NTP Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

D. Sibold
PTB
S. Roettger

K. Teichel
PTB

July 04, 2014

Network Time Security
draft-ietf-ntp-network-time-security-04.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].

Status of This Memo

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

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

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

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Security Threats 4
3. Objectives 4
4. Terms and Abbreviations 5
5. NTS Overview 5
5.1. Symmetric and Client/Server Mode 5
5.2. Broadcast Mode 6
6. Protocol Messages 6
6.1. Association Messages 6
6.1.1. Message Type: "client_assoc" 7
6.1.2. Message Type: "server_assoc" 7
6.2. Certificate Messages 7
6.2.1. Message Type: "client_cert" 8
6.2.2. Message Type: "server_cert" 8
6.3. Cookie Messages 9
6.3.1. Message Type: "client_cook" 9
6.3.2. Message Type: "server_cook" 9
6.4. Unicast Time Synchronisation Messages 10
6.4.1. Message Type: "time_request" 10
6.4.2. Message Type: "time_response" 10
6.5. Broadcast Parameter Messages 11
6.5.1. Message Type: "client_bpar" 11
6.5.2. Message Type: "server_bpar" 11
6.6. Broadcast Message 12
6.6.1. Message Type: "server_broad" 12
7. Protocol Sequence 13
7.1. The Client 13
7.1.1. The Client in Unicast Mode 13
7.1.2. The Client in Broadcast Mode 15
7.2. The Server 16
7.2.1. The Server in Unicast Mode 16

7.2.2. The Server in Broadcast Mode 17
8. Server Seed Considerations 17
8.1. Server Seed Refresh 17
8.2. Server Seed Algorithm 17
8.3. Server Seed Lifetime 17
9. Hash Algorithms and MAC Generation 17
9.1. Hash Algorithms 17
9.2. MAC Calculation 18
10. IANA Considerations 18
11. Security Considerations 18
11.1. Initial Verification of the Server Certificates 18
11.2. Revocation of Server Certificates 19
11.3. Usage of NTP Pools 19
11.4. Denial-of-Service in Broadcast Mode 19
11.5. Delay Attack 19
12. Acknowledgements 20
13. References 20
13.1. Normative References 20
13.2. Informative References 21
Appendix A. Flow Diagrams of Client Behaviour 22
Appendix B. Extension Fields 25
Appendix C. TICTOC Security Requirements 25
Appendix D. Broadcast Mode 26
D.1. Server Preparations 27
D.2. Client Preparation 28
D.3. Sending Authenticated Broadcast Packets 29
D.4. Authentication of Received Packets 29
Appendix E. Random Number Generation 30
Authors' Addresses 30

1. Introduction

Time synchronization protocols are increasingly utilized 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 and also 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 "Security Requirements of Time Protocols in Packet Switched Networks" [I-D.ietf-tictoc-security-requirements].

3. Objectives

The objectives of the NTS specification 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.

- o Hybrid mode: Both secure and insecure communication modes are possible for NTP servers and clients, respectively.

- o 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

MITM Man In The Middle

NTP Network Time Protocol

NTS Network Time Security

PTP Precision Time Protocol

TESLA Timed Efficient Stream Loss-Tolerant Authentication

5. NTS Overview

5.1. Symmetric and Client/Server Mode

NTS applies X.509 certificates to verify the authenticity of the time server and to exchange a symmetric key, the so-called cookie. This cookie is then used to protect authenticity and integrity of the subsequent time synchronization packets by means of a Message Authentication Code (MAC), which is attached to each time synchronization packet. The calculation of the MAC includes the whole time synchronization packet and the cookie which is shared between client and server. It is calculated according to:

```
cookie = MSB_128 (HMAC(server seed, H(certificate of client))),
```

with the server seed as key, where H is a hash function, 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 never changes as long as the server seed stays the same, but the server seed has to be refreshed periodically in order to provide key freshness as required in [I-D.ietf-tictoc-security-requirements]. See Section 8 for details on the seed refresh and Section 7.1.1 for the client’s reaction to it.

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 certificate 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 the 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 message exchange, the hash and signature algorithms that are used throughout 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

- o the NTS message ID "client_assoc",
- o the version number of NTS that the client wants to use (this SHOULD be the highest version number that it supports),
- o the hostname of the client,
- o a selection of accepted hash algorithms,
- o a selection of accepted encryption algorithms, and
- o 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

- o the NTS message ID "server_assoc",
- o 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 message and the highest supported by the server),
- o the hostname of the server, and
- o the server’s choice of algorithm for encryption, for the signatures and for cryptographic hashing , all 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 message exchange, the client receives the certification chain up to a trusted anchor. With the established certification chain the client is able to verify the server’s signatures and, hence, authenticity of future NTS messages from the server 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 NTS message ID "client_cert",
- o the negotiated version number,
- o the client’s hostname, and
- o the signature algorithm negotiated during the association messages.

In the case of NTP, the data is enclosed in a packet’s extension field of type "certificate request".

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 the NTS message ID "server_cert",
- o the version number as transmitted in client_cert,
- o a signature, calculated over the data listed above, with the server’s private key and according to the signature algorithm transmitted in server_cert,
- 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 all of the data listed above.

6.3. Cookie Messages

During this message exchange, 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. The message contains

- o the NTS message ID "client_cook",
- o the negotiated version number,
- o the negotiated signature algorithm,
- o the negotiated encryption algorithm,
- o a 128-bit nonce,
- o the negotiated hash algorithm H,
- o the client’s certificate.

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 server generates the hash of the client’s certificate, as conveyed during client_cook, in order to calculate the cookie according to Section 5.1. This message contains a concatenated datum, which is encrypted with the client’s public key, according to the encryption algorithm transmitted in the client_cook message. The concatenated datum contains

- o the NTS message ID "server_cook"
- o the version number as transmitted in client_cook,
- o the nonce transmitted in client_cook,
- o the cookie, and

- o a signature, created with the server's private key, calculated over

  - * all of the data listed above, and also

  - * the hash of the client's certificate.

  This signature MUST be calculated according to the transmitted signature algorithm from the client_cook message.

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 message exchange, the usual time synchronization process is executed, with the addition of integrity protection for all messages that the server sends. This message can be repeatedly exchanged as often as the client desires and as long as the integrity of the server's time responses is verified successfully.

6.4.1. Message Type: "time_request"

This message is sent by the client when it requests time exchange. It contains

- o the NTS message ID "time_request",

- o the negotiated version number,

- o a 128-bit nonce,

- o the negotiated hash algorithm H,

- o the hash of the client's certificate under H.

In the case of NTP the data is enclosed in the packet's extension field of type "time request".

6.4.2. Message Type: "time_response"

This message is sent by the server, after it received a time_request message. Prior to this the server MUST recalculate the client's cookie by using the hash of the client's certificate and the transmitted hash algorithm. The message contains

- o the NTS message ID "time_response",

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

In the case of NTP, this is a packet with the necessary time synchronization data and 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 message exchange, 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

- o the NTS message ID "client_bpar",
- o the version number negotiated during association in unicast mode,
- o the client’s hostname, and
- o 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

- o the NTS message ID "server_bpar",
- o the version number as transmitted in the client_bpar message,

- o the one-way function used for building the one-way key chain,
- o the last key of the one-way key chain, and
- o 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.
- o 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

Via this message, 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

- o the NTS message ID "server_broad",
- o the version number that the server’s broadcast mode is working under,
- o time broadcast data,
- o the index that belongs to the current interval (and therefore identifies the current, yet undisclosed key),
- o the disclosed key of the previous disclosure interval (current time interval minus disclosure delay),
- o a MAC, calculated with the key for the current time interval, verifying
  - * the message ID,

* the version number, and

* the time data.

For NTP, this 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 its contents.

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. It MUST keep the transmitted values for version number and algorithms available for later checks.

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 client checks that the message contains a conform version number.

   * It also 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. Again, it MUST remember the transmitted values for version number and algorithms for later checks.

4. It awaits a reply in the form of a server_cert message and performs authenticity checks on the certificate chain and the signature for the version number. If one of these checks fails, the client MUST abort the run.

5. Next, it sends a client_cook message to the server. The client MUST save the included nonce until the reply has been processed.

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 is of the expected format: the concatenation of version number, a 128 bit nonce, a 128 bit cookie and a signature value.

* It verifies that the received version number matches the one negotiated before.

* It verifies that the received nonce matches the nonce sent in the client_cook message.

* It verifies the signature using the server’s public key. The signature has to authenticate the version number, the nonce, the cookie, and the hash of the client’s certificate.

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

7. The client sends a time_request message to the server. The client MUST save the included nonce and the transmit_timestamp (from the time synchronization data) as a correlated pair for later verification steps.

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

   * that the transmitted version number matches the one negotiated before,

   * that the transmitted nonce belongs to a previous time_request message,

   * that the transmit_timestamp in that time_request message matches the corresponding time stamp from the synchronization data received in the time_response, and

   * that the appended MAC verifies the received synchronization data, version number and nonce.

   If at least one of the first three checks fails (i.e. if the version number does not match, if the client has never used the nonce transmitted in the time_response message or if it has used the nonce with initial time synchronization data different from that in the response), then 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 (because the cookie might have changed due to a server seed refresh). If

both checks are successful, the client SHOULD continue time synchronization by going back to step 7.

The client's behavior 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. It MUST remember the transmitted values for version number and signature algorithm.

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 the server's signature cannot be verified, the client MUST abort the broadcast run. If all checks are successful, the client MUST remember all the broadcast parameters received for later checks.

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 and requires the loose time synchronization. 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 MUST restart the broadcast sequence with a client_bpar message Section 6.5.1 if the one-way key chain expires.

The client's behavior in broadcast mode can also be seen in Figure 2.

7.2. The Server

7.2.1. The Server in Unicast Mode

To support unicast mode, the server MUST be ready to perform the following actions:

- 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.

- 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.

- Upon receipt of a client_cook message, the server checks whether it supports the given cryptographic algorithms. It then calculates the cookie according to the formula given in Section 5.1. With this, it MUST construct a server_cook message as described in Section 6.3.2.

- 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.

The server MUST refresh its server seed periodically (see Section 8.1).

7.2.2. The Server in Broadcast Mode

A broadcast server MUST also support unicast mode, in order to provide the initial time synchronization which is a precondition for any broadcast association. To support NTS broadcast, the server MUST additionally 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.

8. Server Seed Considerations

The server has to calculate a random seed which has to be kept secret. The server MUST generate a seed for each supported hash algorithm, see Section 9.1.

8.1. Server Seed Refresh

According to the requirements in [I-D.ietf-tictoc-security-requirements] the server MUST refresh each server seed periodically. As a consequence, the cookie memorized by the client becomes obsolete. In this case the client cannot verify the MAC attachted to subsequent time response messages and has to respond accordingly by re-initiating the protocol with a cookie request (Section 6.3).

8.2. Server Seed Algorithm

8.3. Server Seed Lifetime

9. Hash Algorithms and MAC Generation

9.1. Hash Algorithms

Hash algorithms are used at different points: calculation of the cookie and the MAC, and hashing of the client's certificate. 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:

- o it MUST NOT include SHA-1 or weaker algorithms,
- o it MUST include SHA-256 or stronger algorithms.

Note

> Any hash algorithm is prone to be compromised in the future. A successful attack on a hash algorithm would enable any NTS client to derive the server seed from their own cookie. Therefore, the server MUST have separate seed values for its different supported hash algorithms. This way, knowledge gained from an attack on a hash algorithm H can at least only be used to compromise such clients who use hash algorithm H as well.

9.2. 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 9.1).

10. IANA Considerations

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:

- o The validity of the certificates is preconditioned. Usually this will be the case in corporate networks.
- o The client ensures that the certificates are not revoked. To this end, the client uses the Online Certificate Status Protocol (OCSP) defined in [RFC6277].

- o 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 8.1, 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 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.

11.5. Delay Attack

In a packet delay attack, an adversary with the ability to act as a MITM delays time synchronization packets between client and server asymmetrically [I-D.ietf-tictoc-security-requirements]. This prevents the client to measure the network delay, and hence its time offset to the server, accurately [Mizrahi]. The delay attack does

not modifiy the content of the exchanged synchronization packets. Therefore cryptographic means are not feasible to mitigate this attack. However, serveral non-cryptographic precautions can be taken in order to detect this attack.

- o Usage of multiple time servers: this enables the client to detect the attack provided that the adversary is unable to delay the synchronizations packets between the majority of servers. This approach is commonly used in NTP to exclude incorrect time servers [RFC5905].

- o Multiple communication paths: The client and server are utilizing different paths for packet exchange as described in the I-D [I-D.shpiner-multi-path-synchronization]. The client can detect the attack provided that the adversary is unable to manipulate the majority of the available paths [Shpiner]. Note, that this approach is not yet available, neither for NTP nor for PTP.

- o The introduction of a threshold value for the delay time of the synchronization packets. The client can discard a time server if the packet delay time of this time server is larger than the threshold value.

- o Usage of an encrypted connection: the client exchanges all packets with the time server over an encrypted connection (e.g. IPsec). This measure does not mitigate the delay attack but it makes it more difficult for the adversary to identify the time synchronization packets.

12. Acknowledgements

The authors would like to thank Steven Bellovin, 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-10 (work in progress), July 2014.

[I-D.shpiner-multi-path-synchronization]
Shpiner, A., Tse, R., Schelp, C., and T. Mizrahi, "Multi-Path Time Synchronization", draft-shpiner-multi-path-synchronization-03 (work in progress), February 2014.

[Mizrahi] Mizrahi, T., "A game theoretic analysis of delay attacks against time synchronization protocols", in Proceedings of Precision Clock Synchronization for Measurement Control and Communication, ISPCS 2012, pp. 1-6, September 2012.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

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

[Shpiner] Shpiner, A., Revah, Y., and T. Mizrahi, "Multi-path Time Protocols", in Proceedings of Precision Clock Synchronization for Measurement Control and Communication, ISPCS 2013, pp. 1-6, September 2013.

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 behavior 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
5.4	Protection against DoS attacks	SHOULD	OK
5.5	Replay protection	MUST	OK
5.6	Key freshness.	MUST	OK
	Security association.	SHOULD	OK
	Unicast and multicast associations.	SHOULD	OK
5.7	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
5.9	Protection against Packet Delay and Interception Attacks	SHOULD	NA*)
5.10	Secure mode	MUST	-
	Hybrid mode	SHOULD	-

*) See discussion in section Section 11.5.

Comparison of NTS sepecification against TICTOC security requirements.

Appendix D. Broadcast Mode

For the broadcast mode, NTS adopts the TESLA protocol, which is based on a one-way key chain. This appendix provides details on the generation and usage of the one-way key chain collected and assembled from [RFC4082]. Note that NTS is using the "not re-using keys" scheme of TESLA as described in section 3.7.2. of [RFC4082].

D.1. Server Preparations

Server setup:

1. The server determines a reasonable upper bound B on the network delay between itself and an arbitrary client, measured in milliseconds.

2. It determines the number n+1 of keys in the one-way key chain. This yields the number n of keys that are usable to authenticate broadcast packets. This number n is therefore also the number of time intervals during which the server can send authenticated broadcast messages before it has to calculate a new key chain.

3. It divides time into n uniform intervals I_1, I_2, ..., I_n. Each of these time intervals has length L, measured in milliseconds. In order to fulfill the requirement 3.7.2. of RFC 4082 the time interval L has to be smaller than the time interval between the broadcast messages.

4. The server generates a random key K_n.

5. Using a one-way function F, the server generates a one-way chain of n+1 keys K_0, K_1, ..., K_{n} according to

   K_i = F(K_{i+1}).

6. Using another one-way function F’, it generates a sequence of n+1 MAC keys K’_0, K’_1, ..., K’_{n-1} according to

   K’_i = F’(K_i).

7. Each MAC key K’_i is assigned to the time interval I_i.

8. The server determines the key disclosure delay d, which is the number of intervals between using a key and disclosing it. Note that although security is still provided for all choices d>0, the choice still makes a difference:

   * If d is chosen too short, the client might discards packets because it fails to verify that the key used for their MAC has not been yet disclosed.

   * If d is chosen too long, the received packets have to be buffered for a unnecessarily long time before they can be verified by the client and subsequently be utilized for time synchronization.

The server SHOULD calculate d according to

d = ceil(2*B / L) + 1,

where ceil gives the smallest integer greater than or equal to its argument.

```
< - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                       Generation of Keys

          F              F               F                 F
 K_0  <-------- K_1  <--------  ...  <-------- K_{n-1} <-------- K_n
  |              |                              |                 |
  |              |                              |                 |
  | F'           | F'                           | F'              | F'
  |              |                              |                 |
  v              v                              v                 v
 K'_0           K'_1            ...           K'_{n-1}          K'_n
          [______________|____       ____|_______________|__________]
                I_1             ...            I_{n-1}           I_n

                  Course of Time/Usage of Keys
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - >
```

A Schematic explanation on the TESLA protocol's one-way key chain

D.2. Client Preparation

A client needs the following information in order to participate in a TESLA broadcast.

- o One key K_i from the one-way key chain, which has to be authenticated as belonging to the server. Typically, this will be K_0.

- o The disclosure schedule of the keys. This consists of:

  - * the length n of the one-way key chain,

  - * the length L of the time intervals I_1, I_2, ..., I_n,

  - * the starting time T_i of an interval I_i. Typically this is the starting time T_1 of the first interval;

  - * the disclosure delay d.

- o The one-way function F used to recursively derive the keys in the one-way key chain,

- o The second one-way function F’ used to derive the MAC keys K’_0, K’_1, ... , K’_n from the keys in the one-way chain.

- o An upper bound D_t on how far its own clock is "behind" that of the server.

Note that if D_t is greater than (d - 1) * L, then some authentic packets might be discarded. If D_t is greater than d * L, then all authentic packets will be discarded. In the latter case, the client should not participate in the broadcast, since there will be no benefit in doing so.

D.3. Sending Authenticated Broadcast Packets

During each time interval I_i, the server sends one authenticated broadcast packet P_i. This packet consists of:

- o a message M_i,

- o the index i (in case a packet arrives late),

- o a MAC authenticating the message M_i, with K’_i used as key,

- o the key K_{i-d}, which is included for disclosure.

D.4. Authentication of Received Packets

When a client receives a packet P_i as described above, it first checks that it has not received a packet with associated index i before. This is done to avoid replay/flooding attacks. A packet that fails this test is discarded.

Next, the client checks that, according to the disclosure schedule and with respect to the upper bound D_t determined above, the server cannot have disclosed the key K_i yet. Specifically, it needs to check that the server’s clock cannot read a time that is in time interval I_{i+d} or later. Since it works under the assumption that the server’s clock is not more than D_t "ahead" of the client’s clock, the client can calculate an upper bound t_i for the server’s clock at the time when P_i arrived by

```
t_i = R + D_t,
```

where R is the client’s clock at the arrival of P_i. This implies that at the time of arrival of P_i, the server could have been in interval I_x at most, with

```
x = floor((t_i - T_1) / L),
```

where floor gives the greatest integer less than or equal to its argument. The client now needs to verify that

$$x < i+d$$

is valid (see also section 3.5 of [RFC4082]). If falsified, it is discarded.

Next the client verifies that a newly disclosed key K_{i-d} belongs to the one-way key chain. To this end it verifies identity with some earlier disclosed key by recursively applies the one-way function F to K_{i-d} (see Clause 3.5 in RFC 4082, item 3).

If a packet P_i passes all tests listed above, it is stored for later authentication. Also, if at this time there is a package with index i-d already buffered, then the client uses the disclosed key K_{i-d} to derive K'_{i-d} and uses that to check the MAC included in package P_{i-d}. On success, it regards M_{i-d} as authenticated.

Appendix E. Random Number Generation

At various points of the protocol, the generation of random numbers is required. The employed methods of generation need to be cryptographically secure. See [RFC4086] for guidelines concerning this topic.

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

Phone: +49-(0)531-592-8421
Email: kristof.teichel@ptb.de

TICTOC Working Group
Internet-Draft
Intended status: Experimental
Expires: April 18, 2016

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 encapsuled timing messages inside an Ethernet pseudowire.

Status of This Memo

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

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

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

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

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Problem Statement 5
4. Timing over MPLS Architecture 5
5. Dedicated LSPs for Timing messages 7
6. Timing over LSP Encapsulation 8
6.1. Timing over UDP/IP over MPLS Encapsulation 8
6.2. Timing over PW Encapsulation 8
7. Timing message Processing 9
8. Protection and Redundancy 10
9. ECMP and Entropy 10
10. PHP 11
11. OAM, Control and Management 11
12. QoS Considerations 11
13. FCS and Checksum Recalculation 11
14. Behavior of LER/LSRs 12
14.1. Behavior of Timing-capable/aware LERs/LSRs 12
14.2. Behavior of non-Timing-capable/aware LSR 12
15. Other considerations 13
16. Security Considerations 13
17. Applicability Statement 14
18. Acknowledgements 14
19. IANA Considerations 14
20. References 15
20.1. Normative References 15
20.2. Informative References 16
Appendix A. Appendix 17
A.1. Routing extensions for Timing-aware Routers 17
A.2. Signaling Extensions for Creating Timing LSPs 17

Authors' Addresses 18

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 Doug Arnold
Internet Draft Meinberg-USA
Intended status: Standards Track Heiko Gerstung
Meinberg

Expires: January 2, 2015

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

draft-ietf-tictoc-ptp-enterprise-profile-03.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on January 2, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

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

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
3. Technical Terms 3
4. Problem Statement 5
5. Network Technology 6
6. Time Transfer and Delay Measurement 7
7. Default Message Rates 7
8. Requirements for Master Clocks 7
9. Requirements for Slave Clocks 9
10. Requirements for Transparent Clocks 9
11. Management and Signaling Messages 9
12. Forbidden PTP Options 10
13. Interoperation with Other PTP Profiles 10
14. Security Considerations 10
15. IANA Considerations 11
16. References 11
16.1. Normative References 11
16.2. Informative References 11
17. Acknowledgments 11
18. Authors addresses 12

1. Introduction

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

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

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

2. Conventions used in this document

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

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

3. Technical Terms

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Problem Statement

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

options it is necessary to create a profile for enterprise networks to achieve interoperability between equipment manufactured by different vendors.

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

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

5. Network Technology

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

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

Similar to NTP, PTP makes the assumption that the one way network delay for Sync Messages and Delay Response Messages are the same. When this is not true it can cause errors in the transfer of time from the Master to the Slave. It is up to the system integrator to design the network so that such effects do not prevent the PTP system from meeting the timing requirements. The details of network asymmetry are outside the scope of this document. See for example, [G8271].

6. Time Transfer and Delay Measurement

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

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

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

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

7. Default Message Rates

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

8. Requirements for Master Clocks

Master clocks SHALL obey the standard Best Master Clock Algorithm from [IEEE1588]. PTP systems using this profile MAY support multiple simultaneous Grandmasters as long as each active Grandmaster is operating in a different PTP domain. When Preferred Master Clocks are not the Best Master in one domain, they SHOULD operate in another domain when they. Using multiple masters can

mitigate rogue master and man-in-the-middle attacks such as delay attacks, packet interception / manipulation attacks. Assuming the path to each master is different, an attacker would have to attack more than one path simultaneously.

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

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

TLV Type (Enumeration16): ORGANIZATION_EXTENSION value = 0003 hex

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

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

IETF Profile number (UInteger8): value = 1

Revision number (UInteger8): value = 1

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

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

Maximum Phase Adjustment Units (Enumeration8):

```
          Value 0 = unknown
          Value 1 = seconds
          Value 3 = milliseconds
          Value 6 = microseconds
          Value 9 = nanoseconds
          Value 12 = picoseconds
          Value 15 = femtoseconds
          All other values reserved for future use
```

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

9. Requirements for Slave Clocks

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

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

10. Requirements for Transparent Clocks

Transparent clocks SHALL NOT change the transmission mode of an Enterprise profile PTP message. For example a Transparent clock SHALL NOT change a unicast message to a multicast message. Transparent clocks SHALL NOT alter the Enterprise Profile TLV of an Announce message, or any other part of an Announce message. Transparent Clocks SHOULD support multiple domains.

11. Management and Signaling Messages

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

12. Forbidden PTP Options

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

13. Interoperation with Other PTP Profiles

Clocks operating in the Enterprise profile will not interoperate with clocks operating in the Power Profile [C37.238], because the

Enterprise Profile requires the End to End Delay Measurement Mechanism and the Power Profile requires the Peer to Peer Delay Measurement Mechanism.

Clocks operating in the Enterprise profile will not interoperate with clocks operating in the Telecom Profile for Frequency Synchronization[G8265.1], because the Enterprise Profile forbids Unicast Message Negotiation, and Unicast Discovery. These features are part of the default manner of operation for the Telecom Profile for Frequency Synchronization.

Clocks operating in the Enterprise profile will interoperate with clocks operating in the default profile if the default profile clocks operate on IPv4 or IPv6 networks, use the End to End Delay Measurement Mechanism, and use management messages in unicast when those messages are directed at a specific clock. If any of these requirements are not met than Enterprise Profile clocks will not interoperate with Default Profile Clocks. The Default Profile is described in Annex J of [IEEE1588].

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

14. Security Considerations

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

15. IANA Considerations

There are no IANA requirements in this specification.

16. References

16.1. Normative References

[IEEE1588] IEEE std. 1588-2008, "IEEE Standard for a Precision Clock Synchronization for Networked Measurement and Control Systems." July, 2008.

[RFC768] Postel, J., "User Datagram Protocol," RFC 768, August, 980.

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

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

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

16.2. Informative References

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

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

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

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

17. Acknowledgments

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

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

18. Authors' Addresses

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

Email: doug.arnold@meinberg-usa.com

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

Email: Heiko.gerstung@meinberg.de

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2015

G. Mirsky
S. Ruffini
Ericsson
J. Drake
Juniper Networks
S. Bryant
Cisco Systems
A. Vainshtein
ECI Telecom
June 30, 2014

Residence Time Measurment in MPLS network
draft-mirsky-mpls-residence-time-02

Abstract

This document specifies G-ACh based Residence Time Measurement and how it can be used by time synchronization protocols being transported over MPLS domain.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 1, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
1.1. Conventions used in this document 2
1.1.1. Terminology 2
1.1.2. Requirements Language 3
2. Residence Time Measurement 3
3. G-ACh for Residence Time Measurement 4
4. Control Plane Theory of Operation 5
4.1. RSVP-TE Control Plane Operation to Support RTM 5
5. Data Plane Theory of Operation 6
6. Applicable PTP Scenarios 6
7. IANA Considerations 7
7.1. New RTM G-ACh 7
7.2. New RTM TLV Registry 7
8. Security Considerations 7
9. Acknowledgements 8
10. References 8
10.1. Normative References 8
10.2. Informative References 9
Authors' Addresses 9

1. Introduction

Time synchronization protocols, Network Time Protocol version 4 (NTPv4) [RFC5905] and Precision Time Protocol (PTP) Version 2, a.k.a. IEEE-1588 v.2, can be used to syncronized clocks across network domain. In some scenarios calculation of time packet of time syncronization protocol spends within a node, called Residence Time, can improve accuracy of clock syncronization. This document defines new Generalized Associated Channel (G-ACh) that can be used in Multi-Protocol Label Switching (MPLS) network to measure Residence Time over Label Switched Path (LSP). Transport of packets of a time synchronization protocol over MPLS domain is outside of scope of this document.

1.1. Conventions used in this document

1.1.1. Terminology

MPLS: Multi-Protocol Label Switching

ACH: Associated Channel

TTL: Time-to-Live

G-ACh: Generic Associated Channel

GAL: Generic Associated Channel Label

NTP: Network Time Protocol

ppm: part per million

PTP: Precision Time Protocol

LSP: Label Switched Path

LSR: Label Switched Router

OAM: Operations, Administration, and Maintenance

RTM: Residence Time Measurement

IGP: Internal Gateway Protocol

1.1.2. 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 [RFC2119].

2. Residence Time Measurement

Packet Loss and Delay Measurement for MPLS Networks [RFC6374] can be used to measure one-way or two-way end-to-end propagation delay over LSP or PW. But none of these metrics is useful for time syncronization across a network. For example, PTPv2 uses "residence time", time it takes for a PTPv2 event packet to transit a node. The residence times are accumulated in the correctionField of the PTP event messages or of the associated follow-up messages (or Delay_Resp message associated with the Delay_Req message) in case of two-step clocks. The residence time values are specific to each output PTP port and message.

Note the delay of propagation over a link connected to a port receiving the PTP event message is handled by IEEE 1588 [IEEE.1588.2008] by means of specific messages, Pdelay_Req and Pdelay_Resp,or Delay_Req and Delay_Resp depending on the applicable delay mechanism, peer-to-peer or delay request-response mechanism respectively.

This document proposes mechanism to accumulate packet residence time from all LSRs that support the mechanism across the particular LSP.

3. G-ACh for Residence Time Measurement

RFC 5586 [RFC5586] and RFC 6423 [RFC6423] extended applicability of PW Associated Channel (ACH) [RFC5085] to LSPs. G-ACh presents mechanism to transport OAM and other control messages and trigger their processing by arbitrary transient LSRs through controlled use of Time-to-Live (TTL) value.

Packet format for Residence Time Measurement (RTM) presented in Figure 1

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 1|Version|   Reserved    |          RTM Channel          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Scratch Pad                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type               |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Value                              |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: G-ACh packet format for Residence Time Measurement

The Version field is set to 0, as defined in RFC 4385 [RFC4385]. The Reserved field must be set to 0 on transmit and ignored on receipt. The RTM G-ACh field, value to be allocated by IANA, identifies the packet as such. The Scratch Pad field is 8 octets in length and is used to accumulate the residence time spent in LSRs transited by the packet on its path from ingress LSR to egress LSR. Its format is IEEE double precision and its units are nanoseconds.

The Type field identifies type of Value that the TLV carries. IANA will be asked to create sub-registry in Generic Associated Channel (G-ACh) Parameters Registry called "MPLS RTM TLV Registry". The Length field is number of octets of the Value field. The optional Value field may be used to carry a packet of a given time synchronization protocol. If the packet carried in the RTM message, then it accordingly identified by distinct Type, and may be NTP [RFC5905] or PTP [IEEE.1588.2008]. It is important to note that the packet may be authenticated or encrypted and carried over MPLS LSP

edge to edge unchanged while residence time being accumulated in the Scratch Pad field. The TLV MUST be included in the RTM.

4. Control Plane Theory of Operation

A router will announce its support for RTM in a new sub-TLV, the RTM Capable TLV which will be defined in a subsequent version of this document, for the router capabilities TLV defined in RFC 4970 (OSPF) [RFC4970] and RFC 4971 (IS-IS) [RFC4971].

The operation of RTM depends upon TTL expiry to deliver an RTM packet from one RTM capable LSR to the next along the path from ingress LSR to egress LSR, which means that an RTM capable LSR needs to be able to compute a TTL which will cause the expiry of an RTM packet on the next RTM capable LSR.

However, because of Equal Cost Multipath, labels distributed by LDP do not instantiate a single path between a given ingress/egress LSR pair but rather a graph and different flows will take different paths through this graph. This means one doesn’t know the path that RTM packets will take or even if they all take the same path. So, in an environment in which not all routers in an IGP domain support RTM, it is effectively impossible to use TTL expiry to deliver RTM packets and hence RTM cannot be used for LSPs instantiated using LDP. In the special but important case of environment in which all routers in an IGP domain support RTM, setting the TTL to 1 will always cause the expiry of an RTM packet on the next RTM capable downstream LSR and hence in such an environment, RTM can be used for LSPs instantiated using LDP.

Generally speaking, RTM is more useful for an LSP instantiated using RSVP-TE [RFC3209] because the LSP’s path can be known.

4.1. RSVP-TE Control Plane Operation to Support RTM

An ingress LSR that wishes to perform RTM along a path through an MPLS network to an egress LSR verifies that the selected egress LSR supports RTM via the egress LSR’s advertisement of the RTM Capable TLV. In the Path message that the ingress LSR uses to instantiate the LSP to that egress LSR it places initialized Record Route and RTM Set (see below) Objects, which tell the egress LSR that RTM is desired for this LSP.

In the Resv message that the egress LSR sends in response to the received Path message, it includes initialized Record Route and RTM Set objects. The latter object will be defined in a subsequent version of this document and it contains an ordered list, from egress LSR to ingress LSR, of the RTM capable LSRs along the LSP’s path.

Each such LSR will use the use the ID of the first LSR in the RTM Set Object in conjunction with the Record Route Object to compute the hop count to its downstream RTM capable LSR. It will also insert its ID at the beginning of the RTM Set Object before forwarding the Resv upstream.

After the ingress LSR receives the Resv, it will begin sending RTM packets to the first RTM capable LSR on the LSP's path. Each RTM packet has its Scratch Pad field initialized and its TTL set to expire on that LSR.

It should be noted that RTM can also be used for LSPs instantiated using [RFC3209] in an environment in which all routers in an IGP support RTM. In this case the RTM Set Object is not used.

5. Data Plane Theory of Operation

After instantiating an LSP for a path using RSVP-TE [RFC3209] as described in Section 4.1 or if this is the special case of homogeneous RTM-capable IP/MPLS domain discussed in the last paragraph of Section 4, ingress LSR MAY begin sending RTM packets to the first RTM capable downstream LSR on that path. Each RTM packet has its Scratch Pad field initialized and its TTL set to expire on the next downstream LSR. Each RTM capable LSR that receives an RTM packet records the time at which it receives that packet as well as the time at which it transmits that packet; this should be done as close to the physical layer as possible. Just prior to sending that packet, it takes the difference between those two times and adds it to the value in the Scratch Pad field. Note, for the purpose of calculating a residence time, a free running clock may be sufficient, as, for example, 4.6 ppm accuracy leads to 4,6 ns error for residence time in the order of 1 ms.

The RTM capable LSR also sets the RTM packet's TTL to expire on the next RTM capable downstream from it LSR.

The egress LSR may then use the value in the Scratch Pad field to perform time correction. For example, the egress LSR may be a PTP Boundary Clock synchronized to a Master Clock and will use the value in the Scratch Pad Field to update PTP's Correction Field.

6. Applicable PTP Scenarios

The proposed approach can be directly integrated in a PTP network based on delay request-response mechanism. The RTM capable LSR nodes act as end-to-end transparent clocks, and typically boundary clocks, at the edges of the MPLS network, use the value in the Scratch Pad

field to update the correctionField of the corresponding PTP event packet prior to performing the usual PTP processing.

Under certain assumptions the proposed solution in a network where peer delay mechanism is used is also possible. The solution in this case requires the definition of a specific protocol to be used to calculate the link delays according to a peer delay link measurement approach. This is not described in this version of the draft.

7. IANA Considerations

7.1. New RTM G-ACh

IANA is requested to reserve a new G-ACh as follows:

Value	Description	Reference
X	Residence Time Measurement	This document

Table 1: New Residence Time Measurement

7.2. New RTM TLV Registry

IANA is requested to create sub-registry in Generic Associated Channel (G-ACh) Parameters Registry called "MPLS RTM TLV Registry". All code points within this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC5226] This document defines the following new values RTM TLV type

Value	Description	Reference
0	Reserved	This document
TBD1	No payload	This document
TBD2	PTPv2	This document
TBD3	NTP	This document

Table 2: RTM TLV Type

8. Security Considerations

Routers that support Residence Time Measurement are subject to the same security considerations as defined in [RFC5586] and [RFC6423].

9. Acknowledgements

TBD

10. References

10.1. Normative References

[IEEE.1588.2008] "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588, March 2008.

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

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.

[RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

[RFC5085] Nadeau, T. and C. Pignataro, "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.

[RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", RFC 5586, June 2009.

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

[RFC6423] Li, H., Martini, L., He, J., and F. Huang, "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, November 2011.

10.2. Informative References

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, September 2011.

Authors' Addresses

Greg Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

Stefano Ruffini
Ericsson

Email: stefano.ruffini@ericsson.com

John Drake
Juniper Networks

Email: jdrake@juniper.net

Stewart Bryant
Cisco Systems

Email: stbryant@cisco.com

Alexander Vainshtein
ECI Telecom

Email: Alexander.Vainshtein@ecitele.com