Internet Engineering Task Force M. Lichvar
Internet-Draft Red Hat
Intended status: Standards Track Jun 24, 2021
Expires: December 26, 2021

NTP Over PTP
draft-mlichvar-ntp-over-ptp-00

Abstract

This document specifies a transport for the Network Time Protocol (NTP) client-server mode using the Precision Time Protocol (PTP) to enable hardware timestamping on hardware that can timestamp PTP messages but not NTP messages.

1. Introduction

The Precision Time Protocol (PTP) [IEEE1588] was designed for highly accurate synchronization of clocks in a network. It relies on hardware timestamping supported in network devices (e.g. interface controllers, switches, and routers) to eliminate the impact of processing and queueing delays on PTP measurements.

PTP was originally designed for multicast communication. Later was added a unicast mode, which can be used in larger networks with partial on-path PTP support (e.g. telecom profiles G.8265.1 and G.8275.2).

The Network Time Protocol [RFC5905] does not rely on hardware timestamping support, but implementations can use it if it is available to avoid the impact of processing and queueing delays, similarly to PTP. The client-server mode of NTP is functionally similar to the PTP unicast mode.

An issue for NTP is hardware that can specifically timestamp only PTP packets. This limitation comes from their design, which does not allow the timestamps to be captured or retrieved at the same rate as packets can be received or transmitted. A filter needs to be implemented in the hardware to inspect each packet and timestamp only those that actually need it. The filter can be usually configured for the PTP transport (e.g. UDPv4, UDPv6, 802.3) and sometimes even the message type (e.g. sync message or delay request) to further reduce the rate of timestamps on the server or client side. This limitation prevents hardware timestamping of NTP messages. It also prevents timestamping of PTP messages if they are secured at the transport layer or below (e.g. IPSec or MACSec).

This document specifies a new transport for NTP to enable the PTP-specific timestamping support. It adds a new extension field (TLV) for PTP to contain NTP messages.

NTP over PTP does not disrupt normal operation of PTP. A network and even a single host can support both at the same time.

The specification does not take advantage of the PTP correctionField modified by PTP transparent clocks as their support for the unicast mode seems to be rare or nonexistent.

The client/server mode of NTP, even if using the PTP transport, has several advantages when compared to the PTP unicast mode:

o It is more secure. It can use existing security mechanisms specified for NTP like Network Time Security [RFC8915], not losing any of its features. The PTP unicast mode allows an almost-infinite traffic amplification, which can be exploited for denial-of-service attacks and can only be limited by security mechanisms using client authentication.

- It needs fewer messages and less network bandwith to get the same number of timestamps.

- It is better suited for synchronization in networks without full on-path support. It does not assume the network delay is constant and the number of measurements in opposite directions is symmetric (in PTP sync messages and delay requests have independent timing).

1.1. 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].

2. PTP transport for NTP

A new TLV is defined for PTP to contain NTP messages in the client and server mode. Using other NTP modes in the TLV is not specified. Any transport specified for PTP that supports unicast messaging can be used for NTP over PTP, e.g. UDP on IPv4 and IPv6.

The type value of the NTP TLV is TBD. The TLV contains the whole NTP message as would normally be the UDP payload, without any modifications. The TLV does not propagate through boundary clocks.

If the UDP transport is used for PTP, the UDP source and destination port numbers MUST be the PTP event port (319). Client port randomization would break the timestamping.

The NTP TLV MUST be included in a delay request message. The originTimestamp field and all fields of the header SHOULD be zero, except:

- messageType is 1 (delay request)

- versionPTP is 2

- messageLength is the length of the PTP message including the NTP TLV

- domainNumber is TBD

- flagField has the unicastFlag bit set

An NTP client using the PTP transport sends NTP requests in PTP messages to the server at the same rate as it would normally send them over UDP.

A server which supports the NTP TLV MUST check for the domainNumber of TBD and respond to an NTP request with a single PTP message containing the NTP response using the same PTP message format. It MUST NOT send a delay response message.

A server which does not support the NTP TLV will not recognize the domain number and ignore the message. If it responded to messages in the domain (e.g. due to misconfiguration), it would send a delay response (to port 320 if using the UDP transport), which would be ignored by the client.

Any authenticator fields included in the NTP messages MUST be calculated only over the NTP message following the header of the NTP TLV.

Timestamps SHOULD NOT be adjusted for the beginning of the NTP data in the PTP message. They SHOULD still correspond to the ending of the transmission and beginning of the reception (e.g. start of delimiter in the Ethernet frame).

Any modifications of the correctionField made by potential one-step end-to-end transparent clocks in the network SHOULD be ignored by the server and client.

3. Security Considerations

The PTP transport prevents NTP clients from randomizing their source port. It has no other impact on security of NTP.

4. References

4.1. Normative References

[IEEE1588]
Institute of Electrical and Electronics Engineers, "IEEE std. 1588-2019, "IEEE Standard for a Precision Clock Synchronization for Networked Measurement and Control Systems."", 11 2019, <https://www.ieee.org>.

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

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

4.2. Informative References

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <https://www.rfc-editor.org/info/rfc8915>.

Author's Address

Miroslav Lichvar
Red Hat
Purkynova 115
Brno 612 00
Czech Republic

Email: mlichvar@redhat.com