IP Performance Measurement Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

Y. Wang
T. Zhou
Huawei
H. Yang
China Mobile
C. Liu
China Unicom

February 22, 2021

Simple Two-way Active Measurement Protocol Extensions for Hop-by-Hop OAM Data Collection

draft-wang-ippm-stamp-hbh-extensions-03

Abstract

This document defines optional TLVs which are carried in Simple Two-way Active Measurement Protocol (STAMP) test packets to enhance the STAMP base functions. Such extensions to STAMP enable OAM data measurement and collection at every node and link along a STAMP test packet's delivery path without maintaining a state for each configured STAMP-Test session at every devices.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Table of Contents

1. Introduction

Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] enables the measurement of both one-way and round-trip performance metrics, such as delay, delay variation, and packet loss. In the STAMP session, the bidirectional packet flow is transmitted between STAMP Session-Sender and STAMP Session-Reflector. The STAMP Session-Reflector receives test packets transmitted from Session-Sender and acts according to the configuration. However, the performance of intermediate nodes and links that STAMP test packets traverse are invisible. In addition, the STAMP instance must be configured at every intermediate node to measure the performance per node and link

that test packets traverse, which increases the complexity of OAM in large-scale networks.

STAMP Extensions have defined several optional TLVs to enhance the STAMP base functions. These optional TLVs are defined as updates of the STAMP Optional Extensions [RFC8972]. This document extents optional TLVs to STAMP, which enables performance measurement at every intermediate node and link along a STAMP test packet's delivery path, such as measurement of delay, delay variation, packet loss, and record of link errors and route information. The following sections describe the use of TLVs for STAMP that extend STAMP capability beyond its base specification.

2. Terminology

Following are abbreviations used in this document:

STAMP: Simple Two-way Active Measurement Protocol.

IOAM: In-situ OAM.

HbH: Hop-by-Hop.

3. TLV Extensions to STAMP

3.1. IOAM-Tracing-Data TLV

STAMP Session-Sender MAY place the IOAM-Tracing-Data TLV in Session-Sender test packets to record the IOAM tracing data at every IOAM capable node along the Session-Sender test packet's forward-delivery path. As STAMP uses symmetrical packets, the Session-Sender MUST set the Length value as a multiple of 4 octets according to the number of nodes and the IOAM-Trace-Type (i.e. a 24-bit identifier which specifies which data types are used in the node data list [I-D.ietf-ippm-ioam-data]). And the node-data-copied-list fields MUST be set to zero upon Session-Sender test packets transmission and ignored upon receipt.

The IOAM-Tracing-Data TLV has the following format:

```
 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
+--------------------------------+-------------------------------+
|   IOAM-Tracing-Data Type       |            Length             |
+----------------------------------------------------------------+
|                    node-data-copied-list [0]                   |
+----------------------------------------------------------------+
|                    node-data-copied-list [1]                   |
+----------------------------------------------------------------+
~                               ...                              ~
+----------------------------------------------------------------+
|                    node-data-copied-list [n]                   |
+----------------------------------------------------------------+
```

Fig. 1 IOAM-Tracing-Data TLV Format

where fields are defined as the following:

o IOAM-Tracing-Data Type: To be assigned by IANA.

o Length: A 2-octet field that indicates the length of the value field in octets and equal to a multiple of 4 octets dependent on the number of nodes and IOAM-Trace-Type bits.

o node-data-copied-list [0..n]: A variable-length field, which record the copied content of each node data element determined by the IOAM-Trace-Type. The order of packing the data fields in each node data element follows the bit order of the IOAM-Trace-Type field (see section 4.4.1 of [I-D.ietf-ippm-ioam-data]). The last node data element in this list is the node data of the first IOAM trace capable node in the path.

In an IOAM domain, the STAMP Session-Sender and the STAMP Session-Reflector MAY be configured as the IOAM encapsulating node and the IOAM decapsulating node. The STAMP Session-Sender (i.e. the IOAM encapsulating node) generates the test packet with the IOAM Tracing Data TLV. For applying the IOAM Trace-Option functionalities to the Session-Sender test packet, the Session-Sender must inserts the "trace option header" and allocate an node-data-list array [I-D.ietf-ippm-ioam-data] into "option data" fields of Hop-by-Hop Options header in IPv6 packets [I-D.ietf-ippm-ioam-ipv6-options], and sets the corresponding bits in the IOAM-Trace-Type. Also, the STAMP Session-Sender allocates a node-data-copied-list array in the optional IOAM-Tracing-Data TLV to store OAM data retrieved from every IOAM transit node along the Session-Sender test packet's delivery path.

When the STAMP Session-Reflector (i.e. the IOAM decapsulating node) received the STAMP Session-Sender test packet with the IOAM-Tracing-Data TLV, it MUST copy the node-data-list array into the node-data-copied-list array carried in the Session-Reflector test packet before transmission and MUST remove the IOAM-Data-Fields. Hence, carrying IOAM-Tracing-Data TLV in STAMP test packets enables OAM data collection and measurement at every node and link.

Also the STAMP Session-Reflector MAY be configured as IOAM encapsulating node to apply the IOAM Trace-Option functionalities to the Session-Reflector test packet. Hence, OAM data collection and measurement can be also enabled at every node and link along the Session-Reflector test packet's backward delivery path. When the reflected packet arrives at the Session-Sender, it can be either locally processed or sent to the centralized controller.

3.2. Forward HbH Delay TLV

STAMP Session-Sender MAY place the Forward HbH Delay TLV in Session-Sender test packets to record the ingress timestamp and the egress timestamp at every intermediate nodes along the Session-Sender test packet's forward path. The Session-Sender MUST set the Length value according to the number of explicitly listed intermediate nodes in the forward path and the timestamp formats. There are several methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905] and IEEE 1588v2 Precision Time Protocol (PTP) [IEEE.1588.2008]. For example, if a 64-bit timestamp format defined in NTP is used, the Length value MUST be set as a multiple of 16 octets. The Timestamp Tuple list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission.

The Forward HbH Delay TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|   Forward HbH Delay Type      |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                                                               |
|                     Timestamp Tuple list [1]                  |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
~                               ...                             ~
+---------------------------------------------------------------+
|                                                               |
|                     Timestamp Tuple list [n]                  |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
```

Fig. 2 Forward HbH Delay TLV Format

where fields are defined as the following:

- o Forward HbH Delay Type: To be assigned by IANA.

- o Length: A 8-bit field that indicates the length of the value portion in octets and MUST be a multiple of 16 octets according to the number of explicitly listed intermediate nodes in the forward path.

- o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the forward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.

- o Timestamp Tuple list [1..n]: A variable-length field, which record the timestamp when the Session-Sender test packet is received at the ingress of the n-th intermediate node and the timestamp when the Session-Sender test packet is sent at egress of the n-th intermediate node. For example, if a 64-bit timestamp format defined in NTP is used, the length of each Timestamp Tuple (ingress timestamp [n], egress timestamp [n]) must be 16 octets. The Timestamp Tuple list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the Timestamp Tuple list [1] records the timestamps when the Session-Sender test packet received and forwarded at the last intermediate node of a explicit path, the second element records the penultimate Timestamp Tuple when the Session-Sender

test packet received and forwarded at the penultimate intermediate node of a explicit path, and so on.

In the following reference topology, Node N1, N2, N3, N4 and N5 are SRv6 capable nodes. Node N1 is the STAMP Session-Sender and Node N5 is the STAMP Session-Reflector. T1 is the Timestamp taken by the Session-Sender (i.e. N1) at the start of transmitting the test packet. T2 is the Receive Timestamp when the test packet was received by the Session-Reflector (i.e. N5). T3 is the Timestamp taken by the Session-Reflector at the start of transmitting the test packet. T4 is the Receive Timestamp when the test packet was received by the Session-Sender. Timestamp tuples (t1,t2), (t3,t4) and (t5,t6) are the timestamps when the test packet received and transmitted by sequence of intermediate nodes in the forward path. Timestamp Tuples (t7,t8), (t9,t10) and (t11,t12) are the timestamps when the test packet received and transmitted by sequence of intermediate nodes in the backward path.

```
======          ======          ======          ======         ======
|    | T1--->t1 |    | t2--->t3 |    | t4--->t5 |    | t6--->T2|    |
| N1 |==========| N2 |==========| N3 |==========| N4 |=========| N5 |
|    | T4<---t12|    |t11<---t10|    | t9<---t8 |    | t7<---T3|    |
======          ======          ======          ======         ======
```

Fig. 3 Reference Topology

The STAMP Session-Sender (i.e. Node N1) generates the STAMP test packet with the Forward HbH Delay TLV. When an intermediate node receives the STAMP test packet, the node punts the packet to control plane and fills the ingress timestamp [n] filed in the Timestamp Tuple list [n]. Then the time taken by the intermediate node transmitting the test packet is recorded in to egress timestamp [n] field. The mechanism of timestamping and punting packet to control plane is outside the scope of this specification.

When the STAMP Session-Reflector received the test packet with the Forward HbH Delay TLV, it MUST copy the Forward HbH Delay TLV into the Session-Reflector test packet before its transmission. Using Forward HbH Delay TLV in STAMP testing enables delay measurement per link in the forward path.

3.3. Backward HbH Delay TLV

STAMP Session-Sender MAY place the Backward HbH Delay TLV in Session-Sender test packets to record the ingress timestamp and egress timestamp when Session-Reflector test packets are received and sent at every intermediate nodes in the backward path. The Session-Sender

MUST set the Length value according to the number of explicitly listed intermediate nodes in the backward path and the timestamp formats. There are several methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905] and IEEE 1588v2 Precision Time Protocol (PTP) [IEEE.1588.2008]. For example, if a 64-bit timestamp format defined in NTP is used, the Length value MUST be set as a multiple of 16 octets. The Timestamp Tuple list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission and ignored upon receipt.

The Backward HbH Delay TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|   Backward HbH Delay Type     |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                                                               |
|                      Timestamp Tuple list [1]                 |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
~                              ...                              ~
+---------------------------------------------------------------+
|                                                               |
|                      Timestamp Tuple list [n]                 |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
```

Fig. 4 Backward HbH Delay TLV Format

where fields are defined as the following:

o Backward HbH Delay Type: To be assigned by IANA.

o Length: A 8-bit field that indicates the length of the value portion in octets and will be a multiple of 16 octets dependent on the number of explicitly listed intermediate nodes in the backward path.

o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the backward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.

o Timestamp Tuple list [1..n]: A variable-length field, which record the timestamp when the reflected test packet is received at the ingress of the n-th intermediate node and the timestamp when the reflected test packet is sent at egress of the n-th intermediate node. For example, if a 64-bit timestamp format defined in NTP is used, the length of each Timestamp tuple (ingress timestamp [n], egress timestamp [n]) must be 16 octets. The Timestamp Tuple list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the Timestamp Tuple list [1] records the timestamps when the reflected test packet received and forwarded at the last intermediate node of a explicit path, the second element records the penultimate Timestamp Tuple when the reflected test packet received and forwarded at the penultimate intermediate node of a explicit path, and so on.

When the STAMP Session-Reflector received the Session-Sender test packet with the Backward HbH Delay TLV, it MUST copy the Backward HbH Delay TLV into the Session-Reflector test packet.

When an intermediate node receives the reflected test packet, the node sends the packet to control plane and fills the ingress timestamp [n] filed of Backward HbH Delay TLV. Then the time taken by the intermediate node transmitting the test packet is recorded in to egress timestamp [n] field of Backward HbH Delay TLV. Using Backward HbH Delay TLV in STAMP testing enables delay measurement per link in the backward path.

3.4. HbH Direct Loss TLV

STAMP Session-Sender MAY place the HbH Direct Loss TLV in Session-Sender test packets to record the number of packets that form a specific data flow received at and transmitted by every intermediate nodes along the forward path. The Session-Sender MUST set the Length value according to the number of explicitly listed intermediate nodes in the forward path. A Counter Tuple is composed of a 64-bit Receive Counter field and a 64-bit Transmit Counter field. The Counter Tuple list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission.

The HbH Direct Loss TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|     HbH Direct Loss Type      |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                                                               |
|                     Counter Tuple list [1]                    |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
~                              ...                              ~
+---------------------------------------------------------------+
|                                                               |
|                     Counter Tuple list [n]                    |
|                                                               |
|                                                               |
+---------------------------------------------------------------+
```

Fig. 5 HbH Direct Loss TLV Format

where fields are defined as the following:

- o HbH Direct Loss Type: To be assigned by IANA.
- o Length: A 8-bit field that indicates the length of the value portion in octets and will be a multiple of 16 octets dependent on the number of explicitly listed intermediate nodes in the forward path.
- o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the forward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.
- o Counter Tuple list [1..n]: A variable-length field, which record the Receive Counter and the Transmit Counter when the data packet is received at and transmitted by the n-th intermediate node. The Counter Tuple list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the Counter Tuple list [1] records the Receive Counter and the Transmit Counter when the data packet is received at and transmitted by the last intermediate node of a explicit path, the second element records the penultimate Counter Tuple when the data packet received and forwarded at the penultimate intermediate node of a explicit path, and so on.

The STAMP Session-Sender generates the STAMP test packet with the HbH Direct Loss TLV. When an intermediate node receives the STAMP test packet, the node punts the packet to control plane and writes the Receive Counter [n] and the Transmit Counter [n] at the Counter Tuple list [n] in the Session-Sender test packet. The mechanism of punting packet to control plane is outside the scope of this specification.

When the STAMP Session-Reflector received the test packet with the HbH Direct Loss TLV, it MUST copy the HbH Direct Loss TLV into the Session-Reflector test packet before its transmission. Using HbH Direct Loss TLV in STAMP testing enables packet loss measurement per node/link in the forward path.

3.5. HbH Bandwidth Utilization TLV

STAMP Session-Sender MAY place the HbH Bandwidth Utilization (BW Utilization) TLV in Session-Sender test packets to record the ingress and egress BW Utilization at every intermediate nodes along the forward path. The Session-Sender MUST set the Length value according to the number of explicitly listed intermediate nodes in the forward path. A BW Utilization Tuple is composed of a 32-bit ingress BW Utilization field and a 32-bit egress BW Utilization field. The BW Utilization Tuple list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission.

The HbH Bandwidth Utilization TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|    HbH BW Utilization Type    |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                BW Utilization Tuple list [1]                  |
|                                                               |
+---------------------------------------------------------------+
~                              ...                              ~
+---------------------------------------------------------------+
|                BW Utilization Tuple list [n]                  |
|                                                               |
+---------------------------------------------------------------+
```

Fig. 6 HbH Bandwidth Utilization TLV Format

where fields are defined as the following:

o HbH BW Utilization Type: To be assigned by IANA.

o Length: A 8-bit field that indicates the length of the value portion in octets and will be a multiple of 8 octets dependent on the number of explicitly listed intermediate nodes in the forward path.

o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the forward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.

o BW Utilization Tuple list [1..n]: A variable-length field, which record the ingress and egress bandwidth utilization when the test packet is received at and transmitted by the n-th intermediate node. The BW Utilization Tuple list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the BW Utilization Tuple list [1] records the ingress and the egress bandwidth utilization when the test packet is received at and transmitted by the last intermediate node of a explicit path, the second element records the penultimate BW Utilization Tuple when the test packet received at and transmitted by the penultimate intermediate node of a explicit path, and so on.

The STAMP Session-Sender generates the STAMP test packet with the HbH BW Utilization TLV. When an intermediate node receives the STAMP test packet, the node punts the packet to control plane and writes the ingress and egress bandwidth utilization at the BW Utilization Tuple list [n] in the Session-Sender test packet. The mechanism of punting packet to control plane is outside the scope of this specification.

When the STAMP Session-Reflector received the test packet with the HbH BW Utilization TLV, it MUST copy the HbH BW Utilization TLV into the Session-Reflector test packet before its transmission. The HbH BW Utilization TLV carried in STAMP test packet is usable to detect and troubleshoot the link congestion in the forward path.

3.6. HbH Timestamp Information TLV

STAMP Session-Sender MAY place the HbH Timestamp Information TLV in Session-Sender test packets to record the ingress and egress Timestamp Information at every intermediate nodes along the forward path. The Timestamp Information includes the source of clock synchronization and the method of timestamp obtainment. There are several types of clock synchronization source, e.g., NTP, PTP. The method of timestamp obtainment may be from control plane (e.g. NTP) or from data plane (e.g. PTP).

A Timestamp Info Tuple is composed of a 8-bit ingress clock source field, a 8-bit ingress timestamp obtainment field, a 8-bit egress clock source field, and a 8-bit egress timestamp obtainment field. The Session-Sender MUST set the Length value according to the number of explicitly listed intermediate nodes in the forward path. The Timestamp Info Tuple list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission.

The HbH Timestamp Information TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|    HbH Timestamp Info Type    |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                Timestamp Info Tuple list [1]                  |
+---------------------------------------------------------------+
~                             ...                               ~
+---------------------------------------------------------------+
|                Timestamp Info Tuple list [n]                  |
+---------------------------------------------------------------+
```

Fig. 6 HbH Timestamp Information TLV Format

where fields are defined as the following:

o HbH Timestamp Info Type: To be assigned by IANA.

o Length: A 8-bit field that indicates the length of the value portion in octets and will be a multiple of 4 octets dependent on the number of explicitly listed intermediate nodes in the forward path.

o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the forward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.

o Timestamp Info Tuple list [1..n]: A variable-length field, which record the source of clock synchronization and the method of timestamp obtainment at the ingress and egress when the test packet is received at and transmitted by the n-th intermediate node. The Timestamp Info Tuple list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the Timestamp Info Tuple list [1] records the source of clock synchronization and the method of timestamp

obtainment at the ingress and egress when the test packet is received at and transmitted by the last intermediate node of a explicit path, the second element records the penultimate Timestamp Info Tuple when the test packet received at and transmitted by the penultimate intermediate node of a explicit path, and so on.

The STAMP Session-Sender generates the STAMP test packet with the HbH Timestamp Information TLV. When an intermediate node receives the STAMP test packet, the node punts the packet to control plane and writes the source of clock synchronization and the method of timestamp obtainment at the Timestamp Info Tuple list [n] in the Session-Sender test packet. The mechanism of punting packet to control plane is outside the scope of this specification.

When the STAMP Session-Reflector received the test packet with the HbH Timestamp Information TLV, it MUST copy the HbH Timestamp Information TLV into the Session-Reflector test packet before its transmission. The HbH Timestamp Information TLV carried in STAMP test packet is usable to query timestamp information from every nodes in the forward path.

Note that the source of clock synchronization, NTP or PTP, is part of configuration of the Session-Sender/Reflector or a particular test session [RFC8762]. This draft recommends every intermediate nodes to be configured to use the same source of clock synchronization.

3.7. HbH Interface Errors TLV

STAMP Session-Sender MAY place the HbH Interface Errors TLV in Session-Sender test packets to record the errors detected on the interface of every intermediate node used to receive the packet along the forward path. The record of interface errors indicates the quality of the interfaces along the forward path and is helpful to analyze the performance degrades associated with the flow.

A Interface Errors is a 32 bits unsigned integer field. This field records the Bit Error Rate (BER) or number of packet drop due to Cyclic Redundancy Check (CRC) errors. The Session-Sender MUST set the Length value according to the number of explicitly listed intermediate nodes in the forward path. The Interface Errors list [1..n] fields MUST be set to zero upon Session-Sender test packets transmission.

The HbH Timestamp Information TLV has the following format:

```
 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
+-------------------------------+---------------+---------------+
|   HbH Interface Errors Type   |     Length    |    Node Left  |
+-------------------------------+---------------+---------------+
|                   Interface Errors list [1]                   |
+---------------------------------------------------------------+
~                              ...                              ~
+---------------------------------------------------------------+
|                   Interface Errors list [n]                   |
+---------------------------------------------------------------+
```

Fig. 6 HbH Timestamp Information TLV Format

where fields are defined as the following:

- o HbH Interface Errors Type: To be assigned by IANA.

- o Length: A 8-bit field that indicates the length of the value portion in octets and will be a multiple of 4 octets dependent on the number of explicitly listed intermediate nodes in the forward path.

- o Node Left: A 8-bit unsigned integer, which indicates the number of intermediate nodes remaining. That is, number of explicitly listed intermediate nodes still to be visited before reaching the destination node in the forward path. The Node Left field is set to n-1, where n is the number of intermediate nodes.

- o Interface Errors list [1..n]: A variable-length field, which record the errors detected on the interface of the n-th intermediate node used to receive the packet along the forward path. The Interface Errors list is encoded starting from the last intermediate node which is explicitly listed. That is, the first element of the Interface Errors list [1] records the interface errors when the test packet is received at the last intermediate node of a explicit path, the second element records the penultimate interface errors when the test packet received at the penultimate intermediate node of a explicit path, and so on.

The STAMP Session-Sender generates the STAMP test packet with the HbH Interface Errors TLV. When an intermediate node receives the STAMP test packet, the node punts the packet to control plane and writes the errors at the Interface Errors list [n] in the Session-Sender test packet. The mechanism of punting packet to control plane is outside the scope of this specification.

When the STAMP Session-Reflector received the test packet with the HbH Interface Errors TLV, it MUST copy the HbH Interface Errors TLV into the Session-Reflector test packet before its transmission. The HbH Interface Errors TLV carried in STAMP test packet is usable to detect interface errors from every intermediate nodes along the forward path.

4. IANA Considerations

IANA is requested to allocate values for the following TLV Type from the "STAMP TLV Type" registry [RFC8972].

Code Point	Description	Reference
TBA1	IOAM Tracing Data TLV	This document
TBA2	Forward HbH Delay TLV	This document
TBA3	Backward HbH Delay TLV	This document
TBA4	HbH Direct Loss TLV	This document
TBA5	HbH BW Utilization TLV	This document
TBA6	HbH Timestamp Information TLV	This document
TBA7	HbH Interface Errors TLV	This document

5. Security Considerations

This document extensions new optional TLVs to STAMP. It does not introduce any new security risks to STAMP.

6. Acknowledgements

The authors would like to thank Giuseppe Fioccola for the reviews and comments.

7. References

7.1. Normative References

[I-D.ietf-ippm-ioam-data]
"Data Fields for In-situ OAM", <https://datatracker.ietf.org/doc/draft-ietf-ippm-ioam-data/>.

[I-D.ietf-ippm-ioam-ipv6-options]
"In-situ OAM IPv6 Options", <https://datatracker.ietf.org/doc/draft-ietf-ippm-ioam-ipv6-options/>.

[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>.

[RFC8762] "Simple Two-Way Active Measurement Protocol", <https://datatracker.ietf.org/doc/rfc8762/>.

[RFC8972] "Simple Two-way Active Measurement Protocol Optional Extensions", <https://datatracker.ietf.org/doc/rfc8972/>.

7.2. Informative References

[IEEE.1588.2008] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", <https://ieeexplore.ieee.org/document/4579760>.

[RFC5905] "Network Time Protocol Version 4: Protocol and Algorithms Specification", <https://www.rfc-editor.org/info/rfc5905>.

Authors' Addresses

Yali Wang
Huawei
156 Beijing Rd., Haidian District
Beijing
China

Email: wangyali11@huawei.com

Tianran Zhou
Huawei
156 Beijing Rd., Haidian District
Beijing
China

Email: zhoutianran@huawei.com

Hongwei Yang
China Mobile
Xibianmen Inner St, 53, Xicheng District
Beijing
China

Email: yanghongwei@chinamobile.com

Chang Liu
China Unicom
Beijing
China

Email: liuc131@chinaunicom.cn