Guidelines for Defining Packet Timestamps

Abstract

   Various network protocols make use of binary-encoded timestamps that
   are incorporated in the protocol packet format, referred to as
   "packet timestamps" for short. This document specifies guidelines
   for defining packet timestamp formats in networking protocols at
   various layers. It also presents three recommended timestamp
   formats. The target audience of this document includes network
   protocol designers. It is expected that a new network protocol that
   requires a packet timestamp will, in most cases, use one of the
   recommended timestamp formats. If none of the recommended formats
   fits the protocol requirements, the new protocol specification should
   specify the format of the packet timestamp according to the
   guidelines in this document.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are candidates for any level of Internet
   Standard; see Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc8877.

Copyright Notice

Table of Contents

1. Introduction

1.1. Background

   Timestamps are widely used in network protocols for various purposes:
   for logging or reporting the time of an event, for messages in delay
   measurement and clock synchronization protocols, and as part of a
   value that is unlikely to repeat (nonce) in security protocols.

   Timestamps are represented in the RFC series in one of two forms:
   text-based timestamps and packet timestamps. Text-based timestamps
   [RFC3339] are represented as user-friendly strings and are widely
   used in the RFC series -- for example, in information objects and
   data models, e.g., [RFC5646], [RFC6991], and [RFC7493]. Packet
   timestamps, on the other hand, are represented by a compact binary
   field that has a fixed size and are not intended to have a human-
   friendly format. Packet timestamps are also very common in the RFC
   series and are used, for example, for measuring delay and for
   synchronizing clocks, e.g., [RFC5905], [RFC4656], and [RFC7323].

1.2. Scope of this Document

   This document presents guidelines for defining a packet timestamp
   format in network protocols. Three recommended timestamp formats are

presented. It is expected that a new network protocol that requires
a packet timestamp will, in most cases, use one of these recommended
timestamp formats. In some cases, a network protocol may use more
than one of the recommended timestamp formats. However, if none of
the recommended formats fits the protocol requirements, the new
protocol specification should specify the format of the packet
timestamp according to the guidelines in this document.

The rationale behind defining a relatively small set of recommended
formats is that it enables significant reuse; network protocols can
typically reuse the timestamp format of the Network Time Protocol
(NTP) [RFC5905] or the Precision Time Protocol (PTP) [IEEE1588],
allowing a straightforward integration with an NTP- or PTP-based
timer. Moreover, since accurate timestamping mechanisms are often
implemented in hardware, a new network protocol that reuses an
existing timestamp format can be quickly deployed using existing
hardware timestamping capabilities.

1.3. How to Use This Document

This document is intended as a reference for network protocol
designers. When defining a network protocol that uses a packet
timestamp, the recommended timestamp formats should be considered
first (Section 4). If one of these formats is used, it should be
referenced along the lines of the examples in Sections 6.1 and 6.2.
If none of the recommended formats fits the required functionality,
then a new timestamp format should be defined using the template in
Section 3.

2. Terminology

2.1. 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
BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

2.2. Abbreviations

NTP   Network Time Protocol [RFC5905]

PTP   Precision Time Protocol [IEEE1588]

TAI   International Atomic Time

UTC   Coordinated Universal Time

2.3. Terms Used in This Document

Timestamp: A value that represents a point in time,
                        corresponding to an event that occurred or is
                        scheduled to occur.

Timestamp error: The difference between the timestamp value and

the value of a reference clock at the time of
                        the event that the timestamp was intended to
                        indicate.

   Timestamp format: The specification of a timestamp, which is
                           represented by a set of attributes that
                           unambiguously defines the syntax and semantics
                           of a timestamp.

   Timestamp accuracy: The mean over an ensemble of measurements of
                           the timestamp error.

   Timestamp precision: The variation over an ensemble of measurements
                           of the timestamp error.

   Timestamp resolution: The minimal time unit used for representing
                           the timestamp.

3. Packet Timestamp Specification Template

   This document recommends using the timestamp formats defined in
   Section 4. In cases where these timestamp formats do not satisfy the
   protocol requirements, the timestamp specification should clearly
   state the reasons for defining a new format. Moreover, it is
   recommended to derive the new timestamp format from an existing
   timestamp format, either a timestamp format from this document or any
   other previously defined timestamp format.

   The timestamp specification must unambiguously define the syntax and
   semantics of the timestamp. The current section defines the minimum
   set of attributes, but it should be noted that in some cases,
   additional attributes or aspects will need to be defined in the
   timestamp specification.

   This section defines a template for specifying packet timestamps. A
   timestamp format specification MUST include at least the following
   aspects:

   Timestamp syntax:
      Size: The number of bits (or octets) used to represent the packet
         timestamp field. If the timestamp is comprised of more than
         one field, the size of each field is specified. Network order
         (big endian) is assumed by default; if this is not the case,
         then this section explicitly specifies the endianity.

   Timestamp semantics:
      Units: The units used to represent the timestamp. If the
         timestamp is comprised of more than one field, the units of
         each field are specified. If a field is limited to a specific
         range of values, this section specifies the permitted range of
         values.

      Resolution: The timestamp resolution; the resolution is equal to
         the timestamp field unit. If the timestamp consists of two or
         more fields using different time units, then the resolution is
         the smallest time unit.

Wraparound: The wraparound period of the timestamp; any further
   wraparound-related considerations should be described here.

Epoch: The origin of the timescale used for the timestamp; the
   moment in time used as a reference for the timestamp value.
   For example, the epoch may be based on a standard time scale,
   such as UTC. Another example is a relative timestamp, in which
   the epoch could be the time at which the device using the
   timestamp was powered up and is not affected by leap seconds
   (see the next attribute).

Leap seconds: This subsection specifies whether the timestamp is
   affected by leap seconds. If the timestamp is affected by leap
   seconds, then it represents the time elapsed since the epoch
   minus the number of leap seconds that have occurred since the
   epoch.

Synchronization aspects:
   The specification of a network protocol that makes use of a packet
   timestamp is expected to include the synchronization aspects of
   using the timestamp. While the synchronization aspects are not
   strictly part of the timestamp format specification, these aspects
   provide the necessary context for using the timestamp within the
   scope of the protocol. In some cases, timestamps are used without
   synchronization, e.g., a timestamp that indicates the number of
   seconds since power-up. In such cases, the Synchronization
   Aspects section will specify that the timestamp does not
   correspond to a synchronized time reference and may discuss how
   this affects the usage of the timestamp. Further details about
   synchronization aspects are discussed in Section 5.

4. Recommended Timestamp Formats

   This document defines a set of recommended timestamp formats.
   Clearly, different network protocols may have different requirements
   and constraints; consequently, they may use different timestamp
   formats. The choice of a specific timestamp format for a given
   protocol may depend on various factors. A few examples of factors
   that may affect the choice of the timestamp format include the
   following:

   * Timestamp size: While some network protocols use a large timestamp
      field, in some cases, there may be constraints with respect to the
      timestamp size, affecting the choice of the timestamp format.

   * Resolution: The time resolution is another factor that may
      directly affect the selected timestamp format. A potentially
      important factor in this context is extensibility; it may be
      desirable to allow a timestamp format to be extensible to a higher
      resolution by extending the field. For example, the resolution of
      the NTP 32-bit timestamp format can be improved by extending it to
      the NTP 64-bit timestamp format in a straightforward way.

   * Wraparound period: The length of the time interval in which the
      timestamp is unique may also be an important factor in choosing

the timestamp format. Along with the timestamp resolution, these
two factors determine the required number of bits in the
timestamp.

* Common format for multiple protocols: If there are two or more
   network protocols that use timestamps and are often used together
   in typical systems, using a common timestamp format should be
   preferred if possible. For example, if the network protocol that
   is being defined typically runs on a PC, then an NTP-based
   timestamp format may allow easier integration with an NTP-
   synchronized timer. In contrast, a protocol that is typically
   deployed on a hardware-based platform may make better use of a
   PTP-based timestamp, allowing more efficient integration with a
   PTP-synchronized timer.

4.1. Using a Recommended Timestamp Format

   A specification that uses one of the recommended timestamp formats
   should specify explicitly that this is a recommended timestamp format
   and point to the relevant section in the current document.

4.2. NTP Timestamp Formats

4.2.1. NTP 64-Bit Timestamp Format

   The Network Time Protocol (NTP) 64-bit timestamp format is defined in
   [RFC5905]. This timestamp format is used in several network
   protocols, including [RFC6374], [RFC4656], and [RFC5357]. Since this
   timestamp format is used in NTP, it should be preferred in network
   protocols that are typically deployed in concert with NTP.

   The format is presented in this section according to the template
   defined in Section 3.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                            Seconds                            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                            Fraction                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
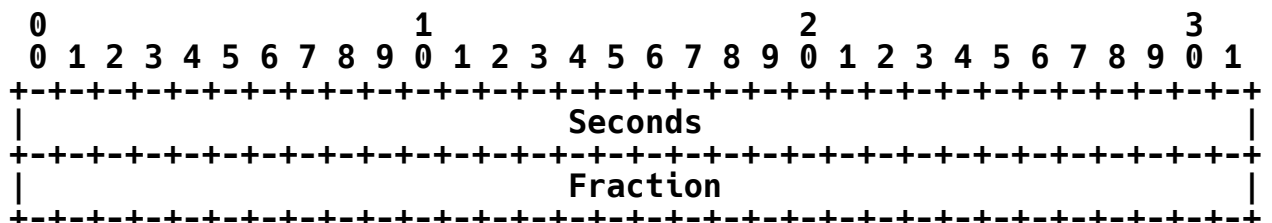```

                    Figure 1: NTP 64-Bit Timestamp Format

   Timestamp field format:
      Seconds:  Specifies the integer portion of the number of seconds
         since the epoch.

         Size:  32 bits.

         Units:  Seconds.

      Fraction:  Specifies the fractional portion of the number of
         seconds since the epoch.

         Size:  32 bits.

Units:  The unit is 2^(-32) seconds, which is roughly equal to
      233 picoseconds.

Epoch:
   The epoch is 1 January 1900 at 00:00 UTC.

   Note: As pointed out in [RFC5905], strictly speaking, UTC did not
   exist prior to 1 January 1972, but it is convenient to assume it
   has existed for all eternity.  The current epoch implies that the
   timestamp specifies the number of seconds since 1 January 1972 at
   00:00 UTC plus 2272060800 (which is the number of seconds between
   1 January 1900 and 1 January 1972).

Leap seconds:
   This timestamp format is affected by leap seconds.  The timestamp
   represents the number of seconds elapsed since the epoch minus the
   number of leap seconds.  Thus, during and possibly before and/or
   after the occurrence of a leap second, the value of the timestamp
   may temporarily be ambiguous, as further discussed in Section 5.

Resolution:
   The resolution is 2^(-32) seconds.

Wraparound:
   This time format wraps around every 2^(32) seconds, which is
   roughly 136 years.  The next wraparound will occur in the year
   2036.

4.2.2.  NTP 32-Bit Timestamp Format

The Network Time Protocol (NTP) 32-bit timestamp format is defined in
[RFC5905].  This timestamp format is used in [METRICS] and [NSHMD].
This timestamp format should be preferred in network protocols that
are typically deployed in concert with NTP.  The 32-bit format can be
used either when space constraints do not allow the use of the 64-bit
format or when the 32-bit format satisfies the resolution and
wraparound requirements.

The format is presented in this section according to the template
defined in Section 3.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Seconds             |           Fraction            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

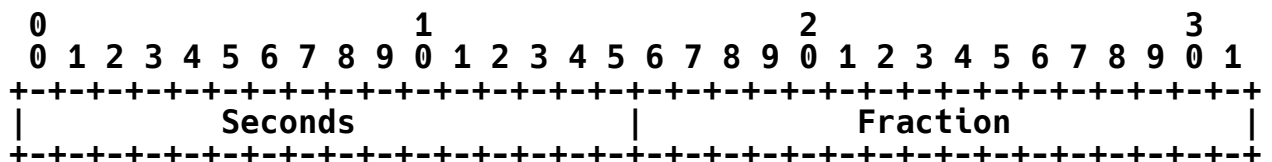   Figure 2: NTP 32-Bit Timestamp Format

Timestamp field format:
   Seconds: Specifies the integer portion of the number of seconds
      since the epoch.

      Size: 16 bits.

Units: Seconds.

Fraction: Specifies the fractional portion of the number of
   seconds since the epoch.

   Size: 16 bits.

   Units: The unit is 2^(-16) seconds, which is roughly equal to
      15.3 microseconds.

Epoch:
   The epoch is 1 January 1900 at 00:00 UTC.

   Note: As pointed out in [RFC5905], strictly speaking, UTC did not
   exist prior to 1 January 1972, but it is convenient to assume it
   has existed for all eternity. The current epoch implies that the
   timestamp specifies the number of seconds since 1 January 1972 at
   00:00 UTC plus 2272060800 (which is the number of seconds between
   1 January 1900 and 1 January 1972).

Leap seconds:
   This timestamp format is affected by leap seconds. The timestamp
   represents the number of seconds elapsed since the epoch minus the
   number of leap seconds. Thus, during and possibly before and/or
   after the occurrence of a leap second, the value of the timestamp
   may temporarily be ambiguous, as further discussed in Section 5.

Resolution:
   The resolution is 2^(-16) seconds.

Wraparound:
   This time format wraps around every 2^(16) seconds, which is
   roughly 18 hours.

4.3. The PTP Truncated Timestamp Format

The Precision Time Protocol (PTP) [IEEE1588] uses an 80-bit timestamp
format. The truncated timestamp format is a 64-bit field, which is
the 64 least significant bits of the 80-bit PTP timestamp. Since
this timestamp format is similar to the one used in PTP, this
timestamp format should be preferred in network protocols that are
typically deployed in PTP-capable devices.

The PTP truncated timestamp format was defined in [IEEE1588v1] and is
used in several protocols, such as [RFC6374], [RFC7456], [RFC8186],
and [ITU-T-Y.1731].

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Seconds                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Nanoseconds                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
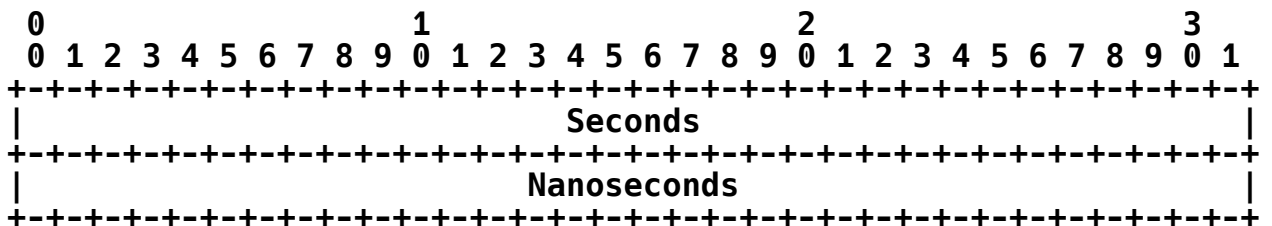
Figure 3: PTP Truncated Timestamp Format

   Timestamp field format:
      Seconds: Specifies the integer portion of the number of seconds
         since the epoch.

         Size: 32 bits.

         Units: Seconds.

      Nanoseconds: Specifies the fractional portion of the number of
         seconds since the epoch.

         Size: 32 bits.

         Units: Nanoseconds. The value of this field is in the range 0
            to (10^(9))-1.

   Epoch:
      The PTP [IEEE1588] epoch is 1 January 1970 00:00:00 TAI.

   Leap seconds:
      This timestamp format is not affected by leap seconds.

   Resolution:
      The resolution is 1 nanosecond.

   Wraparound:
      This time format wraps around every 2^(32) seconds, which is
      roughly 136 years. The next wraparound will occur in the year
      2106.

5. Synchronization Aspects

   A specification that defines a new timestamp format or uses one of
   the recommended timestamp formats should include a Synchronization
   Aspects section. Note that the recommended timestamp formats defined
   in this document (Section 4) do not include the synchronization
   aspects of these timestamp formats, but it is expected that
   specifications of network protocols that make use of these formats
   should include the synchronization aspects. Examples of a
   Synchronization Aspects section can be found in Section 6.

   The Synchronization Aspects section should specify all the
   assumptions and requirements related to synchronization. For
   example, the synchronization aspects may specify whether nodes
   populating the timestamps should be synchronized among themselves and
   whether the timestamp is measured with respect to a central reference
   clock such as an NTP server. If time is assumed to be synchronized
   to a time standard such as UTC or TAI, it should be specified in this
   section. Further considerations may be discussed in this section,
   such as the required timestamp accuracy and precision.

   Another aspect that should be discussed in this section is leap
   second [RFC5905] considerations. The timestamp specification
   template (Section 3) specifies whether the timestamp is affected by

leap seconds. It is often the case that further details about leap
seconds will need to be defined in the Synchronization Aspects
section. Generally speaking, a leap second is a one-second
adjustment that is occasionally applied to UTC in order to keep it
aligned with solar time. A leap second may be either positive or
negative, i.e., the clock may either be shifted one second forward or
backward. All leap seconds that have occurred up to the publication
of this document have been in the backward direction, and although
forward leap seconds are theoretically possible, the text throughout
this document focuses on the common case, which is the backward leap
second. In a timekeeping system that considers leap seconds, the
system clock may be affected by a leap second in one of three
possible ways:

*   The clock is turned backwards one second at the end of the leap
    second.

*   The clock is frozen during the duration of the leap second.

*   The clock is slowed down during the leap second and adjacent time
    intervals until the new time value catches up. The interval for
    this process, commonly referred to as "leap smear", can range from
    several seconds to several hours before, during, and/or after the
    occurrence of the leap second.

The way leap seconds are handled depends on the synchronization
protocol and is thus not specified in this document. However, if a
timestamp format is defined with respect to a timescale that is
affected by leap seconds, the Synchronization Aspects section should
specify how the use of leap seconds affects the timestamp usage.

6. Timestamp Use Cases

   Packet timestamps are used in various network protocols. Typical
   applications of packet timestamps include delay measurement, clock
   synchronization, and others. The following table presents a (non-
   exhaustive) list of protocols that use packet timestamps and the
   timestamp formats used in each of these protocols.

Protocol	Recommended Formats			Other
	NTP 64-Bit	NTP 32-Bit	PTP Trunc.	
NTP [RFC5905]	+			
OWAMP [RFC4656]	+			
TWAMP [RFC5357]	+			
TWAMP [RFC8186]	+		+	
TRILL [RFC7456]			+	
MPLS [RFC6374]			+	
TCP [RFC7323]				+

```
+------------------+----------+----------+----------+------+
|  RTP [RFC3550]   |    +     |          |          |  +   |
+------------------+----------+----------+----------+------+
|  IPFIX [RFC7011] |          |          |          |  +   |
+------------------+----------+----------+----------+------+
|    BinaryTime    |          |          |          |  +   |
|    [RFC6019]     |          |          |          |      |
+------------------+----------+----------+----------+------+
|     [METRICS]    |    +     |    +     |          |      |
+------------------+----------+----------+----------+------+
|      [NSHMD]     |          |    +     |    +     |      |
+------------------+----------+----------+----------+------+
```

           Table 1: Protocols That Use Packet Timestamps

   The rest of this section presents two hypothetical examples of
   network protocol specifications that use one of the recommended
   timestamp formats. The examples include the text that specifies the
   information related to the timestamp format.

6.1. Example 1

   Timestamp:
      The timestamp format used in this specification is the NTP
      [RFC5905] 64-bit format, as described in Section 4.2.1 of RFC
      8877.

   Synchronization aspects:
      It is assumed that the nodes that run this protocol are
      synchronized to UTC using a synchronization mechanism that is
      outside the scope of this document. In typical deployments, this
      protocol will run on a machine that uses NTP [RFC5905] for
      synchronization. Thus, the timestamp may be derived from the NTP-
      synchronized clock, allowing the timestamp to be measured with
      respect to the clock of an NTP server. Since the NTP time format
      is affected by leap seconds, the current timestamp format is
      similarly affected. Thus, the value of a timestamp during and
      possibly before and/or after a leap second may be temporarily
      inaccurate.

6.2. Example 2

   Timestamp:
      The timestamp format used in this specification is the PTP
      [IEEE1588] truncated format, as described in Section 4.3 of RFC
      8877.

   Synchronization aspects:
      It is assumed that the nodes that run this protocol are
      synchronized among themselves. Nodes may be synchronized to a
      global reference time. Note that if PTP [IEEE1588] is used for
      synchronization, the timestamp may be derived from the PTP-
      synchronized clock, allowing the timestamp to be measured with
      respect to a PTP grandmaster clock.

7. Packet Timestamp Control Field

In some cases, it is desirable to have a control field that describes the structure, format, content, and properties of timestamps. Control information about the timestamp format can be conveyed in some protocols using a dedicated control plane protocol or may be made available at the management plane, for example, using a YANG data model. An optional control field allows some of the control information to be attached to the timestamp.

An example of a packet timestamp control field is the Error Estimate field, defined by Section 4.1.2 of [RFC4656], which is used in the One-Way Active Measurement Protocol (OWAMP) [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) [RFC5357]. The Root Dispersion and Root Delay fields in the NTP header [RFC5905] are two examples of fields that provide information about the timestamp precision. Another example of an auxiliary field is the Correction Field in the PTP header [IEEE1588]; its value is used as a correction to the timestamp and may be assigned by the sender of the PTP message and updated by transit nodes (Transparent Clocks) in order to account for the delay along the path.

This section defines high-level guidelines for defining packet timestamp control fields in network protocols that can benefit from such timestamp-related control information. The word "requirements" is used in its informal context in this section.

7.1. High-Level Control Field Requirements

A control field for packet timestamps must offer an adequate feature set and fulfill a series of requirements to be usable and accepted. The following list captures the main high-level requirements for timestamp fields.

1. Extensible Feature Set: Protocols and applications depend on various timestamp characteristics. A timestamp control field must support a variable number of elements (components) that either describe or quantify timestamp-specific characteristics or parameters. Examples of potential elements include timestamp size, encoding, accuracy, leap seconds, reference clock identifiers, etc.

2. Size: Essential for an efficient use of timestamp control fields is the trade-off between supported features and control field size. Protocols and applications may select the specific control field elements that are needed for their operation from the set of available elements.

3. Composition: Applications may depend on specific control field elements being present in messages. The status of these elements may be either mandatory, conditional mandatory, or optional, depending on the specific application and context. A control field specification must support applications in conveying or negotiating (a) the set of control field elements along with (b) the status of any element (i.e., mandatory, conditional mandatory, or optional) by defining appropriate data structures and identity codes.

4. Category: Control field elements can characterize either static
    timestamp information (e.g., timestamp size in bytes and
    timestamp semantics: NTP 64-bit format) or runtime timestamp
    information (e.g., estimated timestamp accuracy at the time of
    sampling: 20 microseconds to UTC). For efficiency reasons, it
    may be meaningful to support separation of these two concepts:
    while the former (static) information is typically valid
    throughout a protocol session and may be conveyed only once, at
    session establishment time, the latter (runtime) information
    augments any timestamp instance and may cause substantial
    overhead for high-traffic protocols.

Proposals for timestamp control fields will be defined in separate
documents and are out of scope of this document.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

A network protocol that uses a packet timestamp MUST specify the
security considerations that result from using the timestamp. This
section provides an overview of some of the common security
considerations of using timestamps.

Any metadata that is attached to control or data packets, and
specifically packet timestamps, can facilitate network
reconnaissance; by passively eavesdropping on timestamped packets, an
attacker can gather information about the network performance and the
level of synchronization between nodes.

In some cases, timestamps could be spoofed or modified by on-path
attackers, thus attacking the application that uses the timestamps.
For example, if timestamps are used in a delay measurement protocol,
an attacker can modify en route timestamps in a way that manipulates
the measurement results. Integrity protection mechanisms, such as
Message Authentication Codes (MACs), can mitigate such attacks. The
specification of an integrity protection mechanism is outside the
scope of this document as, typically, integrity protection will be
defined on a per-network-protocol basis and not specifically for the
timestamp field.

Another potential threat that can have a similar impact is delay
attacks. An attacker can maliciously delay some or all of the en
route messages, with the same harmful implications as described in
the previous paragraph. Mitigating delay attacks is a significant
challenge; in contrast to spoofing and modification attacks, the
delay attack cannot be prevented by cryptographic integrity
protection mechanisms. In some cases, delay attacks can be mitigated
by sending the timestamped information through multiple paths,
allowing detection of and resistance to an attacker that has access
to one of the paths.

In many cases, timestamping relies on an underlying synchronization

mechanism. Thus, any attack that compromises the synchronization
mechanism can also compromise protocols that use timestamping.
Attacks on time protocols are discussed in detail in [RFC7384].

10. References

10.1. 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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
           2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
           May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[IEEE1588] IEEE, "IEEE Standard for a Precision Clock Synchronization
           Protocol for Networked Measurement and Control Systems",
           DOI 10.1109/IEEESTD.2008.4579760, IEEE Std. 1588-2008,
           July 2008, <https://doi.org/10.1109/IEEESTD.2008.4579760>.

[IEEE1588v1]
           IEEE, "IEEE Standard for a Precision Clock Synchronization
           Protocol for Networked Measurement and Control Systems",
           DOI 10.1109/IEEESTD.2002.94144, IEEE Std. 1588-2002,
           October 2002,
           <https://doi.org/10.1109/IEEESTD.2002.94144>.

[ITU-T-Y.1731]
           ITU-T, "Operations, administration and maintenance (OAM)
           functions and mechanisms for Ethernet-based networks",
           ITU-T Recommendation G.8013/Y.1731, August 2015.

[METRICS] Morton, A., Bagnulo, M., Eardley, P., and K. D'Souza,
           "Initial Performance Metrics Registry Entries", Work in
           Progress, Internet-Draft, draft-ietf-ippm-initial-
           registry-16, 9 March 2020, <https://tools.ietf.org/html/
           draft-ietf-ippm-initial-registry-16>.

[NSHMD]   Guichard, J., Smith, M., Kumar, S., Majee, S., and T.
           Mizrahi, "Network Service Header (NSH) MD Type 1: Context
           Header Allocation (Data Center)", Work in Progress,
           Internet-Draft, draft-ietf-sfc-nsh-dc-allocation-02, 25
           September 2018, <https://tools.ietf.org/html/draft-ietf-
           sfc-nsh-dc-allocation-02>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
           Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
           <https://www.rfc-editor.org/info/rfc3339>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
           Jacobson, "RTP: A Transport Protocol for Real-Time
           Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,

                 July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
           Zekauskas, "A One-way Active Measurement Protocol
           (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006,
           <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.
           Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
           RFC 5357, DOI 10.17487/RFC5357, October 2008,
           <https://www.rfc-editor.org/info/rfc5357>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying
           Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646,
           September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[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>.

[RFC6019] Housley, R., "BinaryTime: An Alternate Format for
           Representing Date and Time in ASN.1", RFC 6019,
           DOI 10.17487/RFC6019, September 2010,
           <https://www.rfc-editor.org/info/rfc6019>.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay
           Measurement for MPLS Networks", RFC 6374,
           DOI 10.17487/RFC6374, September 2011,
           <https://www.rfc-editor.org/info/rfc6374>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types",
           RFC 6991, DOI 10.17487/RFC6991, July 2013,
           <https://www.rfc-editor.org/info/rfc6991>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
           "Specification of the IP Flow Information Export (IPFIX)
           Protocol for the Exchange of Flow Information", STD 77,
           RFC 7011, DOI 10.17487/RFC7011, September 2013,
           <https://www.rfc-editor.org/info/rfc7011>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R.
           Scheffenegger, Ed., "TCP Extensions for High Performance",
           RFC 7323, DOI 10.17487/RFC7323, September 2014,
           <https://www.rfc-editor.org/info/rfc7323>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in
           Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384,
           October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC7456] Mizrahi, T., Senevirathne, T., Salam, S., Kumar, D., and
           D. Eastlake 3rd, "Loss and Delay Measurement in
           Transparent Interconnection of Lots of Links (TRILL)",
           RFC 7456, DOI 10.17487/RFC7456, March 2015,
           <https://www.rfc-editor.org/info/rfc7456>.

[RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493,
           DOI 10.17487/RFC7493, March 2015,
           <https://www.rfc-editor.org/info/rfc7493>.

[RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588
           Timestamp Format in a Two-Way Active Measurement Protocol
           (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017,
           <https://www.rfc-editor.org/info/rfc8186>.

Acknowledgments

Authors' Addresses

Tal Mizrahi
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com

Joachim Fabini
TU Wien
Gusshausstrasse 25/E389
1040 Vienna
Austria

Phone: +43 1 58801 38813
Email: Joachim.Fabini@tuwien.ac.at
URI: http://www.tc.tuwien.ac.at/about-us/staff/joachim-fabini/

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown, NJ 07748
United States of America

Phone: +1 732 420 1571
Email: acmorton@att.com