-------+-------+                                     |
   |                     |                                             |
   |            /--------*--------\                                    |
   |           v                   v                                   |
   |      .---------.         .---------.                              |
   |     ( Verified  )       ( Falsified )                             |
   |      '----+----'         '----+----'                              |
   |           |                   |                                   |
   |           v                   v                                   |
   |    +-------------+   +------------------+                         |
   |    |Sync. Process|   |Discard Previous  |                         |
   |    +------+------+   +--------+---------+                         |
   |           |                   |                                   |
   +-----------+                   +-----------------------------------+
```

Figure 2: The client's behaviour in NTS broadcast mode.

Appendix B. Extension fields

In Section 6, some new extension fields for NTP packets are introduced. They are listed here again, for reference.

name	used in
"association"	client_assoc server_assoc
"certificate request"	client_cert
"certificate"	server_cert
"cookie request"	client_cook
"cookie transmit"	server_cook

"time request"	time_request
"time response"	time_response
"broadcast request"	client_bpar
"broadcast parameters"	server_bpar
"broadcast message"	server_broad

Appendix C. TICTOC Security Requirements

The following table compares the NTS specifications against the TICTOC security requirements [I-D.ietf-tictoc-security-requirements].

Section	Requirement from I-D tictoc security-requirements-05	Requirement level	NTS
5.1.1	Authentication of Servers	MUST	OK
5.1.1	Authorization of Servers	MUST	OK
5.1.2	Recursive Authentication of Servers (Stratum 1)	MUST	OK
5.1.2	Recursive Authorization of Servers (Stratum 1)	MUST	OK
5.1.3	Authentication and Authorization of Slaves	MAY	-
5.2	Integrity protection.	MUST	OK
5.3	Protection against DoS attacks	SHOULD	OK
5.4	Replay protection	MUST	OK
5.5.1	Key freshness.	MUST	OK
5.5.2	Security association.	SHOULD	OK
5.5.3	Unicast and multicast associations.	SHOULD	OK
5.6	Performance: no degradation in quality of time transfer.	MUST	OK

	Performance: lightweight computation	SHOULD	OK
	Performance: storage, bandwidth	SHOULD	OK
5.7	Confidentiality protection	MAY	NO
5.8	Protection against Packet Delay and Interception Attacks	SHOULD	NA*)
5.9.1	Secure mode	MUST	-
5.9.2	Hybrid mode	MAY	-

*) Ensured by NTP via multi-source configuration.

Comparsion of NTS sepecification against TICTOC security requirements.

Appendix D. Broadcast Mode

Authors' Addresses

Dieter Sibold
Physikalisch-Technische Bundesanstalt
Bundesallee 100
Braunschweig D-38116
Germany

Phone: +49-(0)531-592-8420
Fax: +49-531-592-698420
Email: dieter.sibold@ptb.de

Stephen Roettger

Email: stephen.roettger@googlemail.com

Kristof Teichel
Physikalisch-Technische Bundesanstalt
Bundesallee 100
Braunschweig D-38116
Germany

Email: kristof.teichel@ptb.de

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

October 16, 2015

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

Abstract

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

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

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

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

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

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

PTP defines several clock types, including ordinary clocks, boundary clocks, end-to-end transparent clocks, and peer-to-peer transparent clocks. Transparent clocks are situated at intermediate nodes and update the Correction Field inside PTP messages in order to reflect the time required to transit the node.

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

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

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

2. Terminology

1588: The timing distribution protocol defined in IEEE 1588.

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

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

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

NTP: The timing distribution protocol defined in RFC 5905.

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

PTP: Precision Time Protocol. See 1588.

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

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

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

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

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

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

3. Problem Statement

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

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

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

4. Timing over MPLS Architecture

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

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

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

```
 +--------+     +-------+     +-------+     +-------+     +--------+
 |Switch, |     |       |     |       |     |       |     |Switch, |
 | Router |-----|  LER  |-----|  LSR  |-----|  LER  |-----| Router |
 |        |     |  OC   |     |  TC   |     |  OC   |     |        |
 +--------+     +-------+     +-------+     +-------+     +--------+
                /                                  \
 +-------+     /                                    \     +-------+
 |  LER  |    /                                      \    |  LER  |
 | Master|---/                                        \---| Slave |
 | Clock |                                                | Clock |
 +-------+                                                +-------+
```

Figure (1) - Deployment example 1 of timing over MPLS network

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

```
 +--------+     +-------+     +-------+     +-------+     +--------+
 |Switch, |     |       |     |       |     |       |     |Switch, |
 | Router |-----|  LER  |-----|  LSR  |-----|  LER  |-----| Router |
 | OC/BC  |     |  BC   |     |  TC   |     |  BC   |     | OC/BC  |
 +--------+     +-------+     +-------+     +-------+     +--------+
```

Figure (2) - Deployment example 2 of timing over MPLS network

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

```
 +--------+     +-------+     +-------+     +-------+     +--------+
 |Switch, |     |       |     |       |     |       |     |Switch, |
 | Router |-----|  LER  |-----|  LSR  |-----|  LER  |-----| Router |
 |OC/TC/BC|     |  TC   |     |  TC   |     |  TC   |     |OC/TC/BC|
 +--------+     +-------+     +-------+     +-------+     +--------+
```

Figure (3) - Deployment example 3 of timing over MPLS network

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

```
 +--------+     +-------+     +-------+     +-------+     +--------+
 |Switch, |     |       |     |       |     |       |     |Switch, |
 | Router |-----|  LER  |-----|  LSR  |-----|  LER  |-----| Router |
 | OC/BC  |     |  BC   |     |  BC   |     |  BC   |     | OC/BC  |
 +--------+     +-------+     +-------+     +-------+     +--------+
```

Figure (4) - Deployment example 3 of timing over MPLS network

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

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

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

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

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

5. Dedicated LSPs for Timing messages

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

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

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

6. Timing over LSP Encapsulation

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

6.1. Timing over UDP/IP over MPLS Encapsulation

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

```
+----------------------+
|   Timing LSP Label   |
+----------------------+
|        IPv4/6        |
+----------------------+
|         UDP          |
+----------------------+
|   timing message     |
+----------------------+
```

Figure (4) - Timing over UDP/IP over MPLS Encapsulation

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

6.2. Timing over PW Encapsulation

Another method of transporting timing over MPLS networks is to use Ethernet encapsulated timing messages, and to transport these in an Ethernet PW which in turn is transported over a Timing LSP. In the case of PTP, the Ethernet encapsulation MUST comply to Annex F of [IEEE-1588] and the Ethernet PW encapsulation to [RFC4448], resulting in the format shown in Figure 5(A).

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

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

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

Figure (5) - Timing over PW Encapsulations

7. Timing message Processing

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

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

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

o SYNC

- o DELAY_REQ (Delay Request)

- o PDELAY_REQ (Peer Delay Request)

- o PDELAY_RESP (Peer Delay Response)

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

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

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

8. Protection and Redundancy

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

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

9. ECMP and Entropy

To ensure the correct operation of slave clocks and avoid error introduced by forward and reverse path delay asymmetry, the physical path taken by timing messages MUST be the same for all timing messages. In particular, the PTP event messages listed in section 7 MUST be routed in the same way.

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

10. PHP

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

11. OAM, Control and Management

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

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

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

12. QoS Considerations

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

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

13. FCS and Checksum Recalculation

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

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

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

14. Behavior of LER/LSRs

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

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

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

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

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

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

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

Non-timing-capable/aware LSRs just perform label switching on the packets encapsulated in Timing LSPs and don’t perform any timing related processing. However, as explained in QoS section, timing packets MUST be still be treated with the highest priority based on their Traffic Class marking.

15. Other considerations

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

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

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

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

16. Security Considerations

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

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

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

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

17. Applicability Statement

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

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

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

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

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

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

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

18. Acknowledgements

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

19. IANA Considerations

There are no IANA requirements in this specification.

20. References

20.1. Normative References

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

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

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

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

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

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

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

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

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

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

20.2. Informative References

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

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

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

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

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

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

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

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

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

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

Appendix A. Appendix

A.1. Routing extensions for Timing-aware Routers

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

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

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

o Capable of processing PTP, NTP or other timing flows

o Capable of performing TC operation

o Capable of performing BC operation

A.2. Signaling Extensions for Creating Timing LSPs

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

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

o Number of VLANs in case of PW encapsulation

- o Time-stamp field Type

  * Correction Field, time-stamp

- o Time-stamp Field format

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

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

Authors' Addresses

Shahram Davari
Broadcom Corp.
San Jose, CA 95134
USA

Email: davari@broadcom.com

Amit Oren
Broadcom Corp.
San Jose, CA 95134
USA

Email: amito@broadcom.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

Email: manav.bhatia@alcatel-lucent.com

Peter Roberts
Alcatel-Lucent
Kanata
Canada

Email: peter.roberts@alcatel-lucent.com

Laurent Montini
Cisco Systems
San Jose CA
USA

Email: lmontini@cisco.com

TICTOC Working Group D.A. Arnold
Internet-Draft Meinberg-USA
Intended status: Standards Track H.G. Gerstung
Expires: 24 January 2025 Meinberg
23 July 2024

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

draft-ietf-tictoc-ptp-enterprise-profile-28

Abstract

This document describes a Precision Time Protocol (PTP) Profile IEEE 1588-2019 [IEEE1588] for use in an IPv4 or IPv6 Enterprise information system environment. The PTP Profile uses the End-to-End delay measurement mechanism, allows both multicast and unicast Delay Request and Delay Response messages.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Precision Time Protocol ("PTP"), standardized in IEEE 1588, has been designed in its first version (IEEE 1588-2002) with the goal to minimize configuration on the participating nodes. Network communication was based solely on multicast messages, which unlike NTP did not require that a receiving node in IEEE 1588-2019 [IEEE1588] need to know the identity of the time sources in the network. This document describes clock roles and PTP Port states using the optional alternative terms timeTransmitter, instead of master, and timeReceiver, instead of slave, as defined in the IEEE 1588g [IEEE1588g] amendment to IEEE 1588-2019 [IEEE1588] .

The "Best TimeTransmitter Clock Algorithm" (IEEE 1588-2019 [IEEE1588] Subclause 9.3), a mechanism that all participating PTP nodes MUST follow, set up strict rules for all members of a PTP domain to determine which node MUST be the active reference time source (Grandmaster). Although the multicast communication model has advantages in smaller networks, it complicated the application of PTP in larger networks, for example in environments like IP based telecommunication networks or financial data centers. It is considered inefficient that, even if the content of a message applies only to one receiver, it is forwarded by the underlying network (IP) to all nodes, requiring them to spend network bandwidth and other resources, such as CPU cycles, to drop the message.

The third edition of the standard (IEEE 1588-2019) defines PTPv2.1 and includes the possibility to use unicast communication between the PTP nodes in order to overcome the limitation of using multicast messages for the bi-directional information exchange between PTP nodes. The unicast approach avoided that. In PTP domains with a lot of nodes, devices had to throw away most of the received multicast messages because they carried information for some other node. The percent of PTP message that are discarded as irrelevant to the receving node can exceded 99% (Estrela and Bonebakker [Estrela_and_Bonebakker]).

PTPv2.1 also includes PTP Profiles (IEEE 1588-2019 [IEEE1588] subclause 20.3). This construct allows organizations to specify selections of attribute values and optional features, simplifying the configuration of PTP nodes for a specific application. Instead of having to go through all possible parameters and configuration options and individually set them up, selecting a PTP Profile on a PTP node will set all the parameters that are specified in the PTP Profile to a defined value. If a PTP Profile definition allows multiple values for a parameter, selection of the PTP Profile will set the profile-specific default value for this parameter. Parameters not allowing multiple values are set to the value defined in the PTP Profile. Many PTP features and functions are optional, and a PTP Profile should also define which optional features of PTP are required, permitted, and prohibited. It is possible to extend the PTP standard with a PTP Profile by using the TLV mechanism of PTP (see IEEE 1588-2019 [IEEE1588] subclause 13.4), defining an optional Best TimeTransmitter Clock Algorithm and a few other ways. PTP has its own management protocol (defined in IEEE 1588-2019 [IEEE1588] subclause 15.2) but allows a PTP Profile to specify an alternative management mechanism, for example NETCONF.

In this document the term PTP Port refers to a logical access point of a PTP instantiation for PTP communincation in a network.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Technical Terms

* Acceptable TimeTransmitter Table: A PTP timeReceiver Clock may maintain a list of timeTransmitters which it is willing to synchronize to.

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

* Announce message: Contains the timeTransmitter Clock properties of a timeTransmitter Clock. Used to determine the Best TimeTransmitter.

* Best timeTransmitter: A clock with a PTP Port in the timeTransmitter state, operating as the Grandmaster of a PTP domain.

* Best TimeTransmitter Clock Algorithm: A method for determining which state a PTP Port of a PTP clock should be in. The state decisions lead to the formation of a clock spanning tree for a PTP domain.

* Boundary Clock: A device with more than one PTP Port. Generally Boundary Clocks will have one PTP Port in timeReceiver state to receive timing and other PTP Ports in timeTransmitter state to re-distribute the timing.

* Clock Identity: In IEEE 1588-2019 this is a 64-bit number assigned to each PTP clock which MUST be globally unique. Often it is derived from the Ethernet MAC address.

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

* End-to-End delay measurement mechanism: A network delay measurement mechanism in PTP facilitated by an exchange of messages between a timeTransmitter Clock and a timeReceiver Clock. These messages might traverse Transparent Clocks and PTP unaware switches. This mechanism might not work properly if the Sync and Delay Request messages traverse different network paths.

* Grandmaster: the timeTransmitter Clock that is currently acting as the reference time source of the PTP domain

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

* TimeTransmitter Clock: a clock with at least one PTP Port in the timeTransmitter state.

* NTP: Network Time Protocol, defined by RFC 5905, see RFC 5905 [RFC5905]

* Ordinary Clock: A clock that has a single Precision Time Protocol PTP Port in a domain and maintains the timescale used in the domain. It may serve as a timeTransmitter Clock, or be a timeReceiver Clock.

* Peer-to-Peer delay measurement mechanism: A network delay measurement mechanism in PTP facilitated by an exchange of messages over the link between adjacent devices in a network. This mechanism might not work properly unless all devices in the network support PTP and the Peer-to-peer measurement mechanism.

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

* PTP: The Precision Time Protocol: The timing and synchronization protocol defined by IEEE 1588.

* PTP Port: An interface of a PTP clock with the network. Note that there may be multiple PTP Ports running on one physical interface, for example, mulitple unicast timeReceivers which talk to several Grandmaster Clocks in different PTP Domains.

* PTP Profile: A set of constraints on the options and features of PTP, designed to optimize PTP for a specific use case or industry. The profile specifies what is required, allowed and forbidden among options and attribute values of PTP.

* PTPv2.1: Refers specifically to the version of PTP defined by IEEE 1588-2019.

* Rogue timeTransmitter: A clock with a PTP Port in the timeTransmitter state, even though it should not be in the timeTransmitter state according to the Best TimeTransmitter Clock Algorithm, and does not set the Alternate timeTransmitter flag.

* TimeReceiver Clock: a clock with at least one PTP Port in the timeReceiver state, and no PTP Ports in the timeTransmitter state.

* TimeReceiver Only clock: An Ordinary Clock which cannot become a timeTransmitter Clock.

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

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

* Unicast Discovery: A mechanism for PTP timeReceivers to establish a unicast communication with PTP timeTransmitters using a configured table of timeTransmitter IP addresses and Unicast Message Negotiation.

* Unicast Negotiation: A mechanism in PTP for timeReceiver Clocks to negotiate unicast Sync, Announce and Delay Request message transmission rates from timeTransmitters.

4. Problem Statement

This document describes how PTP can be applied to work in large enterprise networks. See ISPCS [RFC2026] for information on IETF applicability statements. Such large networks are deployed, for example, in financial corporations. It is becoming increasingly common in such networks to perform distributed time tagged measurements, such as one-way packet latencies and cumulative delays on software systems spread across multiple computers. Furthermore, there is often a desire to check the age of information time tagged by a different machine. To perform these measurements, it is necessary to deliver a common precise time to multiple devices on a network. Accuracy currently required in the Financial Industry range from 100 microseconds to 1 nanoseconds to the Grandmaster. This PTP Profile does not specify timing performance requirements, but such requirements explain why the needs cannot always be met by NTP, as commonly implemented. Such accuracy cannot usually be achieved with a traditional time transfer such as NTP, without adding non-standard customizations such as on-path support, similar to what is done in PTP with Transparent Clocks and Boundary Clocks. Such PTP support is commonly available in switches and routers, and many such devices have already been deployed in networks. Because PTP has a complex range of features and options it is necessary to create a PTP Profile for enterprise networks to achieve interoperability between equipment manufactured by different vendors.

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

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

Interoperability among independent implementations of this PTP Profile has been demonstrated at the ISPCS Plugfest ISPCS [ISPCS].

5. Network Technology

This PTP Profile MUST operate only in networks characterized by UDP RFC 768 [RFC0768] over either IPv4 RFC 791 [RFC0791] or IPv6 RFC 8200 [RFC8200], as described by Annexes C and D in IEEE 1588 [IEEE1588] respectively. A network node MAY include multiple PTP instances running simultaneously. IPv4 and IPv6 instances in the same network node MUST operate in different PTP Domains. PTP Clocks which communicate using IPv4 can transfer time to PTP Clocks using IPv6, or the reverse, if and only if, there is a network node which simultaneously communicates with both PTP domains in the different IP versions.

The PTP system MAY include switches and routers. These devices MAY be Transparent Clocks, Boundary Clocks, or neither, in any combination. PTP Clocks MAY be Preferred timeTransmitters, Ordinary Clocks, or Boundary Clocks. The Ordinary Clocks may be TimeReceiver Only Clocks, or be timeTransmitter capable.

Note that PTP Ports will need to keep tack of the Clock ID of received messages and not just the IP or Layer 2 addresses in any network that includes Transparent Clocks, or might include them in the future. This is important since Transparent Clocks might treat PTP messages that are altered at the PTP application layer as new IP packets and new Layer 2 frames when the PTP messages are retranmitted. In IPv4 networks some clocks might be hidden behind a NAT, which hides their IP addresses from the rest of the network. Note also that the use of NATs may place limitations on the topology of PTP networks, depending on the port forwarding scheme employed. Details of implementing PTP with NATs are out of scope of this document.

PTP, similar to NTP, assumes that the one-way network delay for Sync messages and Delay Response messages are the same. When this is not true it can cause errors in the transfer of time from the timeTransmitter to the timeReceiver. It is up to the system integrator to design the network so that such effects do not prevent the PTP system from meeting the timing requirements. The details of network asymmetry are outside the scope of this document. See for example, ITU-T G.8271 [G8271].

6. Time Transfer and Delay Measurement

TimeTransmitter Clocks, Transparent Clocks and Boundary Clocks MAY be either one-step clocks or two-step clocks. TimeReceiver Clocks MUST support both behaviors. The End-to-End Delay measurement method MUST be used.

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

Sync messages MUST be sent as PTP event multicast messages (UDP port 319) to the PTP primary IP address. Two step clocks MUST send Follow-up messages as PTP general multicast messages (UDP port 320). Announce messages MUST be sent as multicast messages (UDP port 320) to the PTP primary address. The PTP primary IP address is 224.0.1.129 for IPv4 and FF0X:0:0:0:0:0:0:181 for IPv6, where X can be a value between 0x0 and 0xF. The different IPv6 address options are explained in IEEE 1588 IEEE 1588 [IEEE1588] Annex D, Section D.3. These addresses are aloted by IANA, see the Ipv6 Multicast Address Space Registry [IPv6Registry]

Delay Request messages MAY be sent as either multicast or unicast PTP event messages. TimeTransmitter Clocks MUST respond to multicast Delay Request messages with multicast Delay Response PTP general messages. TimeTransmitter Clocks MUST respond to unicast Delay Request PTP event messages with unicast Delay Response PTP general messages. This allows for the use of Ordinary Clocks which do not support the Enterprise Profile, if they are timeReceiver Only Clocks.

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

7. Default Message Rates

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

The logMessageInterval carried in the unicast Delay Response message MAY be set to correspond to the timeTransmitter ports preferred message period, rather than 7F, which indicates message periods are to be negotiated. Note that negotiated message periods are not allowed, see forbidden PTP options (Section 13).

8. Requirements for TimeTransmitter Clocks

TimeTransmitter Clocks MUST obey the standard Best TimeTransmitter Clock Algorithm from IEEE 1588 [IEEE1588]. PTP systems using this PTP Profile MAY support multiple simultaneous Grandmasters if each active Grandmaster is operating in a different PTP domain.

A PTP Port of a clock MUST NOT be in the timeTransmitter state unless the clock has a current value for the number of UTC leap seconds.

If a unicast negotiation signaling message is received it MUST be ignored.

In PTP Networks that contain Transparent Clocks, timeTransmitters might receive Delay Request messages that no longer contains the IP Addresses of the timeReceivers. This is because Transparent Clocks might replace the IP address of Delay Requests with their own IP address after updating the Correction Fields. For this deployment scenario timeTransmitters will need to have configured tables of timeReceivers' IP addresses and associated Clock Identities in order to send Delay Responses to the correct PTP Nodes.

9. Requirements for TimeReceiver Clocks

In a network which contains multiple timeTransmitters in multiple domains, TimeReceivers SHOULD make use of information from all the timeTransmitters in their clock control subsystems. TimeReceiver Clocks MUST be able to function in such networks even if they use time from only one of the domains. TimeReceiver Clocks MUST be able to operate properly in the presence of a rogue timeTransmitter. TimeReceivers SHOULD NOT Synchronize to a timeTransmitter which is not the Best TimeTransmitter in its domain. TimeReceivers will continue to recognize a Best TimeTransmitter for the duration of the Announce Time Out Interval. TimeReceivers MAY use an Acceptable TimeTransmitter Table. If a timeTransmitter is not an Acceptable timeTransmitter, then the timeReceiver MUST NOT synchronize to it. Note that IEEE 1588-2019 requires timeReceiver Clocks to support both two-step or one-step timeTransmitter Clocks. See IEEE 1588 [IEEE1588], subClause 11.2.

Since Announce messages are sent as multicast messages timeReceivers can obtain the IP addresses of a timeTransmitter from the Announce messages. Note that the IP source addresses of Sync and Follow-up messages might have been replaced by the source addresses of a Transparent Clock, so, timeReceivers MUST send Delay Request messages to the IP address in the Announce message. Sync and Follow-up messages can be correlated with the Announce message using the Clock ID, which is never altered by Transparent Clocks in this PTP Profile.

10. Requirements for Transparent Clocks

Transparent Clocks MUST NOT change the transmission mode of an Enterprise Profile PTP message. For example, a Transparent Clock MUST NOT change a unicast message to a multicast message. Transparent Clocks which syntonize to the timeTransmitter Clock might need to maintain separate clock rate offsets for each of the supported domains.

11. Requirements for Boundary Clocks

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

12. Management and Signaling Messages

PTP Management messages MAY be used. Management messages intended for a specific clock, i.e. the IEEE 1588 [IEEE1588] defined attribute targetPortIdentity.clockIdentity is not set to All 1s, MUST be sent as a unicast message. Similarly, if any signaling messages are used they MUST also be sent as unicast messages whenever the message is intended soley for a specific PTP Node.

13. Forbidden PTP Options

Clocks operating in the Enterprise Profile MUST NOT use: Peer-to-Peer timing for delay measurement, Grandmaster Clusters, The Alternate TimeTransmitter option, Alternate Timescales. Unicast discovery, or unicast negotiation. Clocks operating in the Enterprise Profile MUST avoid any optional feature that requires Announce messages to be altered by Transparent Clocks, as this would require the Transparent Clock to change the source address and prevent the timeReceiver nodes from discovering the protocol address of the timeTransmitter.

14. Interoperation with IEEE 1588 Default Profile

Clocks operating in the Enterprise Profile will interoperate with clocks operating in the Default Profile described in IEEE 1588 [IEEE1588] Annex I.3. This variant of the Default Profile uses the End-to-End delay measurement mechanism. In addition, the Default Profile would have to operate over IPv4 or IPv6 networks, and use management messages in unicast when those messages are directed at a specific clock. If either of these requirements are not met than Enterprise Profile clocks will not interoperate with Annex I.3 Default Profile Clocks. The Enterprise Profile will not interoperate with the Annex I.4 variant of the Default Profile which requires use of the Peer-to-Peer delay measurement mechanism.

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

15. Profile Identification

The IEEE 1588 standard requires that all PTP Profiles provide the following identifying information.

PTP Profile:
Enterprise Profile
Profile number: 1
Version: 1.0
Profile identifier: 00-00-5E-01-01-00

This PTP Profile was specified by the IETF

A copy may be obtained at
https://datatracker.ietf.org/wg/tictoc/documents

16. Acknowledgements

The authors would like to thank Richard Cochran, Kevin Gross, John Fletcher, Laurent Montini and many other members of IETF for reviewing and providing feedback on this draft.

This document was initially prepared using 2-Word-v2.0.template.dot and has later been converted manually into xml format using an xml2rfc template.

17. IANA Considerations

There are no IANA requirements in this specification.

18. Security Considerations

Protocols used to transfer time, such as PTP and NTP can be important to security mechanisms which use time windows for keys and authorization. Passing time through the networks poses a security risk since time can potentially be manipulated. The use of multiple simultaneous timeTransmitters, using multiple PTP domains can mitigate problems from rogue timeTransmitters and on-path attacks. Note that Transparent Clocks alter PTP content on-path, but in a manner specified in IEEE 1588-2019 [IEEE1588] that helps with time transfer accuracy. See sections 9 and 10. Additional security mechanisms are outside the scope of this document.

PTP native management messages SHOULD NOT be used, due to the lack of a security mechanism for this option. Secure management can be obtained using standard management mechanisms which include security, for example NETCONF NETCONF [RFC6241].

General security considerations of time protocols are discussed in RFC 7384 [RFC7384].

19. References

19.1. Normative References

[IEEE1588] Institute of Electrical and Electronics Engineers, "IEEE std. 1588-2019, "IEEE Standard for a Precision Clock Synchronization for Networked Measurement and Control Systems."", November 2019, <https://www.ieee.org>.

[IEEE1588g] Institute of Electrical and Electronics Engineers, "IEEE std. 1588g-2022, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment 2: Master-Slave Optional Alternative Terminology"", December 2022, <https://www.ieee.org>.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 2119, DOI 10.17487/RFC2119, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

19.2. Informative References

[Estrela_and_Bonebakker]
Estrela, P. and L. Bonebakker, "Estrela and Bonebakker, "Challenges deploying PTPv2 in a global financial company"", DOI 10.1109/ISPCS.2012.6336634, 2012, <https://www.researchgate.net/publication/260742322_Challenges_deploying_PTPv2_in_a_global_financial_company>.

[G8271] International Telecommunication Union, "ITU-T G.8271/Y.1366, "Time and Phase Synchronization Aspects of Packet Networks"", March 2020, <https://www.itu.int>.

[IPv6Registry]
Venaas, S., "IPv6 Multicast Address Space Registry", February 2024, <https://iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml>.

[ISPCS] Arnold, D., "Plugfest Report", October 2017, <https://www.ispcs.org>.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, DOI 10.17487/RFC2026, October 1996, <https://www.rfc-editor.org/info/rfc2026>.

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

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

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

Authors' Addresses

Doug Arnold
Meinberg-USA
3 Concord Rd
Shrewsbury, Massachusetts 01545
United States of America
Email: doug.arnold@meinberg-usa.com

Heiko Gerstung
Meinberg
Lange Wand 9
31812 Bad Pyrmont
Germany
Email: heiko.gerstung@meinberg.de