ï»¿

Simple Two-Way Active Measurement Protocol

Abstract

   This document describes the Simple Two-way Active Measurement
   Protocol (STAMP), which enables the measurement of both one-way and
   round-trip performance metrics, like delay, delay variation, and
   packet loss.

Status of This Memo

   This is an Internet Standards Track document.

   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). Further information on
   Internet Standards is available in 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/rfc8762.

Copyright Notice

Table of Contents

1. Introduction

   Development and deployment of the Two-Way Active Measurement Protocol
   (TWAMP) [RFC5357] and its extensions (e.g., [RFC6038], which defines
   Symmetrical Size for TWAMP) provided invaluable experience. Several
   independent implementations of both TWAMP and TWAMP Light exist, have
   been deployed, and provide important operational performance
   measurements.

   At the same time, there has been noticeable interest in using a more
   straightforward mechanism for active performance monitoring that can
   provide deterministic behavior and inherent separation of control
   (vendor-specific configuration or orchestration) and test functions.
   Recent work on "Performance Measurement from IP Edge to Customer
   Equipment using TWAMP Light" [BBF.TR-390] by the Broadband Forum
   demonstrates that interoperability among implementations of TWAMP
   Light is difficult because the composition and operation of TWAMP
   Light were not sufficiently specified in [RFC5357]. According to
   [RFC8545], TWAMP Light includes a subset of TWAMP-Test functions.
   Thus, to have a comprehensive tool to measure packet loss and delay
   requires support by other applications that provide, for example,
   control and security.

   This document defines an active performance measurement test
   protocol, Simple Two-way Active Measurement Protocol (STAMP), that
   enables measurement of both one-way and round-trip performance
   metrics, like delay, delay variation, and packet loss. Support of
   some optional TWAMP extensions, e.g., [RFC7750], is discussed in
   [STAMP-OPTION].

2. Conventions Used in This Document

2.1. Terminology

   STAMP: Simple Two-way Active Measurement Protocol

   NTP: Network Time Protocol

   PTP: Precision Time Protocol

   HMAC: Hashed Message Authentication Code

   OWAMP: One-Way Active Measurement Protocol

   TWAMP: Two-Way Active Measurement Protocol

   MBZ: Must be Zero

   PDU: Protocol Data Unit

2.2. 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.

3. Operation and Management of Performance Measurement Based on STAMP

   Figure 1 presents the Simple Two-way Active Measurement Protocol
   (STAMP) Session-Sender and Session-Reflector with a measurement

session. In this document, a measurement session, also referred to
as a "STAMP session", is the bidirectional packet flow between one
specific Session-Sender and one particular Session-Reflector for a
time duration. The configuration and management of the STAMP
Session-Sender, Session-Reflector, and sessions are outside the scope
of this document and can be achieved through various means. A few
examples are Command Line Interface, telecommunication services'
Operational Support System (OSS) / Business Support System (BSS),
SNMP, and NETCONF/YANG-based Software-Defined Networking (SDN)
controllers.

```
        o-----------------------------------------------------------o
        |                      Configuration and                    |
        |                        Management                         |
        o-----------------------------------------------------------o
               ||                                      ||
               ||                                      ||
               ||                                      ||
   +----------------------+                +------------------------+
   | STAMP Session-Sender | <--- STAMP---> | STAMP Session-Reflector |
   +----------------------+                +------------------------+
```
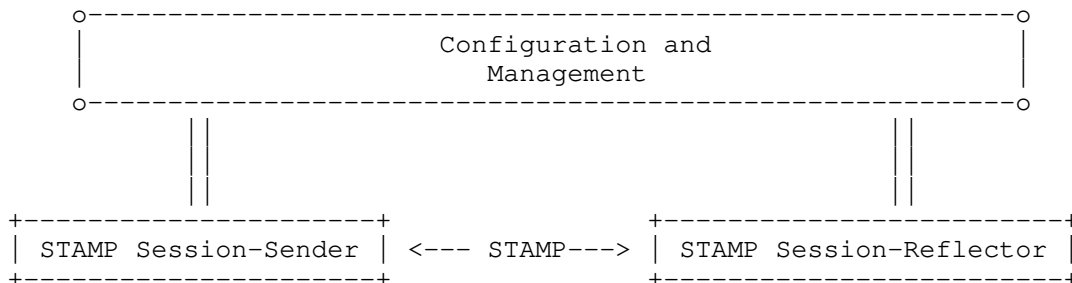
                    Figure 1: STAMP Reference Model

4. Theory of Operation

   The STAMP Session-Sender transmits test packets over UDP transport
   toward the STAMP Session-Reflector. The STAMP Session-Reflector
   receives the Session-Sender's packet and acts according to the
   configuration. Two modes of the STAMP Session-Reflector characterize
   the expected behavior and, consequently, performance metrics that can
   be measured:

   Stateless:
      The STAMP Session-Reflector does not maintain test state and will
      use the value in the Sequence Number field in the received packet
      as the value for the Sequence Number field in the reflected
      packet. As a result, values in the Sequence Number and Session-
      Sender Sequence Number fields are the same, and only round-trip
      packet loss can be calculated while the reflector is operating in
      stateless mode.

   Stateful:
      STAMP Session-Reflector maintains the test state, thus allowing
      the Session-Sender to determine directionality of loss using the
      combination of gaps recognized in the Session Sender Sequence
      Number and Sequence Number fields, respectively. As a result,
      both near-end (forward) and far-end (backward) packet loss can be
      computed. That implies that the STAMP Session-Reflector MUST
      maintain a state for each configured STAMP-Test session, thereby
      uniquely associating STAMP-Test packets with one such session
      instance and, thus, enabling the addition of a sequence number in
      the test reply that is individually incremented by one on a per-
      session basis.

   STAMP supports two authentication modes: unauthenticated and
   authenticated. Unauthenticated STAMP-Test packets, defined in
   Sections 4.2.1 and 4.3.1, ensure interworking between STAMP and TWAMP
   Light, as described in Section 4.6 regarding packet formats.

   By default, STAMP uses symmetrical packets, i.e., the size of the
   packet transmitted by the Session-Reflector equals the size of the
   packet received by the Session-Reflector.

4.1. UDP Port Numbers in STAMP Testing

   A STAMP Session-Sender MUST use UDP port 862 (TWAMP-Test Receiver
   Port) as the default destination UDP port number. A STAMP
   implementation of the Session-Sender MUST be able to be used as the
   destination UDP port numbers from the User Ports (aka Registered
   Ports) and Dynamic Ports (aka Private or Ephemeral Ports) ranges

defined in [RFC6335]. Before using numbers from the User Ports
range, the possible impact on the network MUST be carefully studied
and agreed on by all users of the network domain where the test has
been planned.

By default, an implementation of the STAMP Session-Reflector MUST
receive STAMP-Test packets on UDP port 862. An implementation of the
Session-Reflector that supports this specification MUST be able to
define the port number to receive STAMP-Test packets from User Ports
and Dynamic Ports ranges, which are defined in [RFC6335]. STAMP
defines two different test packet formats: one for packets
transmitted by the STAMP Session-Sender and one for packets
transmitted by the STAMP Session-Reflector.

4.2. Session-Sender Behavior and Packet Format

A STAMP Session-Reflector supports the symmetrical size of test
packets, as defined in Section 3 of [RFC6038], as the default
behavior. A reflected base test packet includes information from the
Session-Reflector and, thus, is larger. To maintain the symmetry
between base STAMP packets, the base STAMP Session-Sender packet
includes the Must-Be-Zero (MBZ) field to match to the size of a base
reflected STAMP test packet. Hence, the base STAMP Session-Sender
packet has a minimum size of 44 octets in unauthenticated mode (see
Figure 2) and 112 octets in the authenticated mode (see Figure 4).
Generating variable length of a test packet in STAMP is defined in
[STAMP-OPTION].

4.2.1. Session-Sender Packet Format in Unauthenticated Mode

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Timestamp                            |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Error Estimate        |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
|                                                               |
|                      MBZ   (30 octets)                        |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                Figure 2: STAMP Session-Sender Test Packet Format in
                            Unauthenticated Mode

The fields are defined as following:

* The Sequence Number field is four octets long. For each new
   session, its value starts at zero and is incremented by one with
   each transmitted packet.

* The Timestamp field is eight octets long. The STAMP node MUST
   support the Network Time Protocol (NTP) version 4 64-bit timestamp
   format [RFC5905], the format used in [RFC5357]. The STAMP node
   MAY support the IEEE 1588v2 Precision Time Protocol (PTP)
   truncated 64-bit timestamp format [IEEE.1588.2008], the format
   used in [RFC8186]. The use of the specific format, NTP or PTP, is
   part of configuration of the Session-Sender or the particular test
   session.

* The Error Estimate field is two octets long with the format
   displayed in Figure 3:

```
              0                   1
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             |S|Z|   Scale   |  Multiplier   |
             +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
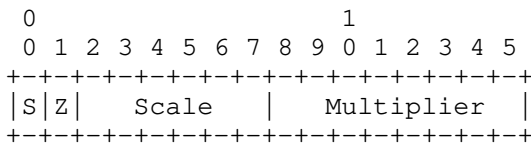
                  Figure 3: Error Estimate Format

   The S, Scale, and Multiplier fields are interpreted as they are
   defined in Section 4.1.2 of [RFC4656]. The Z field is interpreted
   as it is defined in Section 2.3 of [RFC8186]:

   0: NTP 64-bit format of a timestamp

   1: PTPv2 truncated format of a timestamp

   The default behavior of the STAMP Session-Sender and Session-
   Reflector is to use the NTP 64-bit timestamp format (Z field value
   of 0). An operator using configuration/management function MAY
   configure the STAMP Session-Sender and Session-Reflector to use
   the PTPv2 truncated format of a timestamp (Z field value of 1).
   Note that an implementation of a Session-Sender that supports this
   specification MAY be configured to use the PTPv2 format of a
   timestamp even though the Session-Reflector is configured to use
   NTP format.

   * The MBZ field in the Session-Sender unauthenticated packet is 30
      octets long. It MUST be all zeroed on the transmission and MUST
      be ignored on receipt.

4.2.2. Session-Sender Packet Format in Authenticated Mode

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
   |                       MBZ (12 octets)                         |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                         Timestamp                             |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Error Estimate         |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
   ~                                                               ~
   |                       MBZ (70 octets)                         |
   ~                                                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
   |                       HMAC (16 octets)                        |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
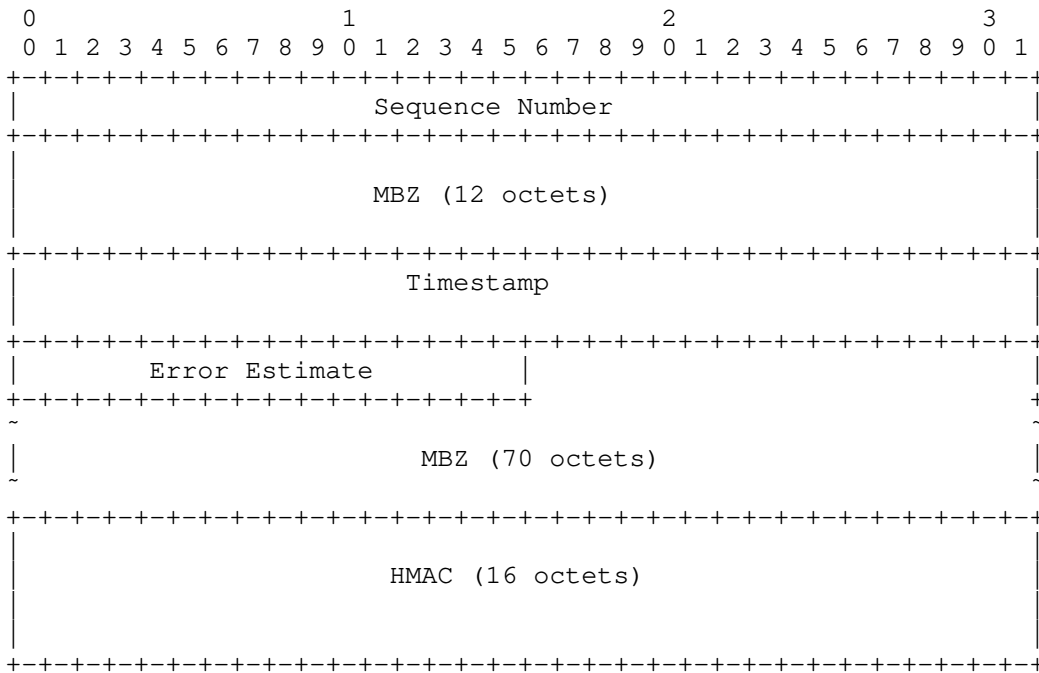
           Figure 4: STAMP Session-Sender Test Packet Format in
                            Authenticated Mode

   The field definitions are the same as the unauthenticated mode,
   listed in Section 4.2.1. Also, MBZ fields are used to make the
   packet length a multiple of 16 octets. The value of the field MUST
   be zeroed on transmission and MUST be ignored on receipt. Note, that
   both MBZ fields are used to calculate a key hashed message
   authentication code (HMAC) [RFC2104] hash. Also, the packet includes
   an HMAC hash at the end of the PDU. The detailed use of the HMAC
   field is described in Section 4.4.

4.3. Session-Reflector Behavior and Packet Format

The Session-Reflector receives the STAMP-Test packet and verifies it.
If the base STAMP-Test packet is validated, the Session-Reflector
that supports this specification prepares and transmits the reflected
test packet symmetric to the packet received from the Session-Sender
copying the content beyond the size of the base STAMP packet (see
Section 4.2).

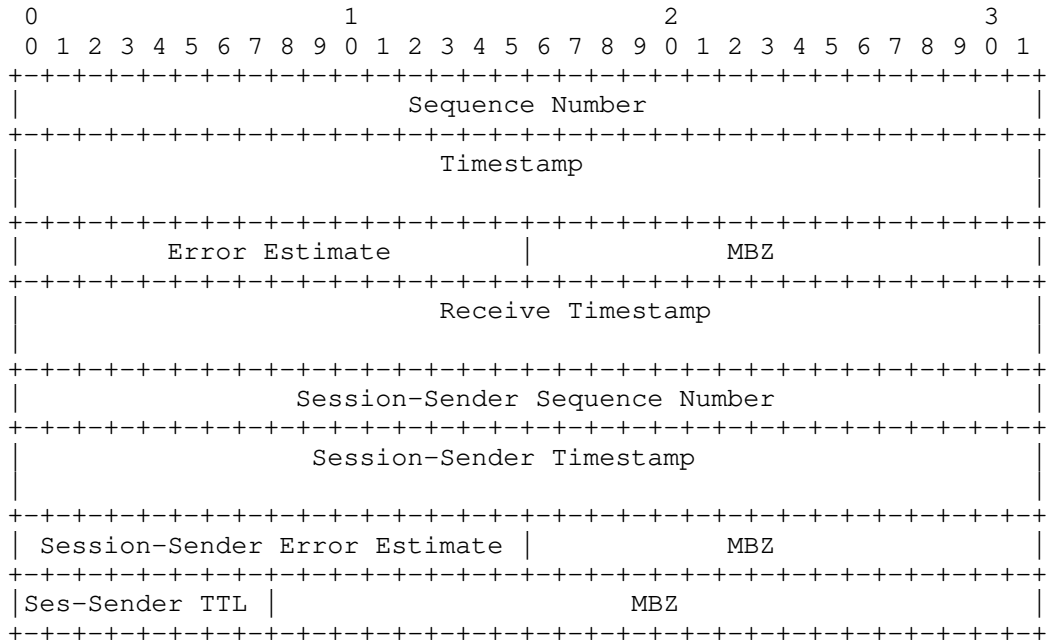4.3.1. Session-Reflector Packet Format in Unauthenticated Mode

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Timestamp                            |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Error Estimate        |              MBZ              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Receive Timestamp                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Session-Sender Sequence Number               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Session-Sender Timestamp                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Session-Sender Error Estimate |              MBZ              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ses-Sender TTL |                    MBZ                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: STAMP Session-Reflector Test Packet Format in
Unauthenticated Mode

Fields are defined as the following:

* The Sequence Number field is four octets long. The value of the
   Sequence Number field is set according to the mode of the STAMP
   Session-Reflector:

   - In the stateless mode, the Session-Reflector copies the value
      from the received STAMP-Test packet's Sequence Number field.

   - In the stateful mode, the Session-Reflector counts the
      transmitted STAMP-Test packets. It starts with zero and is
      incremented by one for each subsequent packet for each test
      session. The Session-Reflector uses that counter to set the
      value of the Sequence Number field.

* The Timestamp and Receive Timestamp fields are each eight octets
   long. The format of these fields, NTP or PTPv2, is indicated by
   the Z field of the Error Estimate field, as described in
   Section 4.2.1. Receive Timestamp is the time the test packet was
   received by the Session-Reflector. Timestamp is the time taken by
   the Session-Reflector at the start of transmitting the test
   packet.

* The Error Estimate field has the same size and interpretation as
   described in Section 4.2.1. It is applicable to both Timestamp
   and Receive Timestamp.

* The Session-Sender Sequence Number, Session-Sender Timestamp, and
   Session-Sender Error Estimate fields are copies of the
   corresponding fields in the STAMP-Test packet sent by the Session-
   Sender.

* The Session-Sender TTL field is one octet long, and its value is
   the copy of the TTL field in IPv4 (or Hop Limit in IPv6) from the

received STAMP-Test packet.

* The MBZ fields are used to achieve alignment of fields within the packet on a four-octet boundary. The value of each MBZ field MUST be zeroed on transmission and MUST be ignored on receipt.

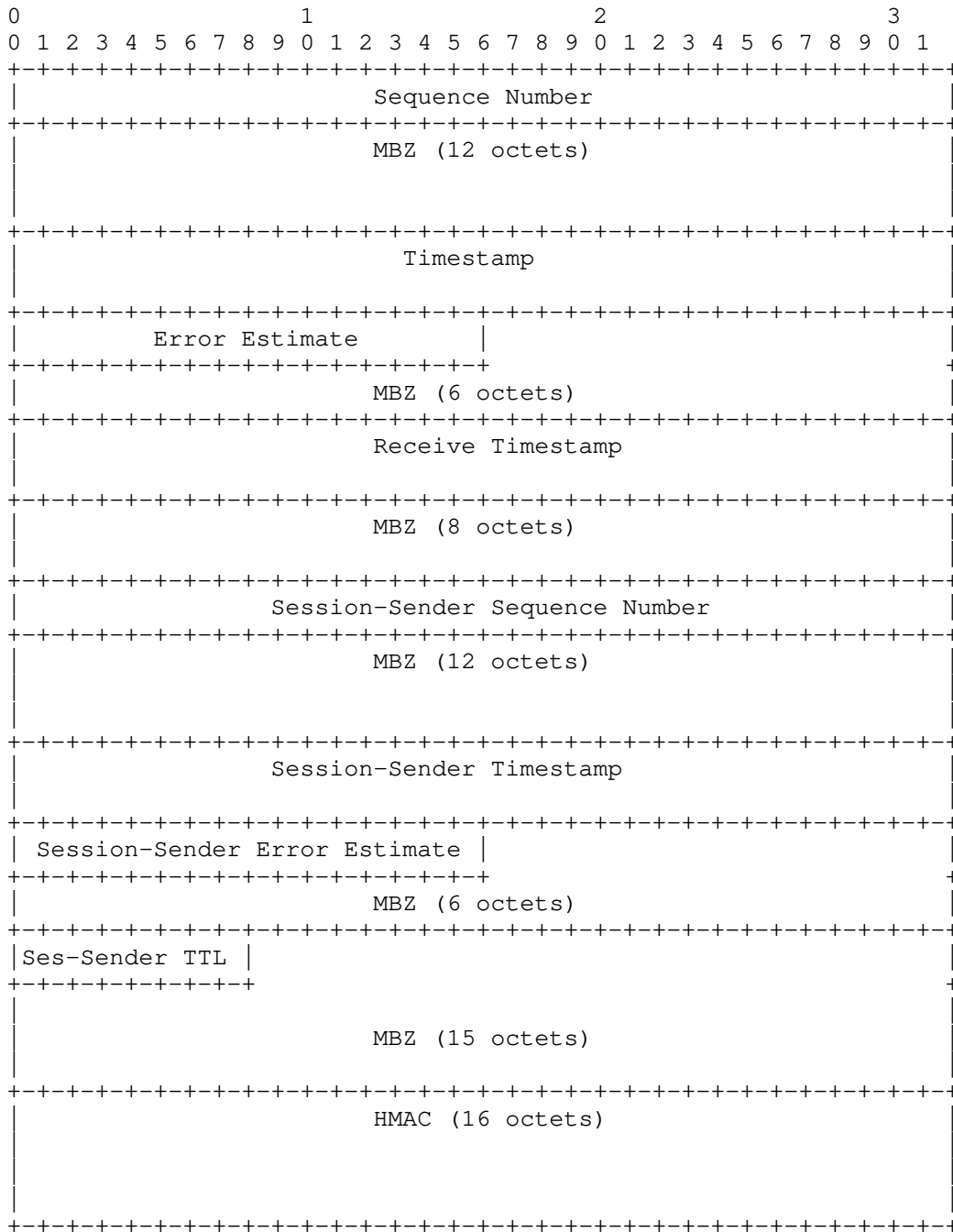4.3.2. Session-Reflector Packet Format in Authenticated Mode

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MBZ (12 octets)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Timestamp                            |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Error Estimate         |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                        MBZ (6 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Receive Timestamp                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MBZ (8 octets)                         |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Session-Sender Sequence Number                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        MBZ (12 octets)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Session-Sender Timestamp                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Session-Sender Error Estimate |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                        MBZ (6 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ses-Sender TTL |                                               |
+-+-+-+-+-+-+-+-+                                               +
|                                                               |
|                        MBZ (15 octets)                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        HMAC (16 octets)                       |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: STAMP Session-Reflector Test Packet Format in
Authenticated Mode

The field definitions are the same as the unauthenticated mode, listed in Section 4.3.1. Additionally, the MBZ field is used to make the packet length a multiple of 16 octets. The value of the field MUST be zeroed on transmission and MUST be ignored on receipt. Note that the MBZ field is used to calculate the HMAC hash value. Also, the STAMP Session-Reflector test packet format in authenticated mode includes the HMAC [RFC2104] hash at the end of the PDU. The detailed use of the HMAC field is in Section 4.4.

4.4. Integrity Protection in STAMP

Authenticated mode provides integrity protection to each STAMP

message by adding Hashed Message Authentication Code (HMAC). STAMP
uses HMAC-SHA-256 truncated to 128 bits (similarly to the use of it
in IPsec defined in [RFC4868]); hence, the length of the HMAC field
is 16 octets. In the authenticated mode, HMAC covers the first six
blocks (96 octets). HMAC uses its own key, which may be unique for
each STAMP-Test session; key management and the mechanisms to
distribute the HMAC key are outside the scope of this specification.
One example is to use an orchestrator to configure the HMAC key based
on the STAMP YANG data model [STAMP-YANG]. HMAC MUST be verified as
early as possible to avoid using or propagating corrupted data.

Future specifications may define the use of other, more advanced
cryptographic algorithms, possibly providing an update to the STAMP
YANG data model [STAMP-YANG].

4.5. Confidentiality Protection in STAMP

If confidentiality protection for STAMP is required, a STAMP-Test
session MUST use a secured transport. For example, STAMP packets
could be transmitted in the dedicated IPsec tunnel or share the IPsec
tunnel with the monitored flow. Also, the Datagram Transport Layer
Security protocol would provide the desired confidentiality
protection.

4.6. Interoperability with TWAMP Light

One of the essential requirements to STAMP is the ability to
interwork with a TWAMP Light device. Because STAMP and TWAMP use
different algorithms in authenticated mode (HMAC-SHA-256 versus HMAC-
SHA-1), interoperability is only considered for unauthenticated mode.
There are two possible combinations for such a use case:

* STAMP Session-Sender with TWAMP Light Session-Reflector

* TWAMP Light Session-Sender with STAMP Session-Reflector

In the former case, the Session-Sender might not be aware that its
Session-Reflector does not support STAMP. For example, a TWAMP Light
Session-Reflector may not support the use of UDP port 862, as
specified in [RFC8545]. Thus, Section 4 permits a STAMP Session-
Sender to use alternative ports. If any of STAMP extensions are
used, the TWAMP Light Session-Reflector will view them as the Packet
Padding field.

In the latter scenario, if a TWAMP Light Session-Sender does not
support the use of UDP port 862, the test management system MUST set
the STAMP Session-Reflector to use UDP port number, as permitted by
Section 4. The Session-Reflector MUST be set to use the default
format for its timestamps, NTP.

A STAMP Session-Reflector that supports this specification will
transmit the base packet (Figure 5) if it receives a packet smaller
than the STAMP base packet. If the packet received from the TWAMP
Session-Sender is larger than the STAMP base packet, the STAMP
Session-Reflector that supports this specification will copy the
content of the remainder of the received packet to transmit a
reflected packet of symmetrical size.

5. Operational Considerations

STAMP is intended to be used on production networks to enable the
operator to assess service level agreements based on packet delay,
delay variation, and loss. When using STAMP over the Internet,
especially when STAMP-Test packets are transmitted with the
destination UDP port number from the User Ports range, the possible
impact of the STAMP-Test packets MUST be thoroughly analyzed. The
use of STAMP for each case MUST be agreed by users of nodes hosting
the Session-Sender and Session-Reflector before starting the STAMP-
Test session.

Also, the use of the well-known port number as the destination UDP

port number in STAMP-Test packets transmitted by a Session-Sender
would not impede the ability to measure performance in an Equal-Cost
Multipath environment, and analysis in Section 5.3 of [RFC8545] fully
applies to STAMP.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

[RFC5357] does not identify security considerations specific to
TWAMP-Test but refers to security considerations identified for OWAMP
in [RFC4656]. Since both OWAMP and TWAMP include control-plane and
data-plane components, only security considerations related to OWAMP-
Test discussed in Sections 6.2 and 6.3 of [RFC4656] apply to STAMP.

STAMP uses the well-known UDP port number allocated for the OWAMP-
Test/TWAMP-Test Receiver Port. Thus, the security considerations and
measures to mitigate the risk of the attack using the registered port
number documented in Section 6 of [RFC8545] equally apply to STAMP.
Because of the control and management of a STAMP-Test being outside
the scope of this specification, only the more general requirement is
set:

   To mitigate the possible attack vector, the control and management
   of a STAMP-Test session MUST use the secured transport.

   The load of the STAMP-Test packets offered to a network MUST be
   carefully estimated, and the possible impact on the existing
   services MUST be thoroughly analyzed before launching the test
   session. Section 3.1.5 of [RFC8085] provides guidance on handling
   network load for UDP-based protocol. While the characteristic of
   test traffic depends on the test objective, it is highly
   recommended to stay in the limits, as provided in [RFC8085].

Use of HMAC-SHA-256 in the authenticated mode protects the data
integrity of the STAMP-Test packets.

8. References

8.1. Normative References

[IEEE.1588.2008]
          IEEE, "IEEE Standard for a Precision Clock Synchronization
          Protocol for Networked Measurement and Control Systems",
          IEEE Standard 1588, July 2008.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-
          Hashing for Message Authentication", RFC 2104,
          DOI 10.17487/RFC2104, February 1997,
          <https://www.rfc-editor.org/info/rfc2104>.

[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>.

[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>.

[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>.

   [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement
              Protocol (TWAMP) Reflect Octets and Symmetrical Size
              Features", RFC 6038, DOI 10.17487/RFC6038, October 2010,
              <https://www.rfc-editor.org/info/rfc6038>.

   [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S.
              Cheshire, "Internet Assigned Numbers Authority (IANA)
              Procedures for the Management of the Service Name and
              Transport Protocol Port Number Registry", BCP 165,
              RFC 6335, DOI 10.17487/RFC6335, August 2011,
              <https://www.rfc-editor.org/info/rfc6335>.

   [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>.

   [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>.

   [RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port
              Assignments for the One-Way Active Measurement Protocol
              (OWAMP) and the Two-Way Active Measurement Protocol
              (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019,
              <https://www.rfc-editor.org/info/rfc8545>.

8.2. Informative References

   [BBF.TR-390]
              Broadband Forum, "Performance Measurement from IP Edge to
              Customer Equipment using TWAMP Light", TR-390 Issue 1, May
              2017.

   [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-
              384, and HMAC-SHA-512 with IPsec", RFC 4868,
              DOI 10.17487/RFC4868, May 2007,
              <https://www.rfc-editor.org/info/rfc4868>.

   [RFC7750] Hedin, J., Mirsky, G., and S. Baillargeon, "Differentiated
              Service Code Point and Explicit Congestion Notification
              Monitoring in the Two-Way Active Measurement Protocol
              (TWAMP)", RFC 7750, DOI 10.17487/RFC7750, February 2016,
              <https://www.rfc-editor.org/info/rfc7750>.

   [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085,
              March 2017, <https://www.rfc-editor.org/info/rfc8085>.

   [STAMP-OPTION]
              Mirsky, G., Xiao, M., Nydell, H., Foote, R., Masputra, A.,
              and E. Ruffini, "Simple Two-way Active Measurement
              Protocol Optional Extensions", Work in Progress, Internet-
              Draft, draft-ietf-ippm-stamp-option-tlv-03, 21 February
              2020, <https://tools.ietf.org/html/draft-ietf-ippm-stamp-
              option-tlv-03>.

   [STAMP-YANG]
              Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active
              Measurement Protocol (STAMP) Data Model", Work in
              Progress, Internet-Draft, draft-ietf-ippm-stamp-yang-05,
              25 October 2019, <https://tools.ietf.org/html/draft-ietf-
              ippm-stamp-yang-05>.

Acknowledgments

suggestions. Also, our sincere thanks to David Ball, Rakesh Gandhi, and Xiao Min for their thorough reviews and helpful comments.

Authors' Addresses

    Greg Mirsky
    ZTE Corp.

    Email: gregimirsky@gmail.com

    Guo Jun
    ZTE Corp.
    68# Zijinghua Road
    Nanjing
    Jiangsu, 210012
    China

    Phone: +86 18105183663
    Email: guo.jun2@zte.com.cn

    Henrik Nydell
    Accedian Networks

    Email: hnydell@accedian.com

    Richard Foote
    Nokia

    Email: footer.foote@nokia.com