Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: October 27, 2019

M. Lichvar
Red Hat

April 25, 2019

NTP Correction Field
draft-mlichvar-ntp-correction-field-04

Abstract

This document specifies an extension field for the Network Time Protocol (NTP) which improves resolution of specific fields in the NTP header and allows network devices such as switches and routers to modify NTP packets with corrections to improve accuracy of the synchronization in the network.

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 https://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 October 27, 2019.

Copyright Notice

1. Introduction

Processing and queueing delays in network switches and routers may be a significant source of jitter and asymmetry in network delay, which has a negative impact on accuracy and stability of clocks synchronized by NTP [RFC5905].

If all network devices on the paths between NTP clients and servers implemented NTP and supported an operation as a server and client, the impact of the delays could be avoided by configuring NTP to make measurements only between devices and hosts that are directly connected to one another. In the Precision Time Protocol (PTP) [IEEE1588], which is a different protocol for synchronization of clocks in networks, such devices are called Boundary Clocks (BC).

A different approach supported by PTP to improve the accuracy uses Transparent Clocks (TC). Instead of fully implementing PTP in order to support an operation as a BC, the devices only modify a correction field in forwarded PTP packets with the time that the packets had to wait for transmission. The final value of the correction is included in the calculation of the delay and offset, which may significantly improve the accuracy and stability of the synchronization.

This document describes an NTP extension field which allows the devices to make a similar correction in forwarded NTP packets.

To better support a highly accurate synchronization, the extension field also improves resolution of the receive and transmit timestamps from the NTP header.

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. Format of Correction Field

The Correction Field is an NTP extension field following RFC 7822 [RFC7822]. The format of the extension field is shown in Figure 1.

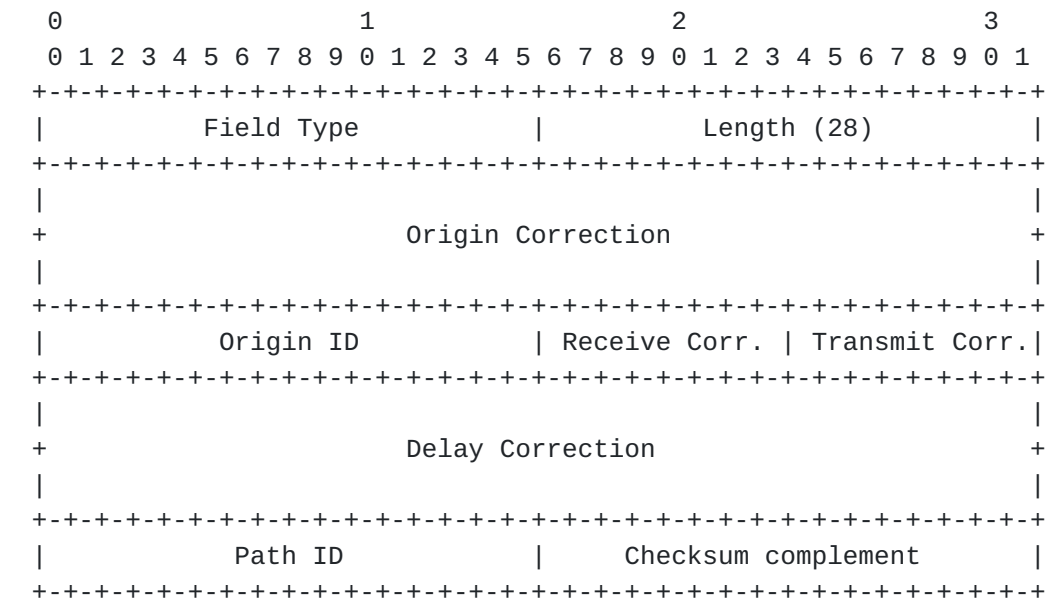

Figure 1: Format of Correction Field

The extension field has the following fields:

Field Type
: The type which identifies the Correction extension field. TBD

Length
: The length of the extension field, which is 28 octets.

Origin Correction
: A field which contains a copy of the final delay correction from the previous packet in the NTP exchange.

Origin ID
: A field which contains a copy of the final path ID from the previous packet in the NTP exchange.

Receive Correction
: An 8-bit extension of the receive timestamp in the NTP header increasing its resolution. The extended receive timestamp has 32 integer bits and 40 fractional bits.

Transmit Correction
: An 8-bit extension of the transmit timestamp in the NTP header increasing its resolution. The extended transmit timestamp has 32 integer bits and 40 fractional bits.

Delay Correction

A signed fixed-point number of nanoseconds with 48 integer bits and 16 fractional bits, which represents the current correction of the network delay that has accumulated for this packet on the path from the source to the destination. The format of this field is identical to the PTP correctionField.

Path ID
: A 16-bit identification number of the path where the delay correction was updated.

Checksum Complement
: A field which can be modified in order to keep the UDP checksum of the packet valid. This allows the UDP checksum to be transmitted before the Correction Field is received and modified. The same field is described in RFC 7821 [RFC7821].

3. Network devices

A network device which is forwarding a packet and supports the Correction Field MUST NOT modify the packet unless all of the following applies:

1. The packet is an IPv4 or IPv6 UDP packet.

2. The source port or destination port is 123.

3. The NTP version is 4.

4. The NTP mode is 1, 2, 3, 4, or 5.

5. The format of the packet is valid per RFC 7822.

6. The packet contains an extension field which has a type of TBD and length of 28 octets.

The device SHOULD add to the current value in the delay correction field the length of an interval between the reception and transmission of the packet. If the packet is transmitted at the same speed as it was received and the length of the packet does not change (e.g. due to adding or removing a VLAN tag), the beginning and end of the interval may correspond to any point of the reception and transmission as long as it is consistent for all forwarded packets of the same length. If the transmission speed or length of the packet is different, the beginning and end of the interval SHOULD correspond to the end of the reception and beginning of the transmission respectively.

If the transmission starts before the reception ends, a negative value may need to be added to the delay correction. The end of the reception SHOULD be determined using the length field of the UDP header and the speed at which the packet is received.

If the device updates the delay correction, it SHOULD also add the identification numbers of the incoming and outgoing port to the path ID.

If the device modified any field of the extension field, it MUST update the checksum complement field in order to keep the current UDP checksum valid, or update the UDP checksum itself.

4. NTP hosts

When an NTP client sends a request to a server and the association is configured to use the Correction Field, it SHOULD add the extension field to the packet. All fields of the extension field except type and length SHOULD be set to zero.

When the server receives a packet which includes the extension field, the response SHOULD also include the extension field.

If the server's clock has a better precision than resolution of the 64-bit NTP timestamp format, the server SHOULD save the additional bits in the receive and transmit correction fields and set the precision field to the corresponding number, which is smaller than -32. Otherwise, the receive and transmit correction fields SHOULD be zero.

The origin correction and origin ID fields SHOULD be set to the delay correction and path ID from the request. The other fields of the Correction Field SHOULD be zero.

When the client receives a response which contains the extension field, it SHOULD check the value of both the origin and delay correction fields. If a correction is larger than a specified maximum (e.g. 1 second), the extension field SHOULD be ignored.

The client MAY log a warning if the origin ID and path ID are not equal, which indicates the network path between the server and client is not symmetric.

If the client's clock has a better precision than resolution of the 64-bit NTP format and the precision field in the response contains a number smaller than -32, the client SHOULD extend the receive and transmit timestamp from the NTP header with the additional bits from the receive and transmit correction fields respectively.

When the client calculates the offset and delay using the formulas from RFC 5905, the origin correction is subtracted from the receive timestamp and the delay correction is added to the transmit timestamp. A conversion is necessary as the corrections are in different units than the timestamps (nanoseconds vs seconds).

An NTP peer follows the rules of both servers and clients. It processes Correction Fields in received packets as a client and sends Correction Fields as a server. A packet which has a zero origin timestamp (i.e. it is not a response to a request) SHOULD have a zero origin correction and zero origin ID in the Correction Field.

A broadcast server using the Correction Field SHOULD always set the origin correction and origin ID fields to zero.

5. Acknowledgements

The Correction Field extension is based on the PTP correction field specified in IEEE 1588-2008.

The author would like to thank Tal Mizrahi and Harlan Stenn for their useful comments.

6. IANA Considerations

IANA is requested to allocate an Extension Field Type for the Correction Field.

7. Security Considerations

NTP packets including the Correction Field cannot be authenticated by a legacy MAC, because the MAC has to cover all extension fields in the packet and devices which are supposed to modify the field are not able to update the MAC.

It is recommended to authenticate NTP packets using an authentication extension field, e.g. the NTS Authenticator and Encrypted Extensions [I-D.ietf-ntp-using-nts-for-ntp] extension field, and add the Correction Field to the packet after the authentication field.

A man-in-the-middle attacker can delay packets in the network in order to increase the measured delay and shift the measured offset by up to half of the extra delay. If the packets contain the Correction Field, the attacker can reduce the delay calculated by the client or peer and shift the offset even more. The maximum correction should be limited (e.g. to 1 second) to prevent the attacker from injecting a larger offset to the measurements.

8. References

8.1. Normative References

[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>.

[RFC7822] Mizrahi, T. and D. Mayer, "Network Time Protocol Version 4 (NTPv4) Extension Fields", RFC 7822, DOI 10.17487/RFC7822, March 2016, <https://www.rfc-editor.org/info/rfc7822>.

8.2. Informative References

[I-D.ietf-ntp-using-nts-for-ntp] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", draft-ietf-ntp-using-nts-for-ntp-18 (work in progress), April 2019.

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

[RFC7821] Mizrahi, T., "UDP Checksum Complement in the Network Time Protocol (NTP)", RFC 7821, DOI 10.17487/RFC7821, March 2016, <https://www.rfc-editor.org/info/rfc7821>.

Author's Address

Miroslav Lichvar
Red Hat
Purkynova 115
Brno 612 00
Czech Republic

Email: mlichvar@redhat.com