NTP Working Group   Tal
Mizrahi
Internet Draft
Marvell
Intended status: Experimental
Expires: October 2014   April 15,
2014

Using UDP Checksum Trailers in the Network Time Protocol (NTP)
draft-mizrahi-ntp-checksum-trailer-02.txt

Abstract

The Network Time Protocol (NTP) allows clients to synchronize to a time server using timestamped protocol messages. To faciliate accurate timestamping, some implementations use hardware-based timestamping engines that integrate the accurate transmission time into every outgoing NTP packet during transmission. Since these packets are transported over UDP, the UDP checksum field is then updated to reflect this modification. This document proposes an extension field that includes a 2-octet Checksum Trailer, allowing timestamping engines to reflect the checksum modification in the last 2 octets of the packet rather than in the UDP checksum field. The behavior defined in this document is interoperable with existing NTP implementations.

Table of Contents

1. Introduction

The Network Time Protocol [NTPv4] allows clients to synchronize their clocks to a time server by exchanging NTP packets. The increasing demand for highly accurate clock synchronization motivates implementations that provide accurate timestamping.

1.1. Intermediate Entities

In this document we use the term 'intermediate entity', referring to an entity that reside on the path between the sender and the receiver of an NTP packet, that modifies this NTP packet en-route. Two examples of intermediate entities are presented below.

In order to facilitate accurate timestamping, an implementation MAY use a hardware based timestamping engine, as shown in Figure 1. In such cases, NTP packets are sent and received by a software layer, whereas a timestamping engine modifies every outgoing NTP packet by incorporating its accurate transmission time into the <Transmit Timestamp> field in the packet.

```
                           NTP client/server
                         +-------------------+
                         |                   |
                         |   +-----------+   |
       Software          |   |    NTP    |   |
                         |   | protocol  |   |
                         |   +-----+-----+   |
                         |         |         |
                         |   +-----+-----+   |
                         |   | Accurate  |   |
       ASIC/FPGA         |   | Timestamp |   |
                         |   |  engine   |   |
                         |   +-----------+   |
                         |         |         |
                         +---------+---------+
                                   |
                                   |NTP packets
                                   |
                             ___   v  _
                            /   \_/ \__
                           /           \_
                          /     IP       /
                          \_  Network   /
                           /           \
                           \__/\_   ___/
                                 \_/
```

Figure 1 Accurate Timestamping in NTP

The accuracy of clock synchronization over packet networks is highly sensitive to delay jitters in the underlying network, which dramatically affects the clock accuracy. To address this challenge, the Precision Time Protocol (PTP) [IEEE1588] defines Transparent Clocks (TCs), intermediate switches and routers that improve the end-to-end accuracy by updating a "Correction Field" in the PTP packet by adding the latency caused by the current TC. In NTP no equivalent entity is currently defined, but future versions of NTP may define an intermediate node that modifies en-route NTP packets using a "Correction Field".

1.2. Updating the UDP Checksum

When the UDP payload is modified by an intermediate entity, the UDP Checksum field needs to be updated to maintain its correctness. When using UDP over IPv4 ([UDP]), an intermediate entity can assign a value of zero in the checksum field, causing the receiver to ignore the checksum field. UDP over IPv6, as defined in [IPv6], does not allow a zero checksum, and requires the UDP checksum field to contain a correct checksum of the UDP payload.

Since an intermediate entity only modifies a specific field in the packet, i.e. the timestamp field, the UDP checksum update can be performed incrementally, using the concepts presented in [Checksum].

A similar problem is addressed in Annex E of [IEEE1588]. When the Precision Time Protocol (PTP) is transported over IPv6, two octets are appended to the end of the PTP payload for UDP checksum updates. The value of these two octets can be updated by an intermediate entity, causing the value of the UDP checksum field to remain correct.

This document defines a similar concept for [NTP], allowing intermediate entities to update NTP packets and maintain the correctness of the UDP checksum by modifying the last 2 octets of the packet. This is performed by adding an NTP extension field at the end of the packet, in which the last two bytes are used as a checksum trailer.

The term Checksum Trailer is used throughout this document and refers to the 2 octets at the end of the UDP payload, used for updating the UDP checksum by intermediate entities.

The usage of the Checksum Trailer can in some cases simplify the implementation, since if the packet data is processed in a serial order, it is simpler to first update the timestamp field, and then update the Checksum Trailer rather than to update the timestamp and then update the UDP checksum, residing at the UDP header.

2. Conventions used in this document

2.1. Terminology

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

2.2. Abbreviations

MAC   Message Authentication Code

NTP   Network Time Protocol

PTP   Precision Time Protocol

UDP   User Datagram Protocol

3. Using UDP Checksum Trailers in NTP

3.1. Overview

The UDP Checksum Trailer is a two-octet trailer that is appended at the end of the UDP payload using an NTP extension field. Figure 2 illustrates the packet format of an NTP packet with a Checksum Trailer extension. The figure illustrates an unauthenticated NTP packet. Section 3.4. provides further details about using the Checksum Trailer in authenticated packets.

```
                 +--------------------------------+
                 |       IPv4 / IPv6 Header       |
                 +--------------------------------+
                 |           UDP Header           |
                 +--------------------------------+
         ^       |                                |
         |       |           NTP packet           |
         |       |                                |
         |       +--------------------------------+
        UDP      | Optional NTP Extension Fields  |
      Payload    +--------------------------------+
         |       |      UDP Checksum Trailer      |
         |       |  Extension Field (28 octets)   |
         v       +--------------------------------+
```

Figure 2 Checksum Trailer in NTP Unauthenticated Packets

3.2. Checksum Trailer in NTP Packets

NTP is transported over UDP, either over IPv4 or over IPv6. This document applies to both NTP over IPv4, and NTP over IPv6.

NTP packets may include one or more extension fields, as defined in [NTPv4]. The Checksum Trailer in NTP packets resides in a dedicated NTP extension field, as shown in Figure 2.

In unauthenticated mode, if the NTP packet includes more than one extension field, the Checksum Trailer extension is always the last extension field. Thus, when NTP authentication is disabled, the Checksum Trailer is the last 2 octets in the UDP payload, and thus the trailer is located at UDP Length - 2 octets after the beginning of the UDP header.

When NTP authentication is enabled, the Checksum Trailer is the last 2 octets before the Message Authentication Code (MAC).

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Field Type           |      Length = 28 octets       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            Padding                            |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |       Checksum Trailer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3 NTP Checksum Trailer Extension Field

Field Type

A dedicated Field Type value is used to identify the Checksum Trailer extension. See Section 6 for further details.

Length

The Checksum Trailer extension field length is 28 octets.

This length guarantees that the host that receives the packet parses it correctly, whether the packet includes a MAC or not. [NTP-Ext] provides further details about the length of an extension field in the absence of a MAC.

Padding

The extension field includes 22 octets of padding. This field SHOULD be set to 0, and SHOULD be ignored by the recipient.

Checksum Trailer

Includes the UDP Checksum Trailer field.

3.2.1. Transmission of NTP with Checksum Trailer

The transmitter of an NTP packet MAY include a Checksum Trailer extension field.

3.2.2. Intermediate Updates of NTP with Checksum Trailer

An intermediate node that receives and alters an NTP packet containing a Checksum Trailer extension MAY use the Checksum Trailer to maintain a correct UDP checksum value.

3.2.3. Reception of NTP with Checksum Trailer

This document does not impose new requirements on the receiving end of an NTP packet.

The UDP layer at the receiving end verifies the UDP Checksum of received NTP packets, and the NTP layer SHOULD ignore the Checksum Trailer extension field.

3.3. Interoperability with Existing Implementations

The behavior defined in this document does not impose new requirements on the reception of NTP packets. Thus, transmitters and intermediate nodes that support the Checksum Trailer can transparently interoperate with existing implementations.

3.4. Using the Checksum Trailer with or without Authentication

A Checksum Trailer SHOULD NOT be used when authentication is enabled.
The Checksum Trailer is effective in unauthenticated mode, allowing the intermediate entity to perform serial processing of the packet without storing-and-forwarding it.

On the other hand, when message authentication is used, an intermediate entity that alters NTP packets must also re-compute the Message Authentication Code (MAC) accordingly. The MAC update typically requires the intermediate entity to store the packet, re-compute its MAC, and then forward it. Thus, the benefit of the checksum trailer is effectively irrelevant when a MAC is used.

4. Security Considerations

This document describes how a Checksum Trailer extension can be used for maintaining the correctness of the UDP checksum.

The purpose of this extension is to ease the implementation of accurate timestamping engines, as described in Figure 1. The extension is intended to be used internally in an NTP client or server, and not intended to be used by intermediate switches and routers that reside between the client and the server. As opposed to PTP [IEEE1588], NTP does not require intermediate switches or routers to modify the content of NTP messages, and thus any such modification should be considered as a malicious MITM attack.

It is important to emphasize that the scheme described in this document does not increase the protocol's vulnerabitliy to MITM attacks; a MITM who maliciously modifies a packet and its checksum trailer is logically equivalent to a MITM attacker who modifies a packet and its UDP Checksum field.

The concept described in this document is intended to be used only in unauthenticated mode. As described in Section 3.4. , the benefits of the Checksum Trailer do not apply when authentication is enabled.

5. IANA Considerations

IANA is requested to allocate an NTP extension Field Type value for the Checksum Trailer extension.

6. Acknowledgments

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

7. References

7.1. Normative References

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

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

[Checksum] Rijsinghani, A., "Computation of the Internet Checksum via Incremental Update", RFC 1624, May 1994.

[UDP]   Postel, J., "User Datagram Protocol", RFC 768, August 1980.

[NTPv4]   Mills, D., Martin, J., Burbank, J., Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

7.2. Informative References

[IEEE1588]   IEEE TC 9 Instrumentation and Measurement Society 2000, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

[NTP-Ext]   Mizrahi, T., Mayer, D., "Using NTP Extension Fields without Authentication", draft-mizrahi-ntp-extension-field (work in progress), January 2014.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com