

NTP Working Group
Internet Draft
Intended status: Experimental
Expires: November 2013

Tal Mizrahi
Marvell
May 28, 2013

**Using UDP Checksum Trailers in the Network Time Protocol (NTP)
draft-mizrahi-ntp-checksum-trailer-00.txt**

Abstract

The Network Time Protocol (NTP) allows clients to synchronize to a time server using timestamped protocol messages. To facilitate 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.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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 November 28, 2013.

Copyright Notice

Copyright (c) 2013 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.](#) Intermediate Entities [3](#)
- [1.2.](#) Updating the UDP Checksum [5](#)
- [2.](#) Conventions used in this document [5](#)
- [2.1.](#) Terminology [5](#)
- [2.2.](#) Abbreviations [6](#)
- [3.](#) Using UDP Checksum Trailers in NTP [6](#)
- [3.1.](#) Overview [6](#)
- [3.2.](#) Checksum Trailer in NTP Packets [6](#)
- [3.2.1.](#) Transmission of NTP with Checksum Trailer..... [8](#)
- [3.2.2.](#) Intermediate Updates of NTP with Checksum Trailer .. [8](#)
- [3.2.3.](#) Reception of NTP with Checksum Trailer [8](#)
- [3.3.](#) Interoperability with Existing Implementations..... [8](#)
- 3.4. Using the Checksum Trailer with or without Authentication 8
- [4.](#) Security Considerations [9](#)
- [5.](#) IANA Considerations [9](#)
- [6.](#) Acknowledgments [10](#)
- [7.](#) References [10](#)
- [7.1.](#) Normative References [10](#)
- [7.2.](#) Informative References [10](#)

[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.

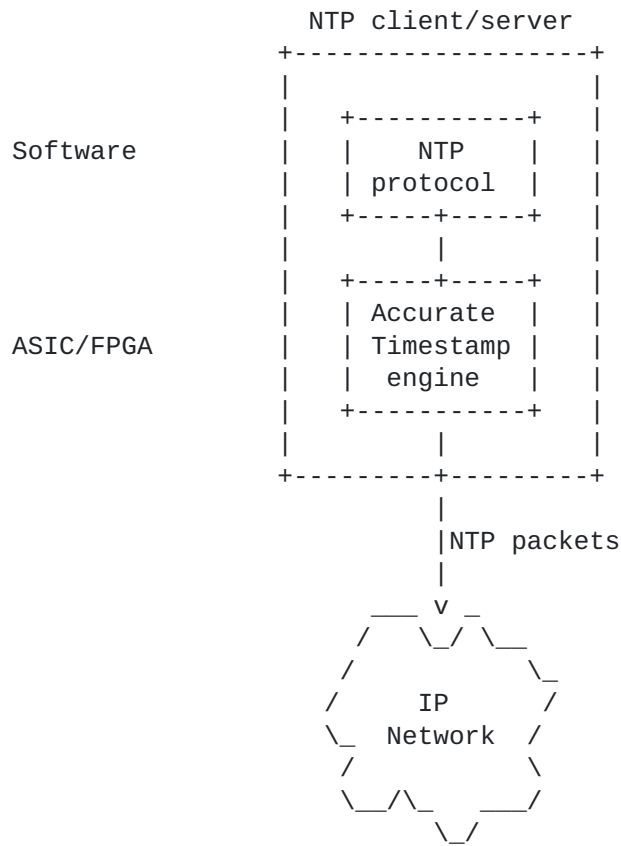


Figure 1 Accurate Timestamping in NTP

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 [\[KEYWORDS\]](#).

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

When message authentication is used, intermediate entities that alter test 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.

Figure 4 illustrates the format of an authenticated NTP packet with a 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] Rijssinghani, 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), February 2013.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com