Stream: Internet Engineering Task Force (IETF)
RFC: 8972
Updates: 8762
Category: Standards Track
Published: January 2021
ISSN: 2070-1721

Authors:

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

RFC 8972
Simple Two-Way Active Measurement Protocol Optional Extensions

Abstract

This document describes optional extensions to Simple Two-way Active Measurement Protocol (STAMP) that enable measurement of performance metrics. The document also defines a STAMP Test Session Identifier and thus updates RFC 8762.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] defines the STAMP base functionalities. This document specifies the use of optional extensions that use Type-Length-Value (TLV) encoding. Such extensions enhance the STAMP base functions, such as measurement of one-way and round-trip delay, latency, packet loss, packet duplication, and out-of-order delivery of test packets. This specification defines optional STAMP extensions, their formats, and the theory of operation. Also, a STAMP Test Session Identifier is defined as an update of the base STAMP specification [RFC8762].

2. Conventions Used in This Document

2.1. Acronyms

BDS	BeiDou Navigation Satellite System
BITS	Building Integrated Timing Supply
CoS	Class of Service
DSCP	Differentiated Services Code Point
ECN	Explicit Congestion Notification
GLONASS	Global Orbiting Navigation Satellite System
GPS	Global Positioning System [GPS]
HMAC	Hashed Message Authentication Code
LORAN-C	Long Range Navigation System Version C
MBZ	Must Be Zero
NTP	Network Time Protocol [RFC5905]
PMF	Performance Measurement Function
PTP	Precision Time Protocol [IEEE.1588.2008]
RP	Reverse Path
SMI	Structure of Management Information

SSID	STAMP Session Identifier
SSU	Synchronization Supply Unit
STAMP	Simple Two-way Active Measurement Protocol
TLV	Type-Length-Value

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

The STAMP Session-Sender transmits test packets to the STAMP Session-Reflector. The STAMP Session-Reflector receives the 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] packets.

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.

A STAMP Session is identified by the 4-tuple (source and destination IP addresses, source and destination UDP port numbers). A STAMP Session-Sender **MAY** generate a locally unique STAMP Session Identifier (SSID). The SSID is a two-octet, non-zero unsigned integer. The SSID generation policy is implementation specific. [NUM-IDS-GEN] thoroughly analyzes common algorithms for identifier generation and their vulnerabilities. For example, an implementation can use the algorithms described in Section 7.1 of [NUM-IDS-GEN]. An implementation **MUST NOT** assign the same identifier to different STAMP test sessions. A Session-Sender **MAY** use the SSID to identify a STAMP test session. If the SSID is used, it **MUST** be present in each test packet of the given test session. In the unauthenticated mode, the 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)                       |
   |                                                               |
   |                                                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                            TLVs                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: The Format of an Extended STAMP Session-Sender Test Packet in Unauthenticated Mode

An implementation of the STAMP Session-Reflector that supports this specification **MUST** identify a STAMP Session using the SSID in combination with elements of the usual 4-tuple for the session. Before a test session commences, a Session-Reflector **MUST** be provisioned with all the elements that identify the STAMP Session. A STAMP Session-Reflector **MUST** discard non-matching STAMP test packets. The means of provisioning the STAMP Session identification is outside the scope of this specification. A conforming implementation of a 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 |                   MBZ                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            TLVs                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: The Format of an Extended STAMP Session-Reflector Test Packet 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 **MAY** stop the session if it receives a zeroed SSID field. An implementation of a Session-Sender **MUST** support control of its behavior in such a scenario. If the test session is not stopped, the Session-Sender can, for example, send a base STAMP packet [RFC8762] or continue transmitting STAMP test packets with the SSID.

The location of the SSID field in the authenticated mode is shown in Figures 3 and 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)                        |
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                            TLVs                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The Format of an Extended STAMP Session-Sender Test Packet 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)                       |
   |                                                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                            TLVs                               ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: The Format of an Extended STAMP Session-Reflector Test Packet in Authenticated Mode

4. TLV Extensions to STAMP

The Type-Length-Value (TLV) encoding scheme provides a flexible extension mechanism for optional informational elements. TLV is an optional field in the STAMP test packet. Multiple TLVs **MAY** be placed in a STAMP test packet. Additional TLVs may be enclosed within a given TLV, subject to the semantics of the (outer) TLV in question. TLVs have a one-octet STAMP TLV Flags field, a one-octet Type field, and a two-octet Length field that is equal to the length of the Value field in octets. If a Type value for a TLV or sub-TLV is in the range for Private Use [RFC8126], the length **MUST** be at least 4, and the first four octets **MUST** be that vendor's Structure of Management Information (SMI) Private Enterprise Code, as recorded in IANA's "SMI Network Management Private Enterprise Codes" subregistry, in network octet order. The rest of the Value field is private to the vendor. The following sections describe the use of TLVs for STAMP that extend the STAMP capability beyond its base specification.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|STAMP TLV Flags|     Type      |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                            Value                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: TLV Format in a STAMP Extended Packet

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. The detailed format and interpretation of flags defined in this specification are below.

Type: A one-octet field that characterizes the interpretation of the Value field. It is allocated by IANA, as specified in Section 5.1.

Length: A two-octet field equal to the length of the Value field in octets.

Value: A variable-length field. Its interpretation and encoding are determined by the value of the Type field.

All multi-byte fields in TLVs defined in this specification are in network byte order.

The format of the STAMP TLV Flags is displayed in Figure 6, and the location of flags is defined in Section 5.2.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|U|M|I|R|R|R|R|R|
+-+-+-+-+-+-+-+-+
```

Figure 6: STAMP TLV Flags Format

The fields are defined as follows:

U (Unrecognized): A one-bit flag. A Session-Sender **MUST** set the U flag to 1 before transmitting an extended STAMP test packet. A Session-Reflector **MUST** set the U flag to 1 if the Session-Reflector has not understood the TLV. Otherwise, the Session-Reflector **MUST** set the U flag in the reflected packet to 0.

M (Malformed): A one-bit flag. A Session-Sender **MUST** set the M flag to 0 before transmitting an extended STAMP test packet. A Session-Reflector **MUST** set the M flag to 1 if the Session-Reflector determined the TLV is malformed, i.e., the Length field value is not valid for the particular type, or the remaining length of the extended STAMP packet is less than the size of the TLV. Otherwise, the Session-Reflector **MUST** set the M flag in the reflected packet to 0.

I (Integrity): A one-bit flag. A Session-Sender **MUST** set the I flag to 0 before transmitting an extended STAMP test packet. A Session-Reflector **MUST** set the I flag to 1 if the STAMP extensions have failed HMAC verification (Section 4.8). Otherwise, the Session-Reflector **MUST** set the I flag in the reflected packet to 0.

R: Reserved flags for future use. These flags **MUST** be zeroed on transmit and ignored on receipt.

A STAMP node, whether Session-Sender or Session-Reflector, receiving a test packet **MUST** determine whether the packet is a base STAMP packet or whether it 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 greater than the length of the UDP header, then the test packet includes one or more STAMP TLVs that immediately follow the base STAMP test packet. A Session-Reflector that does not support STAMP extensions will not process but copy them into the reflected packet, as defined in Section 4.3 of [RFC8762]. A Session-Reflector that supports TLVs will indicate specific TLVs that it did not process by setting the U flag to 1 in those TLVs.

A STAMP Session-Sender that has received a reflected STAMP test packet with extension TLVs **MUST** validate each TLV:

- If the U flag is set, the STAMP system **MUST** skip the processing of the TLV.
- If the M flag is set, the STAMP system **MUST** stop processing the remainder of the extended STAMP packet.
- If the I flag is set, the STAMP system **MUST** discard all TLVs and **MUST** stop processing the remainder of the extended STAMP packet.

- If an implementation of a Session-Reflector does not recognize the Type field value, it **MUST** include a copy of the TLV in the reflected STAMP packet. The Session-Reflector **MUST** set the U flag to 1. The Session-Reflector **MUST** skip the processing of the unrecognized TLV.
- If a TLV is malformed, the processing of extension TLVs **MUST** be stopped. The Session-Reflector **MUST** copy the remainder of the received extended STAMP packet into the reflected STAMP packet. The Session-Reflector **MUST** set the M flag to 1.

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|STAMP TLV Flags|      Type     |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        Extra Padding                          ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Extra Padding TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 1 has been allocated by IANA (Section 5.1).

Length: A two-octet field equal to the length of the Extra Padding field in octets.

Extra Padding: This field **SHOULD** be filled by a sequence of pseudorandom numbers. The field **MAY** be filled with all zeros. An implementation **MUST** control the content of the Extra Padding field.

The Extra Padding TLV is similar to the Packet Padding field in a TWAMP-Test packet [RFC5357]. The use of the Extra Padding TLV is **RECOMMENDED** to perform a STAMP test using test packets that are larger than the base STAMP packet [RFC8762]. The length of the base STAMP packet is 44 octets in the unauthenticated mode or 112 octets in the authenticated mode. The Extra Padding TLV **MAY** be present more than one time in an extended STAMP test packet.

4.2. Location TLV

STAMP Session-Senders **MAY** include the variable-size Location TLV to query location information from the Session-Reflector. The Session-Sender **MUST NOT** fill any information fields except for the STAMP TLV Flags, Type, and Length fields. The Session-Reflector **MUST** verify that the TLV is well formed. If it is not, the Session-Reflector follows the procedure defined in Section 4 for a malformed 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |STAMP TLV Flags|      Type     |           Length              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        Destination Port       |          Source Port          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                          Sub-TLVs                             ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Location TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 2 has been allocated by IANA (Section 5.1).

Length: A two-octet field equal to the length of the Value field in octets.

Destination Port: A two-octet UDP destination port number of the received STAMP packet.

Source Port: A two-octet UDP source port number of the received STAMP packet.

Sub-TLVs: A sequence of sub-TLVs, as defined further in this section. The sub-TLVs are used by the Session-Sender to request location information with generic sub-TLV types, and the Session-Reflector responds with the corresponding more-specific sub-TLVs for the type of address (e.g., IPv4 or IPv6) used at the Session-Reflector.

Note that all fields not filled by either a Session-Sender or Session-Reflector are transmitted with all bits set to zero.

4.2.1. Location Sub-TLVs

A sub-TLV in the Location TLV uses the format displayed in Figure 5. Handling of the U and M flags in the sub-TLV is as defined in Section 4. The I flag **MUST** be set by a Session-Sender and Session-Reflector to 0 before transmission and its value ignored on receipt. The following types of sub-TLVs for the Location TLV are defined in this specification (Table 5 lists the Type values):

Source MAC Address sub-TLV: A 12-octet sub-TLV. The Type value is 1. The value of the Length field **MUST** be equal to 8. The Value field is an 8-octet MBZ field that **MUST** be zeroed on transmission and ignored on receipt.

Source EUI-48 Address sub-TLV: A 12-octet sub-TLV that includes the EUI-48 source MAC address. The Type value is 2. The value of the Length field **MUST** be equal to 8.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        EUI-48  Address                        |
+                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |            MBZ                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: The Value Field of the Source EUI-48 Address Sub-TLV

The Value field consists of the following fields (Figure 9):

EUI-48 Address: A six-octet field.

MBZ: A two-octet field. It **MUST** be zeroed on transmission and ignored on receipt.

Source EUI-64 Address sub-TLV: A 12-octet sub-TLV that includes the EUI-64 source MAC address. The Type value is 3. The value of the Length field **MUST** be equal to 8. The Value field consists of an eight-octet EUI-64 field.

Destination IP Address sub-TLV: A 20-octet sub-TLV. The Type value is 4. The value of the Length field **MUST** be equal to 16. The Value field consists of a 16-octet MBZ field that **MUST** be zeroed on transmit and ignored on receipt.

Destination IPv4 Address sub-TLV: A 20-octet sub-TLV that includes the IPv4 destination address. The Type value is 5. The value of the Length field **MUST** be equal to 16.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IPv4 Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        MBZ (12 octets)                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: IPv4 Address in a Sub-TLV's Value Field

The Value field consists of the following fields (Figure 10):

IPv4 Address: A four-octet field.

MBZ: A 12-octet field. It **MUST** be zeroed on transmit and ignored on receipt.

Destination IPv6 Address sub-TLV: A 20-octet sub-TLV that includes the IPv6 destination address. The Type value is 6. The value of the Length field **MUST** be equal to 16. The Value field is a 16-octet IPv6 Address field.

Source IP Address sub-TLV: A 20-octet sub-TLV. The Type value is 7. The value of the Length field **MUST** be equal to 16. The Value field is a 16-octet MBZ field that **MUST** be zeroed on transmit and ignored on receipt.

Source IPv4 Address sub-TLV: A 20-octet sub-TLV that includes the IPv4 source address. The Type value is 8. The value of the Length field **MUST** be equal to 16. The Value field consists of the following fields (Figure 10):

IPv4 Address: A four-octet field.

MBZ: A 12-octet field. It **MUST** be zeroed on transmit and ignored on receipt.

Source IPv6 Address sub-TLV: A 20-octet sub-TLV that includes the IPv6 source address. The Type value is 9. The value of the Length field **MUST** be equal to 16. The Value field is a 16-octet IPv6 Address field.

4.2.2. Theory of Operation of Location TLV

The Session-Reflector that received an extended STAMP packet with the Location TLV **MUST** include in the reflected packet the Location TLV with a length equal to the Location TLV length in the received packet. Based on the local policy, the Session-Reflector **MAY** leave some fields unreported by filling them with zeroes. An implementation of the stateful Session-Reflector **MUST** provide control for managing such policies.

A Session-Sender **MAY** include the Source MAC Address sub-TLV in the Location TLV. If the Session-Reflector receives the Location TLV that includes the Source MAC Address sub-TLV, it **MUST** include the Source EUI-48 Address sub-TLV if the source MAC address of the received extended test packet is in EUI-48 format. And the Session-Reflector **MUST** copy the value of the source MAC address in the EUI-48 field. Otherwise, the Session-Reflector **MUST** use the Source EUI-64 Address sub-TLV and **MUST** copy the value of the Source MAC Address from the received packet into the EUI-64 field. If the received extended STAMP test packet does not have the Source MAC Address, the Session-Reflector **MUST** zero the EUI-64 field before transmitting the reflected packet.

A Session-Sender **MAY** include the Destination IP Address sub-TLV in the Location TLV. If the Session-Reflector receives the Location TLV that includes the Destination IP Address sub-TLV, it **MUST** include the Destination IPv4 Address sub-TLV if the source IP address of the received extended test packet is of the IPv4 address family. And the Session-Reflector **MUST** copy the value of the destination IP address in the IPv4 Address field. Otherwise, the Session-Reflector **MUST** use the Destination IPv6 Address sub-TLV and **MUST** copy the value of the destination IP address from the received packet into the IPv6 Address field.

A Session-Sender **MAY** include the Source IP Address sub-TLV in the Location TLV. If the Session-Reflector receives the Location TLV that includes the Source IP Address sub-TLV, it **MUST** include the Source IPv4 Address sub-TLV if the source IP address of the received extended test packet is of the IPv4 address family. And the Session-Reflector **MUST** copy the value of the source IP address in the IPv4 Address field. Otherwise, the Session-Reflector **MUST** use the Source IPv6 Address sub-TLV and **MUST** copy the value of the source IP address from the received packet into the IPv6 Address field.

The Location TLV **MAY** be used to determine the last-hop IP addresses, ports, and last-hop MAC address for STAMP packets. The MAC address can indicate a path switch on the last hop. The IP addresses and UDP ports will indicate if there is a NAT router on the path. It allows the Session-Sender to identify the IP address of the Session-Reflector behind the NAT and detect changes in the NAT mapping that could result in sending the STAMP packets to the wrong Session-Reflector.

4.3. Timestamp Information TLV

The STAMP Session-Sender **MAY** include the Timestamp Information TLV to request information from the Session-Reflector. The Session-Sender **MUST NOT** fill any information fields except for STAMP TLV Flags, Type, and Length. All other fields **MUST** be filled with zeroes. The Session-Reflector **MUST** validate the Length value of the TLV. If the value of the Length field is invalid, the Session-Reflector follows the procedure defined in Section 4 for a malformed 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |STAMP TLV Flags|      Type     |           Length              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Sync Src In  | Timestamp In  | Sync Src Out  | Timestamp Out |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                    Optional sub-TLVs                          ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Timestamp Information TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 3 has been allocated by IANA (Section 5.1).

Length: A two-octet field, set equal to the length of the Value field in octets (Figure 5).

Sync Src In: A one-octet field that characterizes the source of clock synchronization at the ingress of a Session-Reflector. There are several methods for synchronizing the clock, e.g., the Network Time Protocol (NTP) [RFC5905]. Table 7 lists the possible values.

Timestamp In: A one-octet field that characterizes the method by which the ingress of the Session-Reflector obtained the timestamp T2. A timestamp may be obtained with hardware assistance via a software API from a local wall clock or from a remote clock (the latter is referred to as a "control plane"). Table 9 lists the possible values.

Sync Src Out: A one-octet field that characterizes the source of clock synchronization at the egress of the Session-Reflector. Table 7 lists the possible values.

Timestamp Out: A one-octet field that characterizes the method by which the egress of the Session-Reflector obtained the timestamp T3. Table 9 lists the possible values.

Optional sub-TLVs: An optional variable-length field.

4.4. Class of Service TLV

The STAMP Session-Sender **MAY** include a Class of Service (CoS) TLV in the STAMP test packet. The format of the CoS TLV is presented in Figure 12.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |STAMP TLV Flags|      Type     |           Length              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   DSCP1   |   DSCP2   |ECN| RP|          Reserved             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Class of Service TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 4 has been allocated by IANA (Section 5.1).

Length: A two-octet field, set equal to the value 4.

DSCP1: The Differentiated Services Code Point (DSCP) intended by the Session-Sender to be used as the DSCP value of the reflected test packet.

DSCP2: The received value in the DSCP field at the ingress of the Session-Reflector.

ECN: The received value in the ECN field at the ingress of the Session-Reflector.

RP (Reverse Path): A two-bit field. A Session-Sender **MUST** set the value of the RP field to 0 on transmission.

Reserved: A 16-bit field that **MUST** be zeroed on transmission and ignored on receipt.

A STAMP Session-Reflector that receives a 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 and ECN fields of the IP header of the received STAMP test packet into the DSCP2 field in the reflected test packet. Finally, the Session-Reflector **MUST** use the local policy to verify whether the CoS corresponding to the value of the DSCP1 field is permitted in the domain. If it is, the Session-Reflector **MUST** set the DSCP field's value in the IP header of the reflected test packet equal to the value of the DSCP1 field of the received test packet. Otherwise, the Session-Reflector **MUST** use the DSCP value of the received STAMP packet and set the value of the RP field to 1. Upon receiving the reflected packet, if the value of the RP field is 0, the Session-Sender will save the DSCP and ECN values for analysis of the CoS in the reverse direction. If the value of the RP field in the received reflected packet is 1, only CoS in the forward direction can be analyzed.

Re-mapping of CoS can be used to provide multiple services (e.g., 2G, 3G, LTE in mobile backhaul networks) over the same network. But if it is misconfigured, then it is often difficult to diagnose the root cause of excessive packet drops of higher-level service while packet drops for lower service packets are at a normal level. Using a CoS TLV in STAMP testing 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 the number of in-profile packets, i.e., packets that form a specific data flow, that had been transmitted and received by the Session-Sender and Session-Reflector, respectively. The definition of "in-profile packet" is outside the scope of this document and is left to the test operators to determine.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |STAMP TLV Flags|      Type     |           Length              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Session-Sender Tx counter  (S_TxC)               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Session-Reflector Rx counter  (R_RxC)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Session-Reflector Tx counter  (R_TxC)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Direct Measurement TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 5 has been allocated by IANA (Section 5.1).

Length: A two-octet field equal to the length of the Value field in octets. The Length field value **MUST** equal 12 octets.

Session-Sender Tx counter (S_TxC): A four-octet field. The Session-Sender **MUST** set its value equal to the number of the transmitted in-profile packets.

Session-Reflector Rx counter (R_RxC): A four-octet field. It **MUST** be zeroed by the Session-Sender on transmit and ignored by the Session-Reflector on receipt. The Session-Reflector **MUST** fill it with the value of in-profile packets received.

Session-Reflector Tx counter (R_TxC): A four-octet field. It **MUST** be zeroed by the Session-Sender and ignored by the Session-Reflector on receipt. The Session-Reflector **MUST** fill it with the value of the transmitted in-profile packets.

A Session-Sender **MAY** include the Direct Measurement TLV in a STAMP test packet. If the received STAMP test packet includes the Direct Measurement TLV, the Session-Reflector **MUST** include it in the reflected test packet. The Session-Reflector **MUST** copy the value from the S_TxC field of the received test packet into the same field of the reflected packet before its transmission.

4.6. Access Report TLV

A STAMP Session-Sender **MAY** include an Access Report TLV (Figure 14) 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|STAMP TLV Flags|     Type      |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ID  |  Resv |  Return Code  |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Access Report TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 6 has been allocated by IANA (Section 5.1).

Length: A two-octet field, set equal to the value 4.

ID (Access ID): A four-bit 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 the following:

1: 3GPP Network

2: Non-3GPP Network

All other values are invalid; a TLV that contains values other than '1' or '2' **MUST** be discarded.

Resv: A four-bit field that **MUST** be zeroed on transmission and ignored on receipt.

Return Code: A one-octet field that identifies the report signal, e.g., available or unavailable. The value is supplied to the STAMP endpoint through some mechanism that is outside the scope of this document. Section 5.6 lists the possible values.

Reserved: A two-octet field that **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 reception of the reflected STAMP test packet that includes the 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 and the network environment. The default value of the retransmission timer for the 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 only an "SW Local" (see Table 9) timestamp in the Follow-Up Timestamp field. But the hosting system might provide a timestamp closer to the start of the actual packet transmission even though it is not possible to deliver the information to the Session-Sender in time for the packet itself. This timestamp might nevertheless be important for the Session-Sender, as it improves the accuracy of network delay measurement 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. The Sequence Number and Follow-Up 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 the Sequence Number and Follow-Up Timestamp fields and set the M flag in the STAMP TLV Flags field in the reflected packet. If the Session-Reflector is in the stateless mode (defined in Section 4.2 of [RFC8762]), it **MUST** zero the Sequence Number and Follow-Up 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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |STAMP TLV Flags|      Type     |           Length              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      Follow-Up Timestamp                      |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Timestamp M  |                Reserved                       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Follow-Up Telemetry TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 7 has been allocated by IANA (Section 5.1).

Length: A two-octet field, set equal to the value 16 octets.

Sequence Number: A four-octet 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 of [RFC8762]), it is the Session-Reflector's Sequence Number of the previous reflected packet.

Follow-Up Timestamp: An eight-octet field, with the format indicated by the Z flag of the Error Estimate field of the STAMP base packet, which is contained in this reflected test packet transmitted by a Session-Reflector, as described in Section 4.2.1 of [RFC8762]. It carries the timestamp when the reflected packet with the specified sequence number was sent.

Timestamp M(ode): A one-octet field that characterizes the method by which the entity that transmits a reflected STAMP packet obtained the Follow-Up Timestamp. Table 9 lists the possible values.

Reserved: A three-octet 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 the 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. All authenticated STAMP base packets (per Sections 4.2.2 and 4.3.2 of [RFC8762]) compatible with this specification **MUST** additionally authenticate the optional TLVs by including the keyed Hashed Message Authentication Code (HMAC) TLV, with the sole exception of when there is only one TLV present and it is the Extended Padding TLV. The HMAC TLV **MUST** follow all TLVs included in a STAMP test packet except for the Extra Padding TLV. If the HMAC TLV appears in any other position in a

STAMP extended test packet, then the situation **MUST** be processed as HMAC verification failure, as defined below in this section. The HMAC TLV **MAY** be used to protect the integrity of STAMP extensions in the STAMP unauthenticated mode. An implementation of STAMP extensions **MUST** provide controls to enable the integrity protection of STAMP extensions in the 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |STAMP TLV Flags|      Type     |           Length              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                              HMAC                             |
     |                                                               |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: HMAC TLV

The fields are defined as follows:

STAMP TLV Flags: An eight-bit field. Its format is presented in Figure 6.

Type: A one-octet field. Value 8 has been allocated by IANA (Section 5.1).

Length: A two-octet field, set equal to the value 16 octets.

HMAC: A 16-octet field that carries the HMAC digest of the text of all preceding TLVs.

As defined in [RFC8762], STAMP uses HMAC-SHA-256 truncated to 128 bits (see [RFC4868]). All considerations regarding using the key listed in Section 4.4 of [RFC8762] are fully applicable to the use of the HMAC TLV. Key management and the mechanisms to distribute the HMAC key are outside the scope of this specification. The HMAC TLV is anticipated to track updates in the base STAMP protocol [RFC8762], including the use of more advanced cryptographic algorithms. HMAC is calculated as defined in [RFC2104] over text as the concatenation of the Sequence Number field of the base STAMP packet and all preceding TLVs. The digest then **MUST** be truncated to 128 bits and written into the HMAC field. If the HMAC TLV is present in the extended STAMP test packet, e.g., 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 the Session-Reflector **MUST** stop processing the received extended STAMP test packet. The Session-Reflector **MUST** copy the TLVs from the received STAMP test packet into the reflected packet. The Session-Reflector **MUST** set the I flag in each TLV copied over into the reflected packet to 1 before transmitting the reflected test packet. If the Session-Sender receives the extended STAMP test packet with I flag set to 1, then the Session-Sender **MUST** stop processing TLVs in the reflected test packet. If HMAC verification by the Session-Sender fails, then the Session-Sender **MUST** stop processing TLVs in the reflected extended STAMP packet.

5. IANA Considerations

IANA has created the following subregistries under the "Simple Two-way Active Measurement Protocol (STAMP) TLV Types" registry.

5.1. STAMP TLV Types Subregistry

IANA has created the "STAMP TLV Types" subregistry. The code points in this registry are allocated according to the registration procedures [RFC8126] described in Table 1.

Range	Registration Procedures
1 - 175	IETF Review
176 - 239	First Come First Served
240 - 251	Experimental Use
252 - 254	Private Use

Table 1: Registration Procedures for the STAMP TLV Types Subregistry

Per this document, IANA has allocated the following values in the "STAMP TLV Types" subregistry:

Value	Description	Reference
0	Reserved	RFC 8972
1	Extra Padding	RFC 8972
2	Location	RFC 8972
3	Timestamp Information	RFC 8972
4	Class of Service	RFC 8972
5	Direct Measurement	RFC 8972
6	Access Report	RFC 8972
7	Follow-Up Telemetry	RFC 8972
8	HMAC	RFC 8972

Value	Description	Reference
255	Reserved	RFC 8972

Table 2: STAMP TLV Types

5.2. STAMP TLV Flags Subregistry

IANA has created the "STAMP TLV Flags" subregistry. The registration procedure is "IETF Review" [RFC8126]. The flags are 8 bits. Per this document, IANA has allocated the following bit positions in the "STAMP TLV Flags" subregistry.

Bit position	Symbol	Description	Reference
0	U	Unrecognized TLV	RFC 8972
1	M	Malformed TLV	RFC 8972
2	I	Integrity check failed	RFC 8972

Table 3: STAMP TLV Flags

5.3. STAMP Sub-TLV Types Subregistry

IANA has created the "STAMP Sub-TLV Types" subregistry. The code points in this registry are allocated according to the registration procedures [RFC8126] described in Table 4.

Range	Registration Procedures
1 - 175	IETF Review
176 - 239	First Come First Served
240 - 251	Experimental Use
252 - 254	Private Use

Table 4: Registration Procedures for the STAMP Sub-TLV Types Subregistry

Per this document, IANA has allocated the following values in the "STAMP Sub-TLV Types" subregistry:

Value	Description	TLV Used	Reference
0	Reserved		RFC 8972
1	Source MAC Address	Location	RFC 8972

Value	Description	TLV Used	Reference
2	Source EUI-48 Address	Location	RFC 8972
3	Source EUI-64 Address	Location	RFC 8972
4	Destination IP Address	Location	RFC 8972
5	Destination IPv4 Address	Location	RFC 8972
6	Destination IPv6 Address	Location	RFC 8972
7	Source IP Address	Location	RFC 8972
8	Source IPv4 Address	Location	RFC 8972
9	Source IPv6 Address	Location	RFC 8972
255	Reserved		RFC 8972

Table 5: STAMP Sub-TLV Types

5.4. STAMP Synchronization Sources Subregistry

IANA has created the "STAMP Synchronization Sources" subregistry. The code points in this registry are allocated according to the registration procedures [RFC8126] described in Table 6.

Range	Registration Procedures
1 - 127	IETF Review
128 - 239	First Come First Served
240 - 249	Experimental Use
250 - 254	Private Use

Table 6: Registration Procedures for the STAMP Synchronization Sources Subregistry

Per this document, IANA has allocated the following values in the "STAMP Synchronization Sources" subregistry:

Value	Description	Reference
0	Reserved	RFC 8972
1	NTP	RFC 8972

Value	Description	Reference
2	PTP	RFC 8972
3	SSU/BITS	RFC 8972
4	GPS/GLONASS/LORAN-C/BDS/Galileo	RFC 8972
5	Local free-running	RFC 8972
255	Reserved	RFC 8972

Table 7: STAMP Synchronization Sources

5.5. STAMP Timestamping Methods Subregistry

IANA has created the "STAMP Timestamping Methods" subregistry. The code points in this registry are allocated according to the registration procedures [RFC8126] described in Table 8.

Range	Registration Procedures
1 - 127	IETF Review
128 - 239	First Come First Served
240 - 249	Experimental Use
250 - 254	Private Use

Table 8: Registration Procedures for the STAMP Timestamping Methods Subregistry

Per this document, IANA has allocated the following values in the "STAMP Timestamping Methods" subregistry:

Value	Description	Reference
0	Reserved	RFC 8972
1	HW Assist	RFC 8972
2	SW Local	RFC 8972
3	Control Plane	RFC 8972
255	Reserved	RFC 8972

Table 9: STAMP Timestamping Methods

5.6. STAMP Return Codes Subregistry

IANA has created the "STAMP Return Codes" subregistry. The code points in this registry are allocated according to the registration procedures [RFC8126] described in Table 10.

Range	Registration Procedures
1 - 127	IETF Review
128 - 239	First Come First Served
240 - 249	Experimental Use
250 - 254	Private Use

Table 10: Registration Procedures for the STAMP Return Codes Subregistry

Per this document, IANA has allocated the following values in the "STAMP Return Codes" subregistry:

Value	Description	Reference
0	Reserved	RFC 8972
1	Network available	RFC 8972
2	Network unavailable	RFC 8972
255	Reserved	RFC 8972

Table 11: STAMP Return Codes

6. Security Considerations

This document defines extensions to STAMP [RFC8762] and inherits all the security considerations applicable to the base protocol. Additionally, the HMAC TLV is defined in this document. Though the HMAC TLV protects the integrity of STAMP extensions, it does not protect against a replay attack. The use of the HMAC TLV is discussed in detail in Section 4.8.

To protect against a malformed TLV, an implementation of a Session-Sender and Session-Reflector **MUST**:

- check the setting of the M flag and
- validate the Length field value.

As this specification defines the mechanism to test DSCP mapping, this document inherits all the security considerations discussed in [RFC2474]. Monitoring and optional control of DSCP using the CoS TLV may be used across the Internet so that the Session-Sender and the Session-Reflector are located in domains that use different CoS profiles. Thus, it is essential that an operator verify the set of CoS values that is used in the Session-Reflector's domain. Also, an implementation of a Session-Reflector **SHOULD** support a local policy to confirm whether the value sent by the Session-Sender can be used as the value of the DSCP field. Section 4.4 defines the use of that local policy.

7. References

7.1. Normative References

[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>.

[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>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

7.2. Informative References

[GPS] "Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Standard", GPS SPS 5th Edition, April 2020.

[IEEE.1588.2008] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std. 1588-2008, DOI 10.1109/IEEESTD.2008.4579760, July 2008, <https://doi.org/10.1109/IEEESTD.2008.4579760>.

[NUM-IDS-GEN] Gont, F. and I. Arce, "On the Generation of Transient Numeric Identifiers", Work in Progress, Internet-Draft, draft-irtf-pearg-numeric-ids-generation-06, 13 January 2021, <https://tools.ietf.org/html/draft-irtf-pearg-numeric-ids-generation-06>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[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>.

[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>.

[TS23501] 3GPP, "Technical Specification Group Services and System Aspects; System Architecture for the 5G System (5GS); Stage 2 (Release 16)", 3GPP TS 23.501, 2019.

Acknowledgments

The authors very much appreciate the thorough review and thoughtful comments received from Tianran Zhou, Rakesh Gandhi, Yuezhong Song, and Yali Wang. The authors express their gratitude to Al Morton for his comments and valuable suggestions. The authors greatly appreciate the comments and thoughtful suggestions received from Martin Duke.

Contributors

The following individual contributed text to this document:

Guo Jun
ZTE Corporation
68# Zijinghua Road
Nanjing
Jiangsu, 210012
China
Phone: +86 18105183663
Email: guo.jun2@zte.com.cn

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
United States of America
Email: adi@apple.com

Ernesto Ruffini
OutSys
via Caracciolo, 65
20155 Milan
Italy
Email: eruffini@outsys.org