Network Working Group   G. Mirsky
Internet-Draft   ZTE Corp.
Intended status: Standards Track   G. Jun
Expires: February 13, 2020   ZTE Corporation
                                                             H. Nydell
                                                      Accedian Networks
                                                              R. Foote
                                                                 Nokia
                                                       August 12, 2019

                  Simple Two-way Active Measurement Protocol
                        draft-ietf-ippm-stamp-07

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Development and deployment of Two-Way Active Measurement Protocol
   (TWAMP) [RFC5357] and its extensions, e.g., [RFC6038] that defined
   features such as Reflect Octets and Symmetrical Size for TWAMP
   provided invaluable experience. Several independent implementations
   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 inherit separation of control (vendor-specific configuration or
   orchestration) and test functions. One of such is Performance
   Measurement from IP Edge to Customer Equipment using TWAMP Light from
   Broadband Forum [BBF.TR-390] used as the reference TWAMP Light that,

according to [RFC8545], includes sub-set of TWAMP-Test functions in
combination with 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.
Some TWAMP extensions, e.g., [RFC7750] are supported by the
extensions to STAMP base specification in
[I-D.ietf-ippm-stamp-option-tlv].

2. Conventions used in this document

2.1. Terminology

AES Advanced Encryption Standard

CBC Cipher Block Chaining

ECB Electronic Cookbook

KEK Key-encryption Key

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 May be Zero

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. Softwarization of Performance Measurement

   Figure 1 presents the Simple Two-way Active Measurement Protocol
   (STAMP) Session-Sender, and Session-Reflector with a measurement
   session. The configuration and management of the STAMP Session-
   Sender, Session-Reflector, and management of the STAMP sessions can
   be achieved through various means. Command Line Interface, OSS/BSS
   (operations support system/business support system as a combination
   of two systems used to support a range of telecommunication services)

   using SNMP or controllers in Software-Defined Networking using
   Netconf/YANG are but a few examples.

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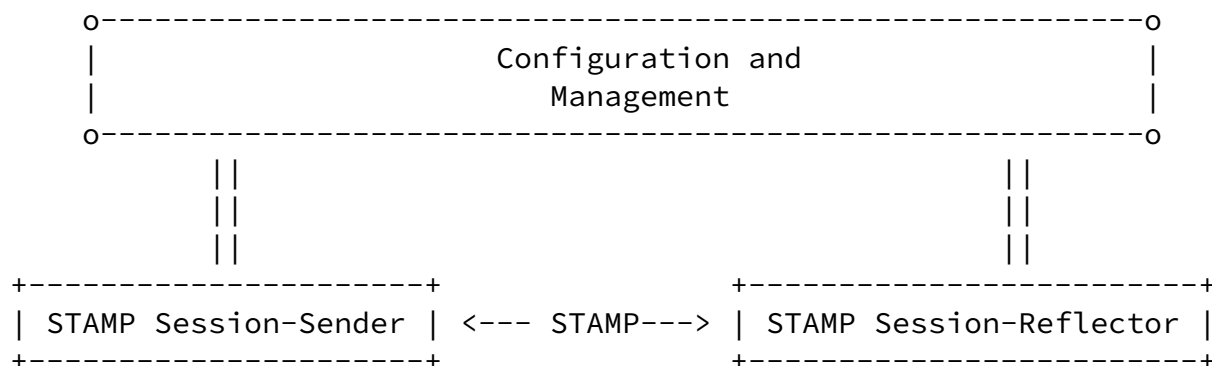
                   Figure 1: STAMP Reference Model

4. Theory of Operation

   STAMP Session-Sender transmits test packets over UDP transport toward
   STAMP Session-Reflector. A STAMP Session-Sender MUST use UDP port
   862 (TWAMP-Test Receiver Port) as the default destination UDP port
   number. A STAMP implementation of Session-Sender MUST be able to use
   UDP port numbers from User, a.k.a. Registered, Ports and Dynamic,
   a.k.a. 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 by all users of
   the network.

STAMP Session-Reflector receives Session-Sender's packet and acts
according to the configuration and optional control information
communicated in the Session-Sender's test packet. An implementation
of STAMP Session-Reflector by default MUST use receive STAMP test
packets on UDP port 862. An implementation of 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
that 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.

STAMP supports two modes: unauthenticated and authenticated.
Unauthenticated STAMP test packets, defined in Section 4.1.1 and
Section 4.2.1, ensure interworking between STAMP and TWAMP Light as
described in Section 4.4 packet formats.

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

4.1. Session-Sender Behavior and Packet Format

Because STAMP supports symmetrical test packets, 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.

4.1.1. Session-Sender Packet Format in Unauthenticated Mode

STAMP 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)                      |
        |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
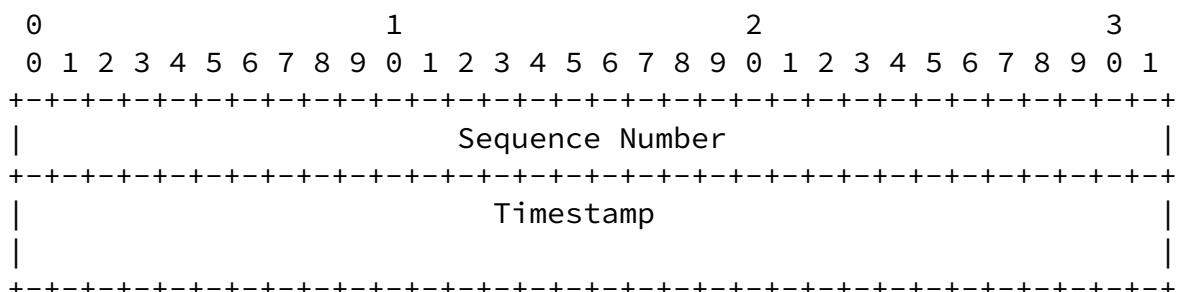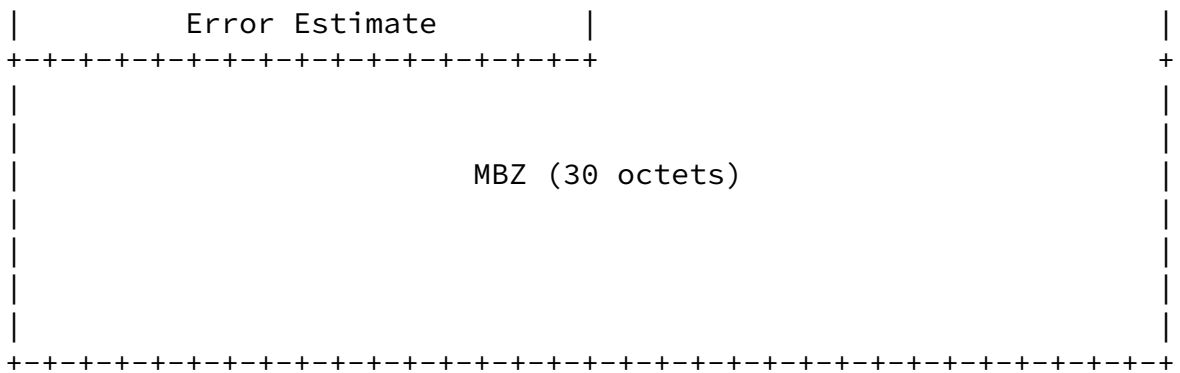
        Figure 2: STAMP Session-Sender test packet format in unauthenticated
                                   mode

   where fields are defined as the following:

   o Sequence Number is four octets long field. For each new session
      its value starts at zero and is incremented with each transmitted
      packet.

   o Timestamp is eight octets long field. STAMP node MUST support
      Network Time Protocol (NTP) version 4 64-bit timestamp format
      [RFC5905], the format used in [RFC5357]. STAMP node MAY support
      IEEE 1588v2 Precision Time Protocol truncated 64-bit timestamp
      format [IEEE.1588.2008], the format used in [RFC8186].

   o Error Estimate is two octets long field with 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  |
           +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
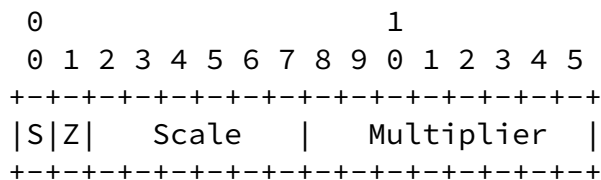
                        Figure 3: Error Estimate Format

   where S, Scale, and Multiplier fields are interpreted as they have
   been defined in section 4.1.2 [RFC4656]; and Z field - as has been
   defined in section 2.3 [RFC8186]:

* 0 - NTP 64 bit format of a timestamp;

        * 1 - PTPv2 truncated format of a timestamp.

        The STAMP Session-Sender and Session-Reflector MAY use, not use,
        or set value of the Z field in accordance with the timestamp
        format in use. This optional field is to enhance operations, but
        local configuration or defaults could be used in its place.

    o May-be-Zero (MBZ) field in the session-sender unauthenticated
       packet is 30 octets long. It MAY be all zeroed on the
       transmission and MUST be ignored on receipt.

4.1.2. Session-Sender Packet Format in Authenticated Mode

    STAMP 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,
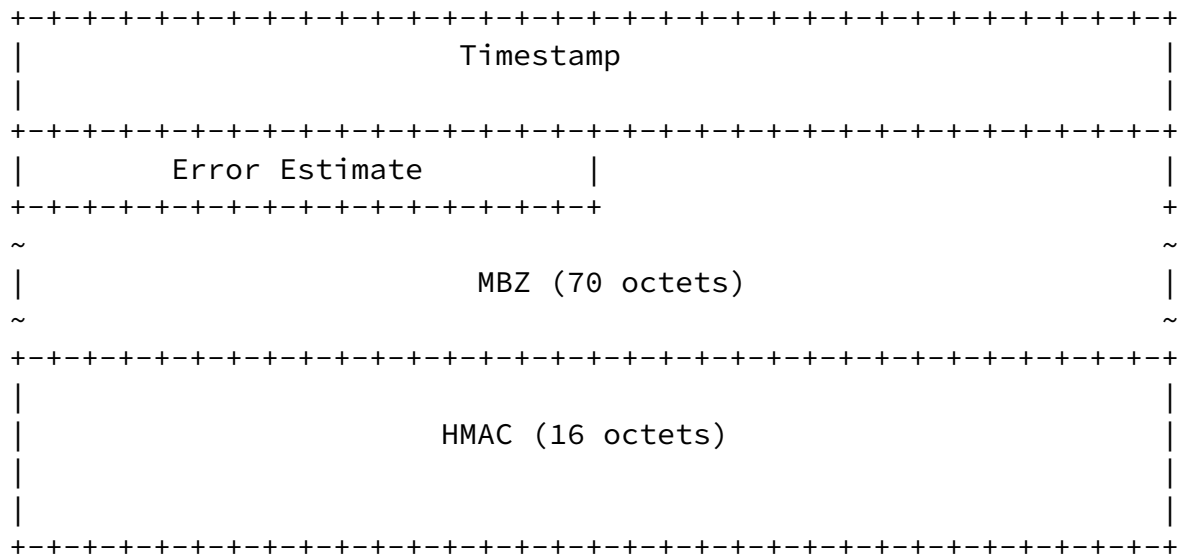   listed in Section 4.1.1. Also, MBZ fields are used to align the
   packet on 16 octets boundary. The value of the field MAY be zeroed
   on transmission and MUST be ignored on receipt. Also, the packet
   includes a key-hashed message authentication code (HMAC) ([RFC2104])
   hash at the end of the PDU. The detailed use of the HMAC field is
   described in Section 4.3.

4.2. Session-Reflector Behavior and Packet Format

   The Session-Reflector receives the STAMP test packet, verifies it,
   prepares and transmits the reflected test packet.

   Two modes of STAMP Session-Reflector characterize the expected
   behavior and, consequently, performance metrics that can be measured:

   o Stateless - STAMP Session-Reflector does not maintain test state
      and will reflect the received sequence number without
      modification. As a result, only round-trip packet loss can be
      calculated while the reflector is operating in stateless mode.

   o Stateful - STAMP Session-Reflector maintains test state thus

enabling the ability to determine forward loss, gaps recognized in
the received sequence number. As a result, both near-end
(forward) and far-end (backward) packet loss can be computed.
That implies that the STAMP Session-Reflector MUST keep a state
for each accepted STAMP-test session, uniquely identifying STAMP-
test packets to one such session instance, and enabling adding a
sequence number in the test reply that is individually incremented
on a per-session basis.

4.2.1. Session-Reflector Packet Format in Unauthenticated Mode
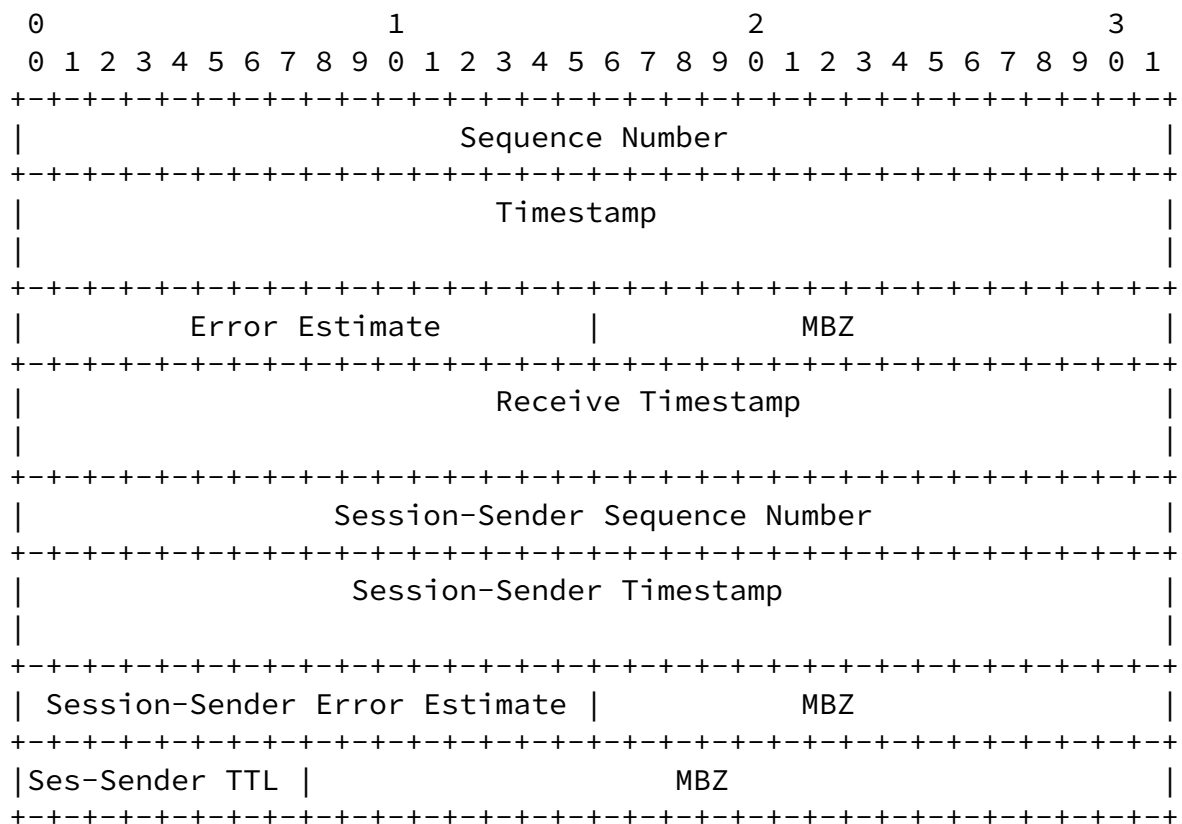
   For unauthenticated mode:

```
    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

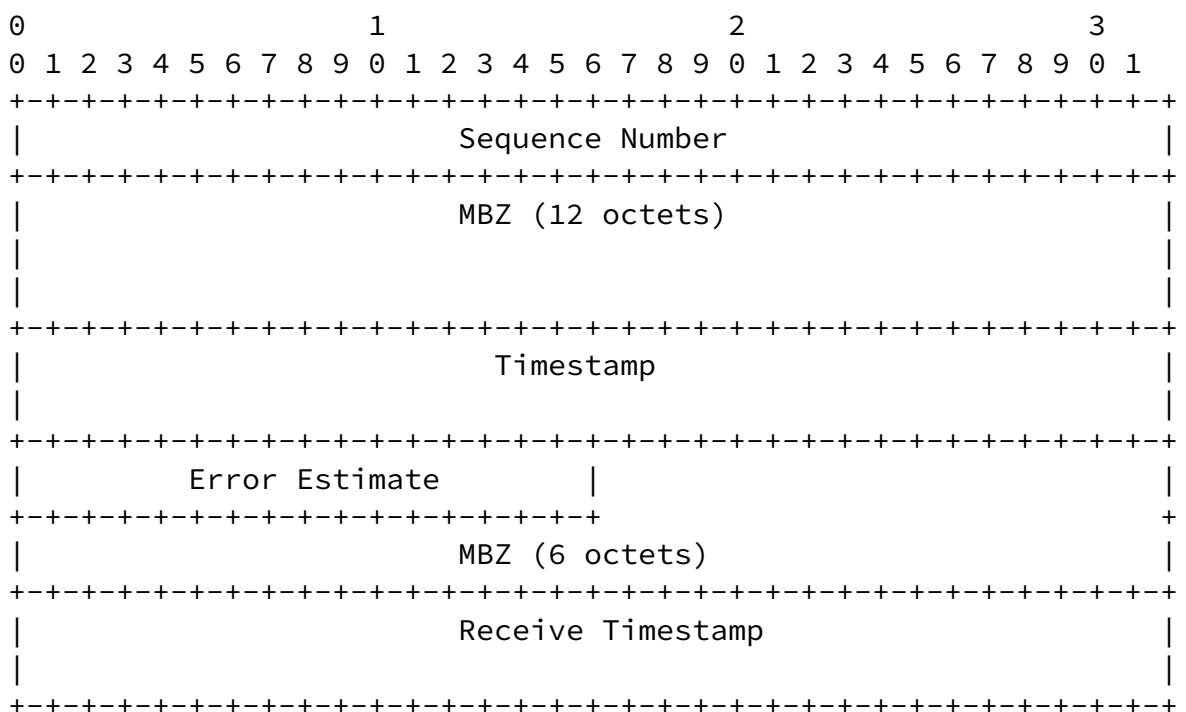   where fields are defined as the following:

   o Sequence Number is four octets long field. 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 received
>   STAMP test packets in each test session and uses that counter
>   to set the value of the Sequence Number field.

o Timestamp and Receiver Timestamp fields are each eight octets
   long. The format of these fields, NTP or PTPv2, indicated by the
   Z flag of the Error Estimate field as described in [Section 4.1](#).

o Error Estimate has the same size and interpretation as described
   in [Section 4.1](#).

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

o Session-Sender TTL is one octet long field, and its value is the
   copy of the TTL field in IPv4 (or Hop Limit in IPv6) from the
   received STAMP test packet.

o MBZ is used to achieve alignment on a four octets boundary. The
   value of the field MAY be zeroed on transmission and MUST be
   ignored on receipt.

[4.2.2](#). Session-Reflector Packet Format in Authenticated Mode
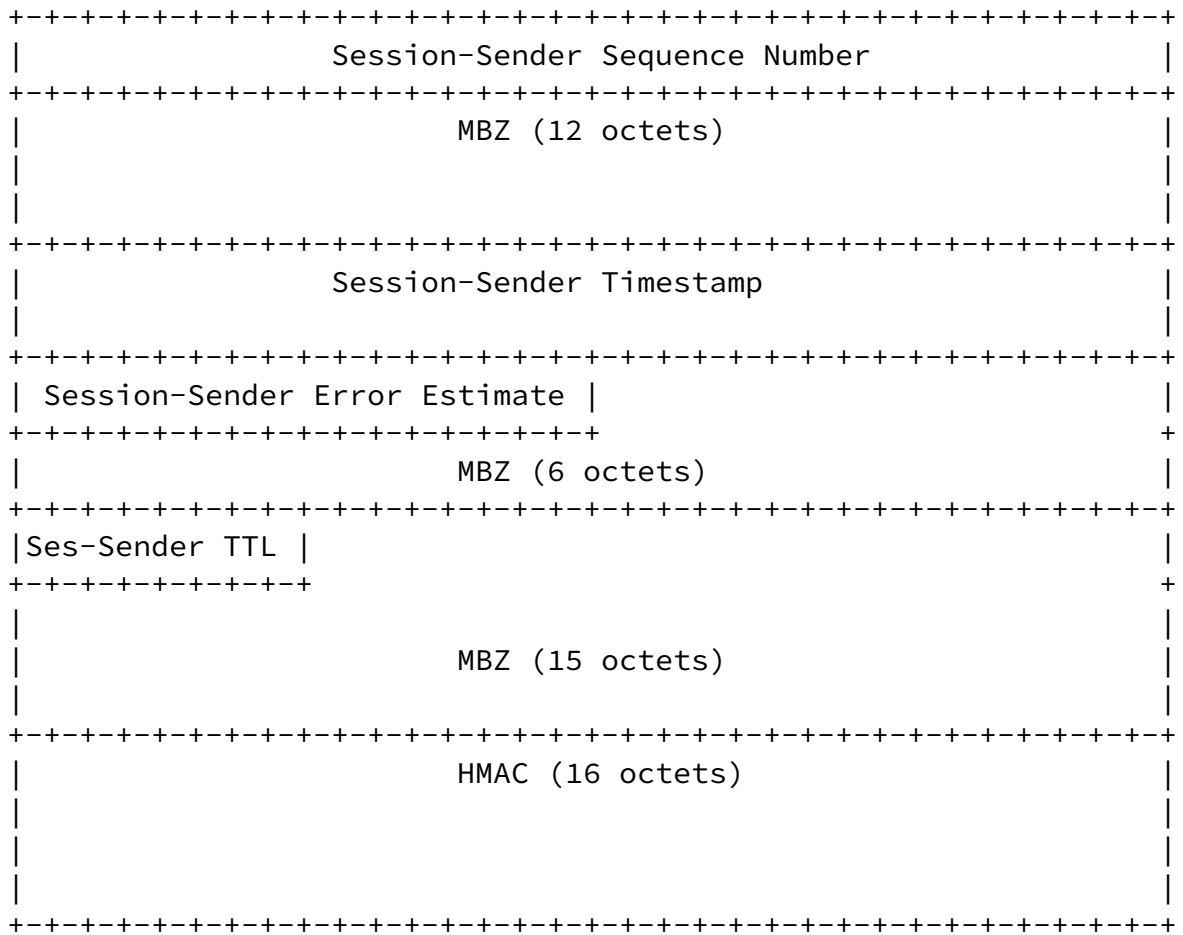
   For the authenticated mode:

```
 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 4.2.1. Additionally, the MBZ field is used to
   align the packet on 16 octets boundary. The value of the field MAY
   be zeroed on transmission and MUST be ignored on receipt. Also,
   STAMP Session-Reflector test packet format in authenticated mode
   includes a key (HMAC) ([RFC2104]) hash at the end of the PDU. The
   detailed use of the HMAC field is in Section 4.3.

4.3. Integrity and Confidentiality Protection in STAMP

To provide integrity protection, each STAMP message is being
authenticated 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. HMAC uses its own key, and the definition of the
mechanism to distribute the HMAC key is outside the scope of this
specification. One example is to use an orchestrator to configure
HMAC key based on STAMP YANG data model [I-D.ietf-ippm-stamp-yang].

HMAC MUST be verified as early as possible to avoid using or
propagating corrupted data.

If confidentiality protection for STAMP is required, encryption at
the higher level MUST be used. For example, STAMP packets could be
transmitted in the dedicated IPsec tunnel or share the IPsec tunnel
with the monitored flow.

4.4. Interoperability with TWAMP Light

One of the essential requirements to STAMP is the ability to
interwork with a TWAMP Light device. There are two possible
combinations for such use case:

o STAMP Session-Sender with TWAMP Light Session-Reflector;

o TWAMP Light Session-Sender with STAMP Session-Reflector.

In the former case, the Session-Sender MAY 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 defined
in [RFC8545]. Thus STAMP Session-Sender MAY use port numbers as
defined in Section 4. If any of STAMP extensions are used, the TWAMP
Light Session-Reflector will view them as Packet Padding field. The
Session-Sender SHOULD use the default format for its timestamps -
NTP. And it MAY use PTPv2 timestamp format.

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
STAMP Session-Reflector to use UDP port number as defined in
Section 4. If the TWAMP Light Session-Sender includes Packet Padding
field in its transmitted packet, the STAMP Session-Reflector will

return the reflected packet of the symmetrical size if the size of
the received test packet is larger than the size of the STAMP base
packet. The Session-Reflector MUST be set to use the default format
for its timestamps, NTP.

STAMP does not support the Reflect Octets capability defined in
[RFC6038]. If the Server Octets field is present in the TWAMP
Session-Sender packet, STAMP Session-Reflector will not copy the
content starting from the Server Octets field but will transmit the
reflected packet of equal size.

5. IANA Considerations

This document doesn't have any IANA action. This section may be
removed before the publication.

6. Security Considerations

In general, all the security considerations related to TWAMP-Test,
discussed in [RFC5357] apply to STAMP. Since STAMP uses the well-
known UDP port number allocated for the OWAMP-Test/TWAMP-Test
Receiver port, the security considerations and measures to mitigate
the risk of the attack using the registered port number documented in
Section 6 [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.

   Load of 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.
   [RFC8085] section 3.1.5 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].

STAMP test packets can be transmitted with the destination UDP port
number from the User Ports range, as defined in Section 4, that is
already or will be assigned by IANA. The possible impact of the

STAMP test packets on the network MUST be thoroughly analyzed, and
the use of STAMP for each case MUST be agreed by all users on the
network before starting the STAMP test session.

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

7. Acknowledgments

Authors express their appreciation to Jose Ignacio Alvarez-Hamelin
and Brian Weis for their great insights into the security and
identity protection, and the most helpful and practical suggestions.
Also, our sincere thanks to David Ball and Rakesh Gandhi or their
thorough reviews and helpful comments.

8. References

8.1. Normative References

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

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

   [I-D.ietf-ippm-stamp-option-tlv]
                 Mirsky, G., Xiao, M., Jun, G., Nydell, H., and R. Foote,
                 "Simple Two-way Active Measurement Protocol Optional
                 Extensions", draft-ietf-ippm-stamp-option-tlv-00 (work in
                 progress), July 2019.

    [I-D.ietf-ippm-stamp-yang]
               Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active
               Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-
               stamp-yang-03 (work in progress), March 2019.

    [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>.

    [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>.

Authors' Addresses

    Greg Mirsky
    ZTE Corp.

    Email: gregimirsky@gmail.com

    Guo Jun
    ZTE Corporation
    68# Zijinghua Road
    Nanjing, Jiangsu 210012

   P.R.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