Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 24, 2020

G. Mirsky
X. Min
ZTE Corp.
H. Nydell
Accedian Networks
R. Foote
Nokia
A. Masputra
Apple Inc.
E. Ruffini
OutSys
February 21, 2020

Simple Two-way Active Measurement Protocol Optional Extensions
draft-ietf-ippm-stamp-option-tlv-03

Abstract

This document describes optional extensions to Simple Two-way Active Measurement Protocol (STAMP) which enable measurement performance metrics in addition to ones supported by the STAMP base specification.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Simple Two-way Active Measurement Protocol (STAMP) [I-D.ietf-ippm-stamp] supports the use of optional extensions that use Type-Length-Value (TLV) encoding. Such extensions are to enhance the STAMP base functions, such as measurement of one-way and round-trip delay, latency, packet loss, as well as ability to detect packet duplication and out-of-order delivery of the test packets. This specification provides definitions of optional STAMP extensions, their formats, and theory of operation.

2. Conventions used in this document

2.1. Terminology

STAMP - Simple Two-way Active Measurement Protocol

DSCP - Differentiated Services Code Point

ECN - Explicit Congestion Notification

NTP - Network Time Protocol

PTP - Precision Time Protocol

HMAC Hashed Message Authentication Code

TLV Type-Length-Value

BITS Building Integrated Timing Supply

SSU Synchronization Supply Unit

GPS Global Positioning System

GLONASS Global Orbiting Navigation Satellite System

LORAN-C Long Range Navigation System Version C

MBZ Must Be Zeroed

CoS Class of Service

PMF Performance Measurement Function

SSID STAMP Session Identifier

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. STAMP Test Session Identifier

STAMP Session-Sender transmits test packets to STAMP Session-Reflector. 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. 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 are compatible on the wire with unauthenticated TWAMP-Test [RFC5357] packet formats.

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

A STAMP Session is identified using 4-tuple (source and destination IP addresses, source and destination UDP port numbers). A STAMP Session-Sender MAY generate locally unique STAMP Session Identifier (SSID). SSID is two octets long non-zero unsigned integer. A Session-Sender MAY use SSID to identify a STAMP test session. If SSID is used, it MUST be present in each test packet of the given test session. In the unauthenticated mode, SSID is located, as displayed in Figure 1.

```
     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        |             SSID              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                                                               |
    |                         MBZ (28 octets)                       |
    |                                                               |
    |                                                               |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type              |           Length              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                            Value                              ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: STAMP Session-Sender test packet format with TLV in unauthenticated mode

An implementation of STAMP Session-Reflector that supports this specification SHOULD identify a STAMP Session using the SSID in combination with elements of the usual 4-tuple. A conforming implementation of STAMP Session-Reflector MUST copy the SSID value from the received test packet and put it into the reflected packet as displayed in Figure 2.

```
     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        |           SSID                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                          Receive Timestamp                    |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                 Session-Sender Sequence Number                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Session-Sender Timestamp                     |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Session-Sender Error Estimate |           MBZ                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Ses-Sender TTL |               Reserved                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type              |           Length              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                            Value                              ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: STAMP Session-Reflector test packet format with TLV in unauthenticated mode

A STAMP Session-Reflector that does not support this specification, will return the zeroed SSID field in the reflected STAMP test packet. The Session-Sender MUST NOT stop the session if it receives a zeroed SSID field.

In the authenticated mode, location of SSID field is shown in Figure 3 and Figure 4.

```
     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         |             SSID              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                                                               ~
    |                         MBZ (68 octets)                       |
    ~                                                               ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                       HMAC (16 octets)                        |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: STAMP Session-Sender test 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       |             SSID              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                        MBZ (4 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 4: STAMP Session-Reflector test packet format in authenticated mode

4. TLV Extensions to STAMP

Type-Length-Value (TLV) encoding scheme provides flexible extension mechanism for optional informational elements. TLV is an optional field in the STAMP test packet. TLVs have the two octets long Type field, two octets long Length field that is the length of the Value field in octets. Type values, see Section 5.1, less than 32768 identify mandatory TLVs that MUST be supported by an implementation. Type values greater than or equal to 32768 identify optional TLVs that SHOULD be ignored if the implementation does not understand or support them. If a Type value for TLV or sub-TLV is in the range for Vendor Private Use, the Length MUST be at least 4, and the first four octets MUST be that vendor's the Structure of Management Information (SMI) Private Enterprise Number, in network octet order. The rest of the Value field is private to the vendor. Following sections describe the use of TLVs for STAMP that extend STAMP capability beyond its base specification.

A STAMP node, whether Session-Sender or Session-Reflector, receiving a test packet MUST determine whether the packet is a base STAMP packet or includes one or more TLVs. The node MUST compare the value in the Length field of the UDP header and the length of the base STAMP test packet in the mode, unauthenticated or authenticated based on the configuration of the particular STAMP test session. If the difference between the two values is larger than the length of UDP header, then the test packet includes one or more STAMP TLVs that immediately follow the base STAMP test packet.

A system that has received a STAMP test packet with extension TLVs MUST validate each fixed-size TLV by verifying that the value in the Length field equals the value defined for the particular type. If the values are not equal, the processing of extension TLVs MUST be stopped and the event logged (logging SHOULD be throttled). Also, if the system is the Session-Reflector in that test, it MUST send (transmission of ICMP Error messages SHOULD be throttled) the ICMP Parameter Problem message with Code set to 0 and the Pointer referring to the Length field of the TLV.

4.1. Extra Padding TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Extra Padding Type       |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         Extra Padding                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Extra Padding TLV

where fields are defined as the following:

- o Extra Padding Type - TBA1 allocated by IANA Section 5.1

- o Length - two octets long field equals length on the Extra Padding field in octets.

- o Extra Padding - a pseudo-random sequence of numbers. The field MAY be filled with all zeroes.

The Extra Padding TLV is similar to the Packet Padding field in TWAMP-Test packet [RFC5357]. The Extra Padding TLV MUST be used to create STAMP test packets of larger size. The Extra Padding TLV MUST be the last TLV in a STAMP test packet.

4.2. Location TLV

STAMP session-sender MAY include the Location TLV to request information from the session-reflector. The session-sender SHOULD NOT fill any information fields except for Type and Length. The session-reflector MUST validate the Length value against the address family of the transport encapsulating the STAMP test packet. If the value of the Length field is invalid, the session-reflector MUST zero all fields and MUST NOT return any information to the session-sender. The session-reflector MUST ignore all other fields of the received Location TLV.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Location Type        |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Source MAC                           |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                     Destination IP Address                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        Source IP Address                      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Destination Port       |          Source Port          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Session-Reflector Location TLV

where fields are defined as the following:

- Location Type - TBA2 allocated by IANA Section 5.1

- Length - two octets long field equals length on the Value field in octets. Length field value MUST be 20 octets for the IPv4 address family. For the IPv6 address family value of the Length field MUST be 44 octets. All other values are invalid.

- Source MAC - 6 octets 48 bits long field. The session-reflector MUST copy Source MAC of received STAMP packet into this field.

- Reserved - two octets long field. MUST be zeroed on transmission and ignored on reception.

- Destination IP Address - IPv4 or IPv6 destination address of the received by the session-reflector STAMP packet.

- o Source IP Address - IPv4 or IPv6 source address of the received by the session-reflector STAMP packet.

- o Destination Port - two octets long UDP destination port number of the received STAMP packet.

- o Source Port - two octets long UDP source port number of the received STAMP packet.

The Location TLV MAY be used to determine the last-hop addressing for STAMP packets including source and destination IP addresses as well as the MAC address of the last-hop router. Last-hop MAC address MAY be monitored by the Session-Sender whether there has been a path switch on the last hop, closest to the Session-Reflector. The IP addresses and UDP port will indicate if there is a NAT router on the path, and allows the Session-Sender to identify the IP address of the Session-Reflector behind the NAT, detect changes in the NAT mapping that could cause sending the STAMP packets to the wrong Session-Reflector.

4.3. Timestamp Information TLV

STAMP session-sender MAY include the Timestamp Information TLV to request information from the session-reflector. The session-sender SHOULD NOT fill any information fields except for Type and Length. The session-reflector MUST validate the Length value of the STAMP test packet. If the value of the Length field is invalid, the session-reflector MUST zero all fields and MUST NOT return any information to the session-sender.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Timestamp Information Type   |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sync. Src In | Timestamp In  | Sync. Src Out | Timestamp Out |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Timestamp Information TLV

where fields are defined as the following:

- o Timestamp Information Type - TBA3 allocated by IANA Section 5.1

- o Length - two octets long field, equals four octets.

- o Sync Src In - one octet long field that characterizes the source of clock synchronization at the ingress of Session-Reflector.

There are several of methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905], Precision Time Protocol (PTP) [IEEE.1588.2008], Synchronization Supply Unit (SSU) or Building Integrated Timing Supply (BITS), or Global Positioning System (GPS), Global Orbiting Navigation Satellite System (GLONASS) and Long Range Navigation System Version C (LORAN-C). The value is one of the listed in Table 4.

- Timestamp In - one octet long field that characterizes the method by which the ingress of Session-Reflector obtained the timestamp T2. A timestamp may be obtained with hardware assist, via software API from a local wall clock, or from a remote clock (the latter referred to as "control plane"). The value is one of the listed in Table 6.

- Sync Src Out - one octet long field that characterizes the source of clock synchronization at the egress of Session-Reflector. The value is one of the listed in Table 4.

- Timestamp Out - one octet long field that characterizes the method by which the egress of Session-Reflector obtained the timestamp T3. The value is one of the listed in Table 6.

4.4. Class of Service TLV

The STAMP session-sender MAY include Class of Service (CoS) TLV in the STAMP test packet. If the CoS TLV is present in the STAMP test packet and the value of the DSCP1 field is zero, then the STAMP session-reflector MUST copy the values of Differentiated Services Code Point (DSCP) ECN fields from the received STAMP test packet into DSCP2 and ECN fields respectively of the CoS TLV of the reflected STAMP test packet. If the value of the DSCP1 field is non-zero, then the STAMP session-reflector MUST use DSCP1 value from the CoS TLV in the received STAMP test packet as DSCP value of STAMP reflected test packet and MUST copy DSCP and ECN values of the received STAMP test packet into DSCP2 and ECN fields of Class of Service TLV in the STAMP reflected a packet. The Session-Sender, upon receiving the reflected packet, will save the DSCP and ECN values for analysis of the CoS in the reverse direction.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Class of Service Type    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   DSCP1   |   DSCP2   |ECN|             Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Class of Service TLV

where fields are defined as the following:

- o Class of Service Type - TBA4 allocated by IANA Section 5.1
- o Length - two octets long field, equals four octets.
- o DSCP1 - The Differentiated Services Code Point (DSCP) intended by the Session-Sender. To be used as the return DSCP from the Session-Reflector.
- o DSCP2 - The received value in the DSCP field at the Session-Reflector in the forward direction.
- o ECN - The received value in the ECN field at the Session-Reflector in the forward direction.
- o Reserved - 18 bits long field, must be zeroed in transmission and ignored on receipt.

A STAMP Session-Sender that includes the CoS TLV sets the value of the DSCP1 field and zeroes the value of the DSCP2 field. A STAMP Session-Reflector that received the test packet with the CoS TLV MUST include the CoS TLV in the reflected test packet. Also, the Session-Reflector MUST copy the value of the DSCP field of the IP header of the received STAMP test packet into the DSCP2 field in the reflected test packet. And, at last, the Session-Reflector MUST set the value of the DSCP field in the IP header of the reflected test packet equal to the value of the DSCP1 field of the test packet it has received.

Re-mapping of CoS in some use cases, for example, in mobile backhaul networks is used to provide multiple services, i.e., 2G, 3G, LTE, over the same network. But if it is misconfigured, then it is often difficult to diagnose the root cause of the problem that is viewed as an excessive packet drop of higher level service while packet drop for lower service packets is at a normal level. Using CoS TLV in STAMP test helps to troubleshoot the existing problem and also verify whether DiffServ policies are processing CoS as required by the configuration.

4.5. Direct Measurement TLV

The Direct Measurement TLV enables collection of "in profile" IP packets that had been transmitted and received by the Session-Sender and Session-Reflector respectfully. The definition of "in-profile packet" is outside the scope of this document.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Direct Measurement Type    |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Session-Sender Tx counter  (S_TxC)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Session-Reflector Rx counter  (R_RxC)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Session-Reflector Tx counter  (R_TxC)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Direct Measurement TLV

where fields are defined as the following:

- Direct Measurement Type - TBA5 allocated by IANA Section 5.1
- Length - two octets long field equals length on the Value field in octets. Length field value MUST be 12 octets.
- Session-Sender Tx counter (S_TxC) is four octets long field.
- Session-Reflector Rx counter (R_RxC) is four octets long field. MUST be zeroed by the Session-Sender and filled by the Session-Reflector.
- Session-Reflector Tx counter (R_TxC) is four octets long field. MUST be zeroed by the Session-Sender and filled by the Session-Reflector.

4.6. Access Report TLV

A STAMP Session-Sender MAY include Access Report TLV (Figure 10) to indicate changes to the access network status to the Session-Reflector. The definition of an access network is outside the scope of this document.

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |       Access Report Type      |           Length              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |   Access ID   |  Return Code  |          Reserved             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Access Report TLV

where fields are defined as follows:

- o Access Report Type - TBA6 allocated by IANA Section 5.1.
- o Length - two octets long field, equals four octets.
- o Access ID - one octet long field that identifies the access network, e.g., 3GPP (Radio Access Technologies specified by 3GPP) or Non-3GPP (accesses that are not specified by 3GPP) [TS23501]. The value is one of Section 5.4.
- o Return Code - one octet long field that identifies the report signal, e.g., available, unavailable. The value is one of Section 5.5.
- o Reserved - two octets long field, must be zeroed on transmission and ignored on receipt.

The STAMP Session-Sender that includes the Access Report TLV sets the value of the Access ID field according to the type of access network it reports on. Also, the Session-Sender sets the value of the Return Code field to reflect the operational state of the access network. The mechanism to determine the state of the access network is outside the scope of this specification. A STAMP Session-Reflector that received the test packet with the Access Report TLV MUST include the Access Report TLV in the reflected test packet. The Session-Reflector MUST set the value of the Access ID and Return Code fields equal to the values of the corresponding fields from the test packet it has received.

The Session-Sender MUST also arm a retransmission timer after sending a test packet that includes the Access Report TLV. This timer MUST be disarmed upon the reception of the reflected STAMP test packet that includes Access Report TLV. In the event the timer expires before such a packet is received, the Session-Sender MUST retransmit the STAMP test packet that contains the Access Report TLV. This retransmission SHOULD be repeated up to four times before the procedure is aborted. Setting the value for the retransmission timer

is based on local policies, network environment. The default value of the retransmission timer for Access Report TLV SHOULD be three seconds. An implementation MUST provide control of the retransmission timer value and the number of retransmissions.

The Access Report TLV is used by the Performance Measurement Function (PMF) components of the Access Steering, Switching and Splitting feature for 5G networks [TS23501]. The PMF component in the User Equipment acts as the STAMP Session-Sender, and the PMF component in the User Plane Function acts as the STAMP Session-Reflector.

4.7. Follow-up Telemetry TLV

A Session-Reflector might be able to put in the Timestamp field only a "SW Local" (see Table 6) timestamp. But the hosting system might provide the timestamp closer to the start of actual packet transmission even though when it is not possible to deliver the information to the Session-Sender in the packet itself. This timestamp might nevertheless be important for the Session-Sender, as it helps in to improve the accuracy of measuring network delay by minimizing the impact of egress queuing delays on the measurement.

A STAMP Session-Sender MAY include the Follow-up Telemetry TLV to request information from the Session-Reflector. The Session-Sender MUST set the Follow-up Telemetry Type and Length fields to their appropriate values. Sequence Number and Timestamp fields MUST be zeroed on transmission by the Session-Sender and ignored by the Session-Reflector upon receipt of the STAMP test packet that includes the Follow-up Telemetry TLV. The Session-Reflector MUST validate the Length value of the STAMP test packet. If the value of the Length field is invalid, the Session-Reflector MUST zero Sequence Number and Timestamp fields. If the Session-Reflector is in stateless mode (defined in Section 4.2 [I-D.ietf-ippm-stamp]), it MUST zero Sequence Number and Timestamp fields.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Follow-up Telemetry Type    |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Follow-up Timestamp                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Timestamp M  |                    Reserved                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Follow-up Telemetry TLV

where fields are defined as follows:

- o Follow-up Telemetry Type - TBA7 allocated by IANA Section 5.1.

- o Length - two octets long field, equals 16 octets.

- o Sequence Number - four octets long field indicating the sequence number of the last packet reflected in the same STAMP-test session. Since the Session-Reflector runs in the stateful mode (defined in Section 4.2 [I-D.ietf-ippm-stamp]), it is the Session-Reflector's Sequence Number of the previous reflected packet.

- o Follow-up Timestamp - eight octets long field, with the format indicated by the Z flag of the Error Estimate field of the packet transmitted by a Session-Reflector, as described in Section 4.1 [I-D.ietf-ippm-stamp]. It carries the timestamp when the reflected packet with the specified sequence number was sent..

- o Timestamp M(ode) - one octet long field that characterizes the method by which the entity that transmits a reflected STAMP packet obtained the Follow-up Timestamp. The value is one of the listed in Table 6.

- o Reserved - the three octest-long field. Its value MUST be zeroed on transmission and ignored on receipt.

4.8. HMAC TLV

The STAMP authenticated mode protects the integrity of data collected in STAMP base packet. STAMP extensions are designed to provide valuable information about the condition of a network, and protecting the integrity of that data is also essential. The keyed Hashed Message Authentication Code (HMAC) TLV MUST be included in a STAMP

test packet in the authenticated mode, excluding when the only TLV present is Extra Padding TLV. The HMAC TLV MUST follow all TLVs included in a STAMP test packet, except for the Extra Padding TLV. The HMAC TLV MAY be used to protect the integrity of STAMP extensions in STAMP 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            HMAC Type          |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             HMAC                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: HMAC TLV

where fields are defined as follows:

o HMAC Type - is two octets long field, value TBA8 allocated by IANA Section 5.1.

o Length - two octets long field, equals 16 octets.

o HMAC - is 16 octets long field that carries HMAC digest of the text of all preceding TLVs.

As defined in [I-D.ietf-ippm-stamp], STAMP uses HMAC-SHA-256 truncated to 128 bits ([RFC4868]). All considerations regarding using the key and key distribution and management listed in Section 4.4 of [I-D.ietf-ippm-stamp] are fully applicable to the use of the HMAC TLV. HMAC is calculated as defined in [RFC2104] over text as the concatenation of all preceding TLVs. The digest then MUST be truncated to 128 bits and written into the HMAC field. In the authenticated mode, HMAC MUST be verified before using any data in the included STAMP TLVs. If HMAC verification by the Session-Reflector fails, then an ICMP Parameter Problem message MUST be generated (with consideration of limiting the rate of error messages). The Code value MUST be set to 0 and the Pointer identifying HMAC Type. Also, both Session-Sender and Session-Reflector SHOULD log the notification that HMAC verification of STAMP TLVs failed. The packet that failed HMAC verification MUST be dropped.

5. IANA Considerations

5.1. STAMP TLV Registry

IANA is requested to create the STAMP TLV Type registry. All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

Value	Description	Reference
0	Reserved	This document
1- 32767	Mandatory TLV, unassigned	IETF Review
32768 - 65279	Optional TLV, unassigned	First Come First Served
65280 - 65519	Experimental	This document
65520 - 65534	Private Use	This document
65535	Reserved	This document

Table 1: STAMP TLV Type Registry

This document defines the following new values in the STAMP TLV Type registry:

Value	Description	Reference
TBA1	Extra Padding	This document
TBA2	Location	This document
TBA3	Timestamp Information	This document
TBA4	Class of Service	This document
TBA5	Direct Measurement	This document
TBA6	Access Report	This document
TBA7	Follow-up Telemetry	This document
TBA8	HMAC	This document

Table 2: STAMP Types

5.2. Synchronization Source Sub-registry

IANA is requested to create Synchronization Source sub-registry as part of STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

Value	Description	Reference
0	Reserved	This document
1- 127	Unassigned	IETF Review
128 - 239	Unassigned	First Come First Served
240 - 249	Experimental	This document
250 - 254	Private Use	This document
255	Reserved	This document

Table 3: Synchronization Source Sub-registry

This document defines the following new values in the Synchronization Source sub-registry:

Value	Description	Reference
1	NTP	This document
2	PTP	This document
3	SSU/BITS	This document
4	GPS/GLONASS/LORAN-C	This document
5	Local free-running	This document

Table 4: Synchronization Sources

5.3. Timestamping Method Sub-registry

IANA is requested to create Timestamping Method sub-registry as part of STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

Value	Description	Reference
0	Reserved	This document
1- 127	Unassigned	IETF Review
128 - 239	Unassigned	First Come First Served
240 - 249	Experimental	This document
250 - 254	Private Use	This document
255	Reserved	This document

Table 5: Timestamping Method Sub-registry

This document defines the following new values in the Timestamping Methods sub-registry:

Value	Description	Reference
1	HW Assist	This document
2	SW local	This document
3	Control plane	This document

Table 6: Timestamping Methods

5.4. Access ID Sub-registry

IANA is requested to create Access ID sub-registry as part of STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 7:

Value	Description	Reference
0	Reserved	This document
1- 127	Unassigned	IETF Review
128 - 239	Unassigned	First Come First Served
240 - 249	Experimental	This document
250 - 254	Private Use	This document
255	Reserved	This document

Table 7: Access ID Sub-registry

This document defines the following new values in the Access ID sub-registry:

Value	Description	Reference
1	3GPP	This document
2	Non-3GPP	This document

Table 8: Access IDs

5.5. Return Code Sub-registry

IANA is requested to create Return Code sub-registry as part of STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 7:

Value	Description	Reference
0	Reserved	This document
1- 127	Unassigned	IETF Review
128 - 239	Unassigned	First Come First Served
240 - 249	Experimental	This document
250 - 254	Private Use	This document
255	Reserved	This document

Table 9: Return Code Sub-registry

This document defines the following new values in the Return Code sub-registry:

Value	Description	Reference
1	Network available	This document
2	Network unavailable	This document

Table 10: Return Codes

6. Security Considerations

Use of HMAC in authenticated mode may be used to simultaneously verify both the data integrity and the authentication of the STAMP test packets.

7. Acknowledgments

Authors much appreciate the thorough review and thoughtful comments received from Tianran Zhou.

8. Contributors

The following people contributed text to this document:

Guo Jun
ZTE Corporation
68# Zijinghua Road
Nanjing, Jiangsu 210012
P.R.China

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

9. References

9.1. Normative References

[I-D.ietf-ippm-stamp]
Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-way Active Measurement Protocol", draft-ietf-ippm-stamp-10 (work in progress), October 2019.

[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>.

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

9.2. Informative References

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

[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>.

[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>.

[TS23501] 3GPP (3rd Generation Partnership Project), "Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2 (Release 16)", 3GPP TS23501, 2019.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

Henrik Nydell
Accedian Networks

Email: hnydell@accedian.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

Adi Masputra
Apple Inc.
One Apple Park Way
Cupertino, CA 95014
USA

Email: adi@apple.com

Ernesto Ruffini
OutSys
via Caracciolo, 65
Milano 20155
Italy

Email: eruffini@outsys.org