IPPM
Internet-Draft
Intended status: Experimental
Expires: January 4, 2021

M. Cociglio
Telecom Italia
G. Fioccola
Huawei Technologies
M. Nilo
F. Bulgarella
Telecom Italia
R. Sisto
Politecnico di Torino

July 3, 2020

Client-Server Explicit Performance Measurements
draft-cfb-ippm-spinbit-measurements-02

Abstract

This document introduces an additional single bit signal to enhance the spin bit [I-D.trammell-ippm-spin] performance in presence of network impairments and application limited flow. In addition, it defines two new explicit per-flow transport-layer signals for hybrid measurement of connection loss rate. The former is a spin-bit dependent signal and uses a single bit. The latter is a standalone solution based on a two bits loss signal and on alternate marking RFC 8321 [RFC8321].

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2021.

Table of Contents

1. Introduction

Both [I-D.trammell-tsvwg-spin] and [I-D.trammell-ippm-spin] define an explicit per-flow transport-layer signal for hybrid measurement of end-to-end RTT. This signal consists of three bits: a spin bit, which oscillates once per end-to-end RTT, and a two-bit Valid Edge Counter (VEC), which compensates for loss and reordering of the spin bit to increase fidelity of the signal in less than ideal network conditions.

In this document it is introduced the delay bit, that is a single bit signal that can be used together with the spin bit by passive observers to measure the RTT of a network flow, avoiding the spin bit ambiguities that arise as soon as network conditions deteriorate. Unlike the spin bit, which is actually set in every packet transmitted on the network, the delay bit is set only once per round trip.

Regarding loss rate measurement, two new algorithms are introduced. The first algorithm enables end-to-end round trip loss rate measurement using a single bit signal called loss bit. This signal is used to mark a train of packets (a portion of traffic) which bounces back an forth two times between endpoints, realizing a two round trip reflection. A passive on-path observer, placed on whatever direction, can trivially count and compare the number of marked packets seen during the two reflections estimating statistically the loss rate experienced by the connection. The second algorithm uses a double square signal and RFC 8321 [RFC8321] to mark the whole traffic exchanged between endpoints. This solution enables different types of measurements providing a complete picture of connection loss events.

This document defines hybrid measurement RFC 7799 [RFC7799] path signals to be embedded into a transport layer protocol, explicitly intended for exposing end-to-end RTT and loss rate information to measurement devices on path.

The document introduces mechanisms applicable to any transport-layer protocol, then explains how to bind the signals to a variety of IETF transport protocols, and in particular to QUIC and TCP.

The application of the spin bit to QUIC is described in [I-D.ietf-quic-spin-exp] which adds the spin bit to QUIC for experimentation purposes.

Note that spin bit, delay bit and loss bits explained in this document are inspired by RFC 8321 [RFC8321]. This is also mentioned in [I-D.trammell-quic-spin].

Note that additional details about the Performance Measurements for QUIC are also described in the paper [ANRW19-PM-QUIC].

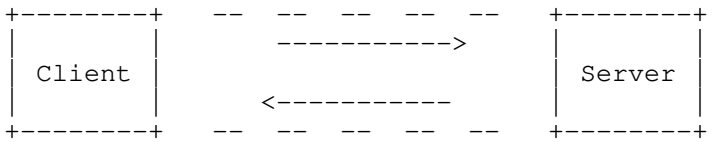
2. Spin bit and Delay bit mechanism

The main idea is to have a single packet, with a second marked bit (the delay bit), that bounces between client and server during the entire connection life. This single packet is called Delay Sample.

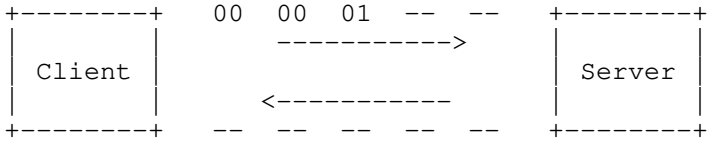
A simple observer placed in an intermediate point, tracking the delay sample and the relative timestamp in every spin bit period, can measure the end-to-end round trip delay of the connection. In the same way as seen with the spin bit, it is possible to carry out other types of measurements using this additional bit. The next paragraphs give an overview of the observer capabilities.

In order to describe the delay sample working mechanism in detail, we have to distinguish two different phases which take part in the delay bit lifetime: initialization and reflection. The initialization is the generation of the delay sample, while the reflection realizes the bounce behavior of this single packet between the two endpoints.

The next figure describes the Delay bit mechanism: the first bit is the spin bit and the second one is the delay bit.



(a) No traffic at beginning.



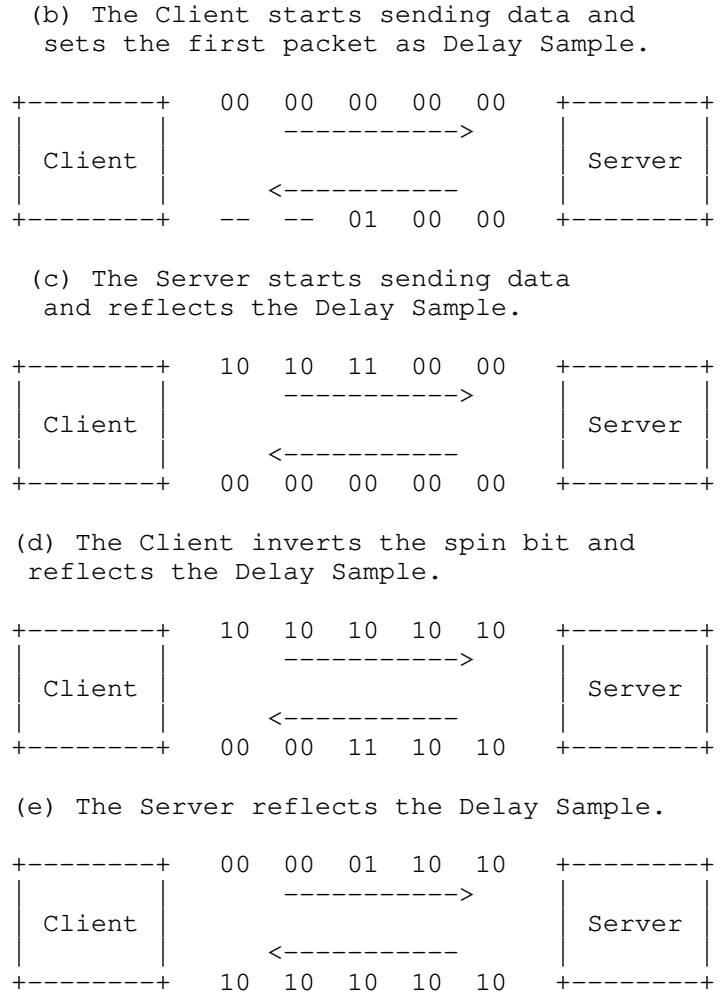


Figure 1: Spin bit and Delay bit

2.1. Delay Sample generation

During this first phase, endpoints play different roles. First of all a single delay sample must be bouncing per round trip period (and so per spin bit period). According to that statement and in order to simplify the general algorithm, the delay sample generation is in charge of just one of the two endpoints:

- the client, when connection starts and spin bit is set to 0, initializes the delay bit of the first packet to 1, so it becomes the delay sample for that marking period. Only this packet is marked with the delay bit set to 1 for this round trip period; the other ones will carry only the spin bit;

- the server never initializes the delay bit to 1; its only task is to reflect the incoming delay bit into the next outgoing packet only if certain conditions occur.

Theoretically, in absence of network impairments, the delay sample should bounce between client and server continuously, for the entire duration of the connection. Actually, that is highly unlikely mainly for two different reasons:

1) the packet carrying the delay bit might be lost during its journey on the network which is unreliable by definition;

2) one of the two endpoints could stop or delay sending data because the application is limiting the amount of traffic transmitted;

To deal with these problems, the algorithm provides a procedure to regenerate the delay sample and to inform a possible observer that a problem has occurred, and then the measurement has to be restarted.

2.1.1. The recovery process

In order to relieve the server from tasks that go beyond the mere reflection of the sample, even in this case the recovery process belongs to the client. A fundamental assumption is that a delay sample is strictly related to its spin bit period. Considering this rule, the client verifies that every spin bit period ends with its delay sample. If that does not happen and a marking period terminates without a delay sample, the client waits a further empty period; then, in the following period, it reinitializes the mechanism by setting the delay bit of the first outgoing packet to 1, making it the new delay sample. The empty period is needed to inform the intermediate points that there was an issue and a new delay measurement session is starting.

2.2. Delay Sample reflection

The reflection is the process that enables the bouncing of the delay sample between client and server. The behavior of the two endpoints is slightly different. With the exception of the client that, as previously exposed, generates a new delay sample, by default the delay bit is set to 0.

Server side reflection: when a packet with the delay bit set to 1 arrives, the server marks the first packet in the opposite direction as the delay sample, if it has the same spin bit value. While if it has the opposite spin bit value this sample is considered lost.

Client side reflection: when a packet with delay bit set to 1 arrives, the client marks the first packet in the opposite direction as the delay sample, if it has the opposite spin bit value. While if it has the same spin bit value this sample is considered lost.

In both cases, if the outgoing marked packet is transmitted with a delay greater than a predetermined threshold after the reception of the incoming delay sample (1ms by default), reflection is aborted and this sample is considered lost.

Note that reflection takes place for the packet that is carrying the delay bit regardless of its position within the period. For this reason it is necessary to introduce that condition of validation in order to identify and discard those samples that, due to reordering, might move to a contiguous period. Furthermore, by introducing a threshold for the retransmission delay of the sample, it is possible to eliminate all those measurements which, due to lack of traffic on the endpoints, would be overestimated and not true. Thus, the maximum estimation error, without considering any other delays due to flow control, would amount to twice the threshold (e.g. 2ms) per measurement, in the worst case.

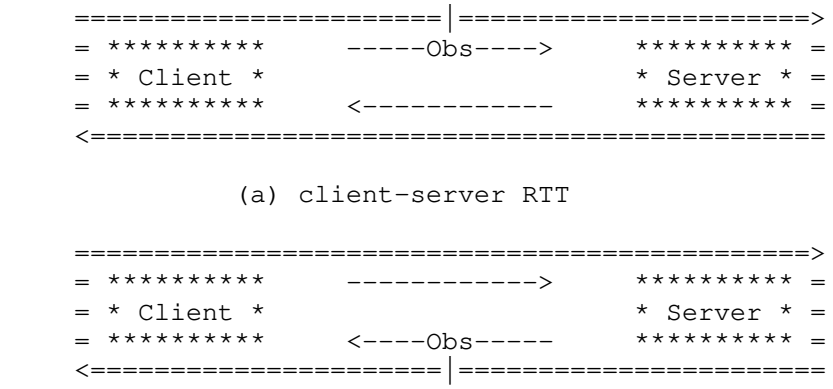
3. Using the Spin bit and Delay bit for Hybrid RTT Measurement

Unlike what happens with the spin bit for which it is necessary to validate or at least heuristically evaluate the goodness of an edge, the delay sample can be used by an intermediate observer as a simple demarcator between a period and the following one eliminating the ambiguities on the calculation of the RTT found with the analysis of the spin-bit only. The measurement types, that can be done from the observation of the delay sample, are exactly the same achievable with the spin bit only.

3.1. End-to-end RTT measurement

The delay sample generation process ensures that only one packet marked with the delay bit set to 1 runs back and forth on the wire between two endpoints per round trip time. Therefore, in order to determine the end-to-end RTT measurement of a QUIC flow, an on-path passive observer can simply compute the time difference between two delay samples observed in a single direction. Note that a measurement, to be valid, must take into account the difference in time between the timestamps of two consecutive delay samples

belonging to adjacent spin-bit periods. For this reason, an observer, in addition to intercepting and analyzing the packets containing the delay bit set to 1, must maintain awareness of each spin period in such a way as to be able to assign each delay sample to its period and, at the same time, identifying those periods that do not contain it.



(a) client-server RTT

(b) server-client RTT

Figure 2: Round-trip time (both direction)

3.2. Half-RTT measurement

An on-path passive observer that is sniffing traffic in both directions -- from client to server and from server to client -- can also use the delay sample to measure "upstream" and "downstream" RTT components. Also known as the half-RTT measurement, it represents the components of the end-to-end RTT concerning the paths between the client and the observer (upstream), and the observer and the server (downstream). It does this by measuring the delay between a delay sample observed in the downstream direction and the one observed in the upstream direction, and vice versa. Also in this case, it should verify that the two delay samples belong to two adjacent periods, for the upstream component, or to the same period for the downstream component.

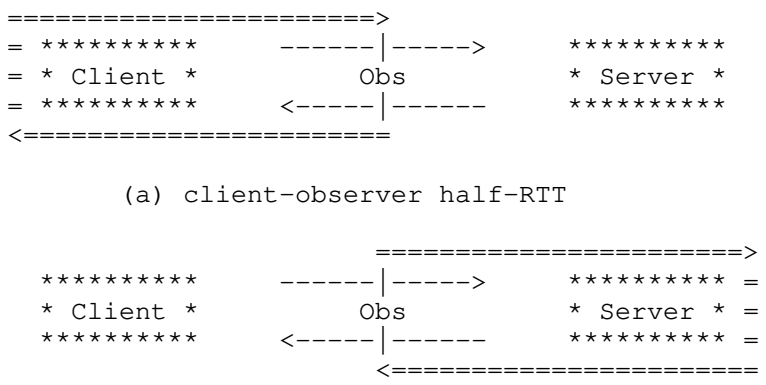


Figure 3: Half Round-trip time (both direction)

3.3. Intra-domain RTT measurement

Taking advantage of the half-RTT measurements it is also possible to calculate the intra-domain RTT which is the portion of the entire RTT used by a QUIC flow to traverse the network of a provider (or part of it). To achieve this result two observers, able to watch traffic in both directions, must be employed simultaneously at ingress and egress of the network to be measured. At this point, to determine the delay between the two observers, it is enough to subtract the two computed upstream (or downstream) RTT components.

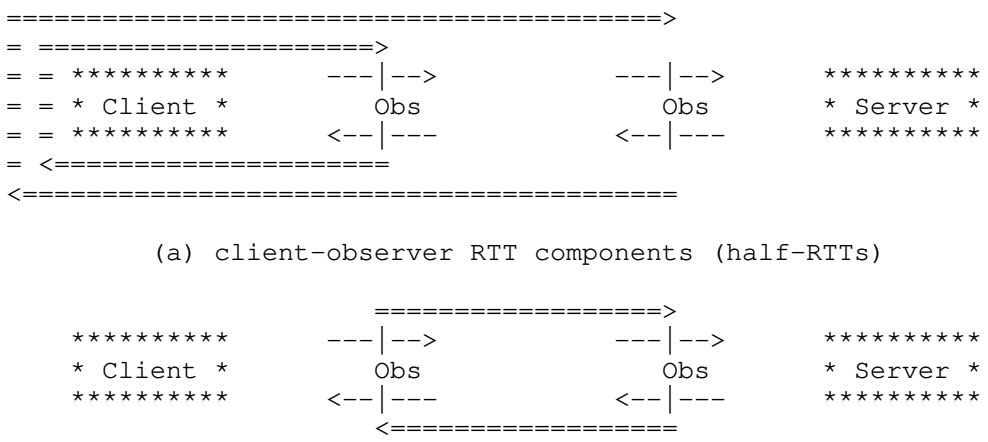


Figure 4: Intra-domain Round-trip time (client-observer: upstream)

The spin bit is an alternate marking generated signal and the only difference than RFC 8321 [RFC8321] is the size of the alternation that will change with the flight size each RTT. So it can be useful to segment the RTT and deduce the contribution to the RTT of the portion of the network between two on-path observers and it can be easily performed by calculating the delay between two or more measurement points on a single direction by applying RFC 8321 [RFC8321].

4. Observer's algorithm and Waiting Interval

Given below is a formal summary of the functioning of the observer every time a delay sample is detected. A packet containing the delay bit set to 1:

- if it has the same spin bit value of the current period and no delay sample was detected in the previous period, then it can be used as a left edge (i.e. to start measuring an RTT sample), but not as a right edge (i.e. to complete and RTT measurement since the last edge). If the observation point is symmetric (i.e. it can see both upstream and downstream packets in the flow) and in the current period a delay sample was detected in the opposite direction (i.e. in the upstream direction), the packet can also be used to compute the downstream RTT component.

- o if it has the same spin bit value of the current period and a delay sample was detected in the previous period, then it can be used at the same time as a left or right edge, and to compute RTT component in both directions.

Like stated previously, every time an empty period is detected, the observer must restart the measurement process and consider the next delay sample that will come as the beginning of a new measure, then as a left edge. As a result, being able to assign the delay sample to the corresponding spin period becomes a crucial factor for the proper functioning of the entire algorithm.

Considering that the division into periods is realized by exploiting the spin bit square wave, it is easy to understand that the presence of spurious spin edges -- caused by packet reordering -- would inevitably lead the observer to overestimate the amount of periods actually present in the transmission. This results in a greater number of empty periods detected and the consequent decrease of the actual RTT samples achievable. Therefore, in order to maximize the performance of the whole algorithm, the observer must implement a mechanism to filter out spurious spin edges.

To face this problem the waiting interval has to be introduced. Basically, every time a spin bit edge is detected, the observer sets a time interval during which it rejects every potential spurious edges observed on the wire. While, at the end of the interval it starts again to accept changes in the spin bit value. This guarantees a proper protection against the spurious edges in relation to the size of the interval itself. For instance, an interval of 5ms is able to filter out edges that have been reordered by a maximum of 5ms. Clearly, the mechanism does its job for intervals smaller than the RTT of the observed connection (if RTT is smaller than the waiting interval the observer can’t measure the RTT).

5. Adding a Loss signal for Packet loss measurement

It is possible to introduce a mechanism to evaluate also the packet loss together with the delay measurement. This can be achieved by introducing the loss signal, a single bit signal whose purpose is to mark a variable number of packets (from live traffic) which are exchanged two times between the endpoints realizing a two round-trip reflection. The overall exchange comprises:

- o The client first selects, generates and consequent transmits to the server a first train of packets, by marking the loss bit to 1;

- o The server, upon reception from the client of each one of the packets included in the first train, reflects to the client a

respective second train of packets of the same size as the first train received, by marking the loss bit to 1;

- The client, upon reception from the server of each one of the packets included in the second train, reflects to the server a respective third train of packets of the same size as the second train received, by marking the loss bit to 1;

- The server, upon reception from the client of each one of the packets included in the third train, finally reflects to the client a respective fourth train of packets of the same size as the third train received, by marking the loss bit to 1.

Packets belonging to the first round (first and second train) represent the Generation Phase while those belonging to the second round (third and fourth train) represent the Reflection Phase.

A passive on-path observer, placed on whatever direction, can trivially count and compare the number of marked packets seen during the two mentioned phases (i.e. the first and third or the second and the fourth trains of packets, depending on which direction is observed) and estimate the loss rate experienced by the connection. This process is repeated continuously to obtain more measurements as long as the endpoints exchange traffic. These measurements can be called Round Trip(RT) losses

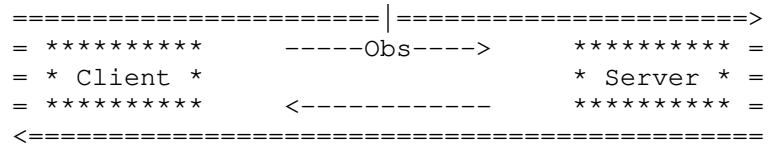
The general algorithm shown above gives an idea of its underlying principles but is not enough to make the whole process working properly.

Firstly, there is the issue that packet rates in the two directions may be different. Therefore, the right number of packets to be marked has to be chosen in order to avoid their congestion on the slowest traffic direction. As a consequence, this number is inevitably equal to the amount of packets transited, indeed, on the slowest direction. This problem can be easily addressed by a method wherein the two endpoints of a communication exchange marked packets interleaved with unmarked packets. From an implementation point of view, this result can be achieved by introducing a single token system that adjusts the number of outgoing marked packets. Basically, the token is enabled every time a packet arrives and disabled when a marked packet is transmitted. Since the creation of the initial train of marked packets is carried out by the client, the management and use of this single token is also assigned to it, which in fact "calculates" the correct number of packets to be marked each time.

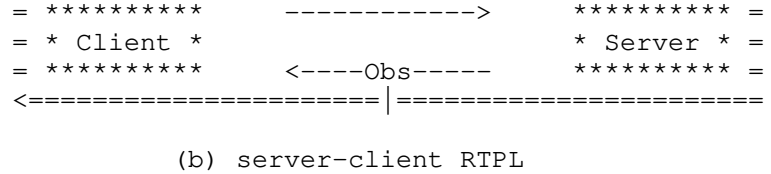
Secondly, a mechanism to individually identify each train of packets must be provided to enable the observer to distinguish between trains belonging to different phases (Generation and Reflection).

5.1. Round Trip Packet Loss measurement

Since the measurements are performed on a portion of the traffic exchanged between client and server, the observer calculates the end-to-end Round Trip Packet Loss that, statistically, will be equal to the loss rate experienced by the connection along the entire network path. So this measurement can be simply referred as the Round Trip Packet Loss (RTPL).



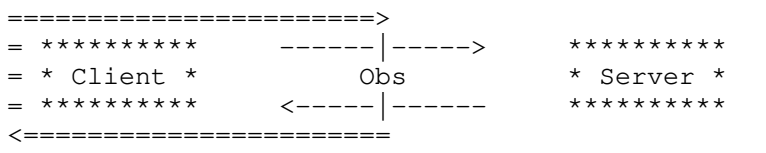
(a) client-server RTPL



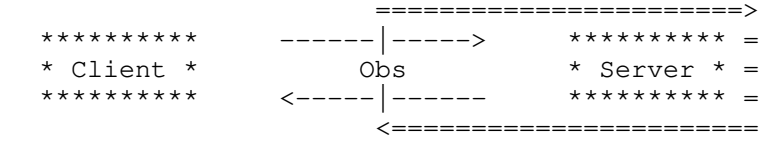
(b) server-client RTPL

Figure 5: Round-trip packet loss (both direction)

In addition, this methodology allows the Half-RTPL measurement and the Intra-domain RTPL measurement, in the same way as described in the previous sections for RTT measurement.

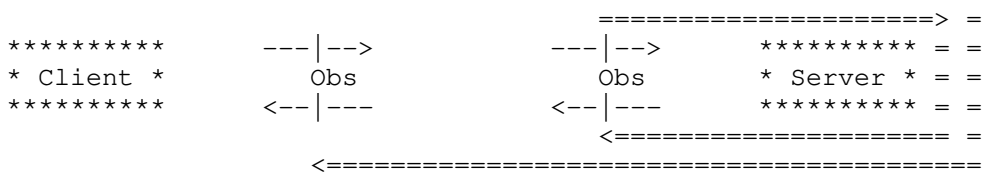


(a) client-observer half-RTPL

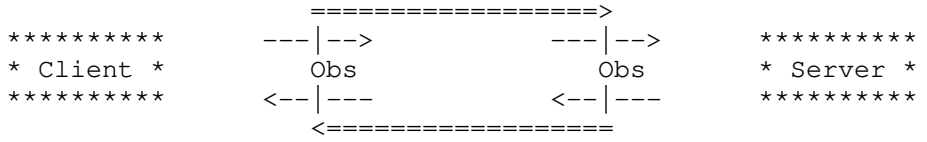


(b) observer-server half-RTPL

Figure 6: Half Round-trip packet loss (both direction)



(a) observer-server RTPL components (half-RTPLs)



(b) the intra-domain RTPL resulting from the subtraction of the above RTPL components

Figure 7: Intra-domain Round-trip packet loss (observer-server)

6. Packet Loss using one bit loss signal

The single bit loss signal, whose basic mechanism was generalized in the previous section, is implemented using just one bit: marked packets have this bit set to 1, whereas unmarked ones have it set to 0. This solution requires a working spin-bit signal used to separate

different trains of packets. In particular, a "pause" of at least one empty spin-bit period is introduced between each phase of the algorithm. An on-path observer can determine in this way if a phase (and therefore a train of packets) is ended and a new one is starting.

The client is in charge of almost the entire complexity of the algorithm. Its task can be summarized in 4 different points:

1. The client starts generating marked packets for two consecutive spin-bit periods; it maintains a generation token that is enabled every time a packet arrives and disabled when another one is forwarded. When this token is disabled, the generation process is paused (i.e. outgoing packets are transmitted unmarked) and resumes as soon as its value returns true, and that happens as soon as a packet is received. In addition, at the end of the first spin-bit period spent in generation, the reflection counter is unlocked to start counting incoming marked packets which will be later reflected;

2. When the generation is completed, the client waits to see in input an empty spin-bit period so as to be sure that everyone has seen at least that empty period. This one will be used by the observer as a divider between generated and reflected packets. During this phase, all the outgoing packets are forwarded with the loss bit set to 0. The reflection counter is still incremented every time a marked packet arrives;

3. The client starts reflecting marked packets until the reflection counter is zeroed; the generation token is also used (in the same way) during this phase to avoid congestion on the slowest traffic direction. In addition, at the end of the first spin-period spent in reflection, the reflection counter is locked to avoid incoming reflected packets incrementing it;

4. When the reflection is completed, the client waits to see in input an empty spin-bit period so as to be sure that everyone has seen at least that empty period. This one will be used by the observer as a divider between reflected and newly generated packets. During this phase, all the outgoing packets are forwarded with the loss bit set to 0. The whole process restarts going back to the first point.

As previously anticipated, the server simply reflects each incoming marked packet sent by the client. It maintains a simple counter that is incremented every time a marked packet arrives and decremented when a marked one is sent in the opposite direction.

6.1. Observer's logic for one bit loss signal

The on-path observer, placed in any direction, counts marked packets and separates different trains detecting empty spin-bit periods between them (one or more). Then, it simply computes the difference between a Generation train and a Reflection train to produce a statistical measurement of the Round Trip Packet Loss (RTPL) and of the connection end-to-end loss rate.

Here is an example. Packets are represented by two digits (first one is the spin bit, second one is the loss bit):

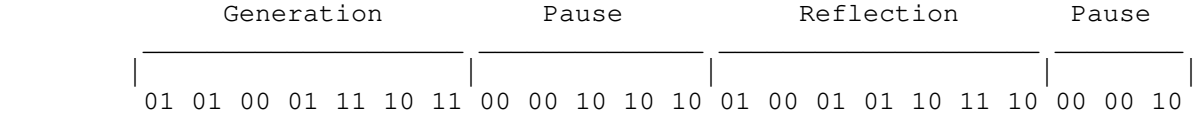


Figure 8: one bit loss signal example

Note that 5 marked packets have been generated of which 4 reflected.

7. Two Bits packet loss measurement using alternate marking

An alternative methodology, based on the classical alternate marking RFC 8321 [RFC8321], can be deployed to enable passive packet loss measurement in a connection oriented communication. This section explains its fundamentals and all the metrics that can be achieved by exploiting this mechanism.

Two new loss bits are introduced:

- Square Bit (Q): this bit is toggled every N outgoing packets generating a square signal as already seen in the alternate marking methodology RFC 8321 [RFC8321].

- Reflection Square Bit (R): this bit is used to reflect the incoming square signal (the one generated by the opposite endpoint) according to the algorithm explained in next Section; in a nutshell, it is used to report the losses found in the opposite transmission channel.

7.1. Setting the square bit (Q) on outgoing packets

The sQuare value is initialized to 0 and is applied to the Q-bit of every outgoing packet. The sQuare value is toggled after sending N packets (e.g. 64). By doing so, each endpoint splits its outgoing

traffic into blocks of N packets with different "packet color" as defined by RFC 8321 [RFC8321]. A single block of N packets is called "marking period". Observation points can estimate upstream losses by counting the number of packets included in a marking period of the produced square signal.

7.2. Setting the reflection square bit (R) on outgoing packets

Unlike the sQuare signal for which packets are transmitted into blocks of fixed size, the Reflection square signal (being an alternate marking signal too) produces blocks of packets whose size varies according to these simple rules:

- when the transmission of a new block starts, its size is set equal to the size of the last marking period whose reception has been completed;

- if, before transmission of the block is terminated, the reception of at least one further marking period is completed, the size of the block is updated to the average size of the further received marking periods. Implementation details follow.

The Reflection square value is initialized to 0 and is applied to the R-bit of every outgoing packet. The Reflection square value is toggled for the first time when the completion of a marking period is detected in the incoming sQuare signal (produced by the opposite node using the Q-bit). When this happens, the number of packets (p), detected within this first marking period, is used to generate a reflection square signal which toggles every M=p packets (at first). This new signal produces blocks of M packets (marked using the R-bit) and each of them is called "reflection marking period".

The M value is then updated every time a completed marking period in the incoming sQuare signal is received, following this formula: M=round(avg(p)).

The parameter avg(p) is the average number of packets in a marking period computed considering all the marking periods received since the beginning of the current reflection marking period.

Looking at the R-bit, observation points have clear indication of losses experienced by the entire opposite channel plus those occurred in the path from the sender up to them (if losses occur in this latter portion of path).

7.2.1. Determining the completion of an incoming marking period

A simple sQuare bit transition cannot be used to determine the completion of a marking period. Indeed, packet reordering can lead to the generation of spurious edges in the sQuare signal. To address this problem, a marking period is considered ended when at least X packets (e.g. 5) with reverse marking (i.e. belonging to the following marking period) have been received.

This same approach can be used by observation points to clean both sQuare and Reflection square signals.

7.3. Observer's logic and passive loss measurements

Since both sQuare and Reflection square bits are toggled at most every N packets (except for the first transition of the R-bit as explained before), an on-path observer can trivially count the number of packets of each marking block and, knowing the value of N, can estimate the amount of loss experienced by the connection. Different metrics can be measured depending on which direction the observer is looking to.

One direction observer:

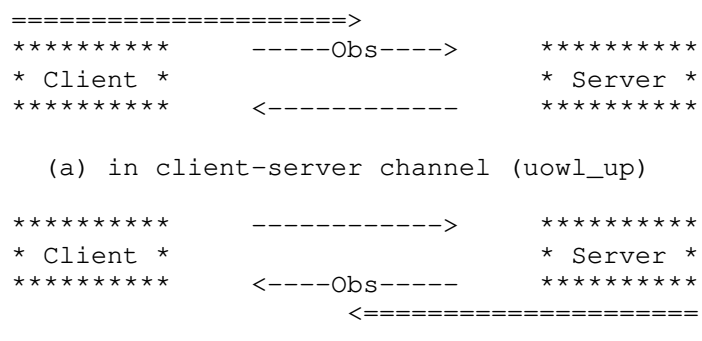
- upstream one-way loss: the loss between the sender and the observation point
- "three-quarters" connection loss: the loss between the receiver and the sender in the opposite direction plus the loss between the sender and the observation point in the observed direction
- full one-way loss in the opposite direction: the loss between the receiver and the sender in the opposite direction

Two directions observer (same metrics seen previously applied to both direction, plus):

- client-observer half round-trip loss: the loss between the client and the observation point in both directions
- observer-server half round-trip loss: the loss between the observation point and the server in both directions
- downstream one-way loss: the loss between the observation point and the receiver (valid for both directions)

7.3.1. Upstream one-way loss

Since packets are continuously Q-bit marked into alternate blocks of size N, knowing the value of N, an on-path observer can estimate the amount of loss occurred from the sender up to it after observing at least N packets. The upstream one-way loss rate ("uowl") is one minus the average number of packets in a block of packets with the same Q value ("p") divided by N ("uowl=1-avg(p)/N").



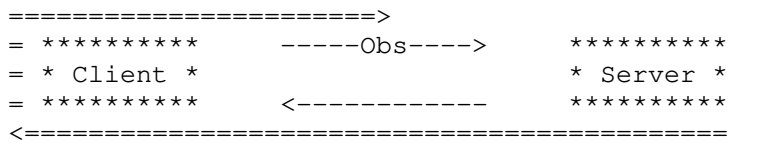
(a) in client-server channel (uowl_up)

(b) in server-client channel (uowl_down)

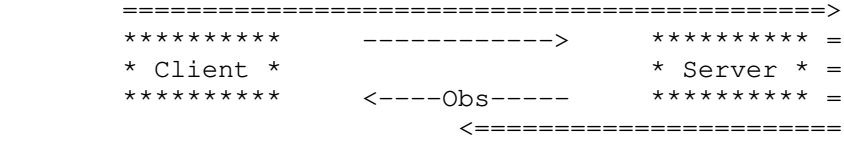
Figure 9: Upstream one-way loss

7.3.2. Three-quarters connection loss

Except for the very first block in which there is nothing to reflect (a complete marking period has not been yet received), packets are continuously R-bit marked into alternate blocks of size lower or equal than N. Knowing the value of N, an on-path observer can estimate the amount of loss occurred in the whole opposite channel plus the loss from the sender up to it in the observation channel. As for the previous metric, the "three-quarters" connection loss rate ("tql") is one minus the average number of packets in a block of packets with the same R value ("t") divided by N ("tql=1-avg(t)/N").



(a) in client-server channel (tql_up)



(b) in server-client channel (tql_down)

Figure 10: Three-quarters connection loss

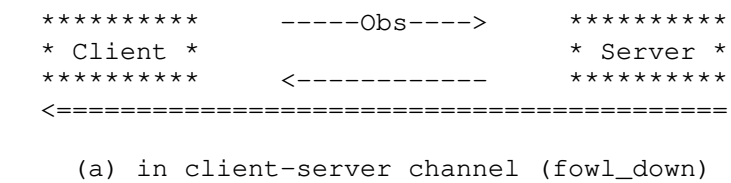
The following metrics derive from these first two metrics.

7.3.3. Full one-way loss in the opposite direction

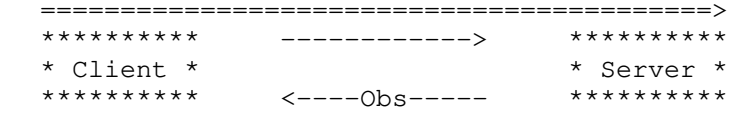
Using the previous metrics, full one-way loss can be computed:

fowl_down = tql_up - uowl_up

fowl_up = tql_down - uowl_down



(a) in client-server channel (fowl_down)



(b) in server-client channel (fowl_up)

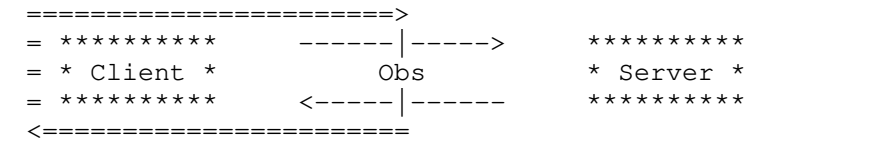
Figure 11: Full one-way loss in the opposite direction

7.3.4. Half round-trip loss

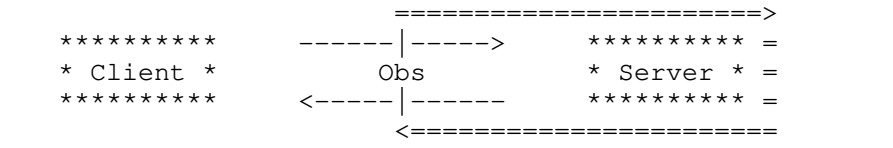
Using the previous metrics, the two half round-trip loss measurements can be computed:

hrtl_co = tql_up - uowl_down

hrtl_os = tql_down - uowl_up



(a) client-observer half round-trip loss (hrtl_co)



(b) observer-server half round-trip loss (hrtl_os)

Figure 12: Half Round-trip loss (both direction)

7.3.5. Downstream one-way loss

Using the previous metrics, downstream one-way loss can be computed:

dowl_up = hrtl_os - uowl_down

dowl_down = hrtl_co - uowl_up

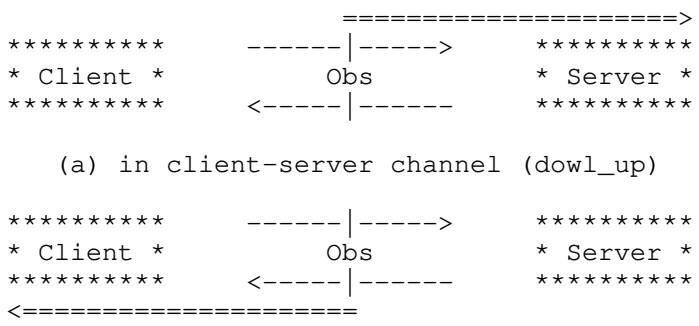


Figure 13: Downstream one-way loss

7.4. Enhancement of reflection period size computation

The use of the rounding function used in the M computation introduces errors. However, these errors can be minimized by storing the rounding applied each time M is computed, and using it during the computation of the M value in the following reflection marking period.

This can be achieved introducing the new r_avg parameter in the previous M formula. The new formula is M=round(avg(p)+r_avg) where r_avg is computed as not rounded M minus rounded M; its initial value is equal to 0.

7.5. Improvement of the resilience to out of sequence

Since endpoints have clear indication about reordered packets, we can use this information to absorb out of sequences in the incoming sQuare wave, even when the marking period threshold (see 7.2.1 Section) has been reached.

This can be achieved by updating the size of the current reflection block while this is being transmitted. The reflection block size is then updated every time an incoming reordered packet of the previous marking period is detected. This can be done if and only if the transmission of the current reflection block is in progress and no packets of the following marking period (Q-bit) have been received.

8. Protocols

8.1. QUIC

The binding of the delay bit signal to QUIC is partially described in [I-D.ietf-quic-transport], which adds the spin bit to the first byte of the short packet header, leaving two reserved bits for future experiments.

To implement the additional signals discussed in this document, the first byte of the short packet header can be modified as follows:

the delay bit (D) can be placed in the first reserved bit (i.e. the fourth most significant bit _0x10_) while the loss bit in the second reserved bit (i.e. the fifth most significant bit _0x08_); the proposed scheme is:

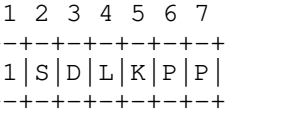


Figure 14: scheme 1

alternatively, the standalone two bits loss signal (QR) can be placed in both reserved bits; the proposed scheme, in this case, is:

Figure 15: scheme 2

8.2. TCP

The signals can be added to TCP by defining bit 4 of bytes 13-14 of the TCP header to carry the spin bit, and eventually bits 5 and 6 to carry additional information, like the delay bit and the 1 bit loss signal (or the two bits loss signal).

9. Security Considerations

The privacy considerations for the hybrid RTT measurement signal are essentially the same as those for passive RTT measurement in general.

10. Acknowledgements

tbc

11. IANA Considerations

tbc

12. References

12.1. Normative References

[I-D.ietf-quic-spin-exp]
Trammell, B. and M. Kuehlewind, "The QUIC Latency Spin Bit", draft-ietf-quic-spin-exp-01 (work in progress), October 2018.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-29 (work in progress), June 2020.

[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>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

12.2. Informative References

[ANRW19-PM-QUIC]
ACM/IRTF Applied Networking Research Workshop 2019 (ANRW'19), "Performance measurements of QUIC communications", DOI 10.1145/3340301.3341127, 2019.

[I-D.trammell-ippm-spin]
Trammell, B., "An Explicit Transport-Layer Signal for Hybrid RTT Measurement", draft-trammell-ippm-spin-00 (work in progress), January 2019.

[I-D.trammell-quic-spin]
Trammell, B., Vaere, P., Even, R., Fioccola, G., Fossati, T., Ihlar, M., Morton, A., and S. Emile, "Adding Explicit Passive Measurability of Two-Way Latency to the QUIC Transport Protocol", draft-trammell-quic-spin-03 (work in progress), May 2018.

[I-D.trammell-tsvwg-spin]
Trammell, B., "A Transport-Independent Explicit Signal for Hybrid RTT Measurement", draft-trammell-tsvwg-spin-00 (work in progress), July 2018.

Authors' Addresses

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Giuseppe Fioccola
Huawei Technologies
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Massimo Nilo
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: massimo.nilo@telecomitalia.it

Fabio Bulgarella
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: fabio.bulgarella@guest.telecomitalia.it

Riccardo Sisto
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Torino 10129
Italy

Email: riccardo.sisto@polito.it

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 13, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
N. Vaghamshi
Reliance
M. Nagarajah
Telstra
R. Foote
Nokia

February 09, 2021

Enhanced Performance and Liveness Monitoring in Segment Routing Networks
draft-gandhi-spring-sr-enhanced-plm-04

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document defines procedures for Enhanced Performance and Liveness Monitoring (PLM) for end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, those reduce the deployment and operational complexities in a network.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 13, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic

through a specific, user-defined paths using a stack of Segments. Built-in Performance Measurement as well as Liveness Monitoring for Connectivity Verification (CV) and Continuity Check (CC) are essential requirements to provide Service Level Agreements (SLAs) in SR networks.

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762]. It eliminates the need for control protocol by using configuration and management model to provision and manage test sessions. The STAMP can be used for Performance Measurement (PM) in SR networks as well as liveness monitoring and connectivity loss detection of SR paths. However, the STAMP requires protocol support on the Session-Reflector to process the STAMP test packets as packets need to be punted from the forwarding fast path (to slow path or control plane) on the Session-Reflector and STAMP reply test packets need to be generated. This limits the scale for number of STAMP test sessions and faster fault detection intervals.

For Liveness Monitoring, Seamless Bidirectional Forwarding Detection (S-BFD) [RFC7880] can be used in SR networks. However, S-BFD requires protocol support on the BFD-Reflector to process the S-BFD packets as packets need to be punted from the forwarding fast path and generate the reply packets thereby limiting the scale for number S-BFD sessions and faster fault detection intervals. In addition, S-BFD protocol is not defined to enable performance measurement in a network.

Enabling multiple protocols, S-BFD for liveness monitoring and STAMP for performance measurement increases the deployment and operational complexities a network. Also, implementing multiple protocols in a hardware significantly increases the development cost.

This document defines procedures for Enhanced Performance and Liveness Monitoring (PLM) for end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, those reduce the deployment and operational complexities in a network. The procedures use the new test packet formats those have the timestamps at the same locations as the base STAMP test packets to leverage the existing hardware support for STAMP.

2. Conventions Used in This Document

2.1. 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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

S-BFD: Seamless Bidirectional Forwarding Detection.

BSID: Binding Segment ID.

ECMP: Equal Cost Multi-Path.

EB: Endpoint Behaviour.

HMAC: Hashed Message Authentication Code.

MBZ: Must be Zero.

MPLS: Multiprotocol Label Switching.

PLM: Performance and Liveness Monitoring.

PM: Performance Measurement.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

SSID: Sender Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

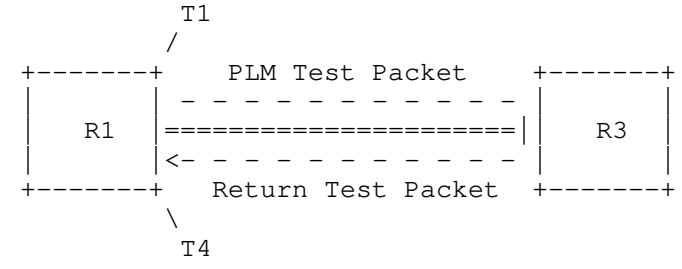
TC: Traffic Class.

TTL: Time To Live.

2.3. Reference Topology

In the reference topology shown below, the Session-Sender R1 initiates a PLM test packet and the Session-Reflector R3 transmits a PLM return test packet. The PLM return test packet is transmitted back to the Session-Sender R1 on the same path or a different path in the reverse direction.

The Session-Sender R1 and Session-Reflector R3 are connected via an SR path [RFC8402]. The SR path may be an SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R3 (called tail-end).



Reference Topology

3. Overview

3.1. Loopback Mode

In loopback mode, the Session-Sender R1 initiates PLM test packets and the Session-Reflector R3 forwards them just like data packets for the regular traffic back to the Session-Sender R1. The PLM test packets are not punted at the Session-Reflector and does not process them and generate PLM return test packets. The Session-Reflector must not drop the loopback PLM test packets, for example, due to a local policy provisioned. No PLM test session is created on the Session-Reflector.

The Source and Destination IP addresses in the PLM test packets are set to the Session-Reflector and the Session-Sender IP addresses, respectively (representing the reverse direction path). The Source and Destination UDP ports in the PLM test packets follow the procedure defined in [RFC8762]. The IPv4 Time To Live (TTL) and IPv6 Hop Limit (HL) are set to 255.

3.2. Loopback Mode Enabled with Network Programming Function

In loopback mode enabled with network programming function, both transmit (T1) and receive (T2) timestamps in data plane are collected by the PLM test packets transmitted in loopback mode as shown in Figure 1. The network programming function optimizes the "operations of punt and generate the PLM test packet" on the Session-Reflector as timestamping is implemented in forwarding fast path in hardware. This helps to achieve higher test session scale and faster failure detection interval.

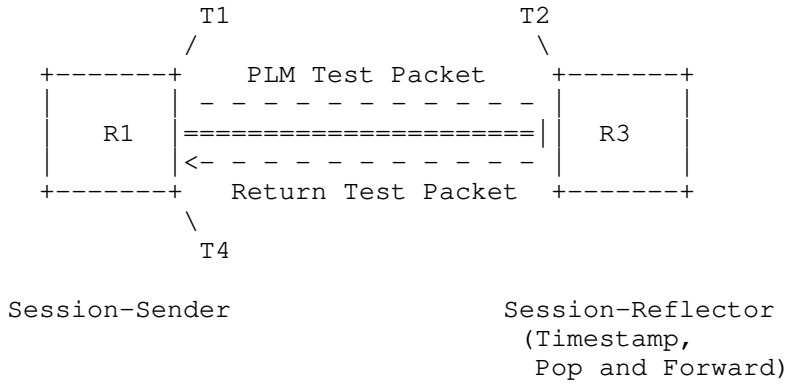


Figure 1: Loopback Mode Enabled with Network Programming Function

The Session-Sender adds transmit timestamp (T1) in the payload of the PLM test packet and clears the receive (T2) timestamp. The Session-Reflector adds the receive timestamp (T2) in the payload of the received PLM test packet in forwarding fast path in hardware without punting the test packet to the slow path (or control-plane). The network programming function enables Session-Reflector to add the receive timestamp (T2) at a specific offset in the payload which is locally provisioned consistently in the network. The payload of the PLM test packet is not modified by the intermediate nodes.

The Session-Reflector only adds the receive timestamp if the source IP address (in case of SR-MPLS) or destination IP address (in case of SRv6) in the PLM test packet matches the local node address to ensure that the PLM test packet reaches the intended Session-Reflector and the receive timestamp is returned by the intended Session-Reflector.

3.3. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 2:

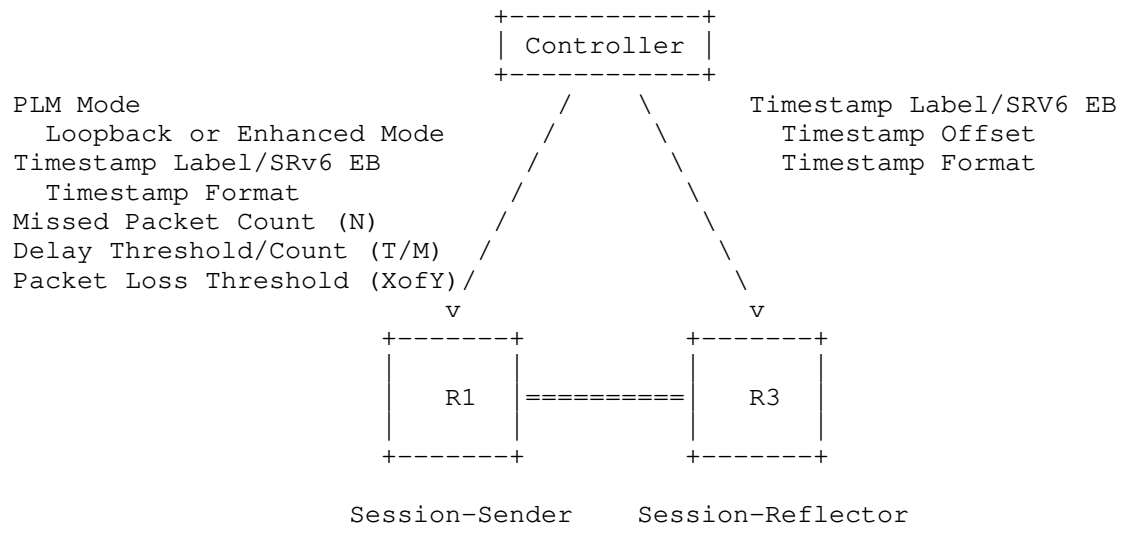


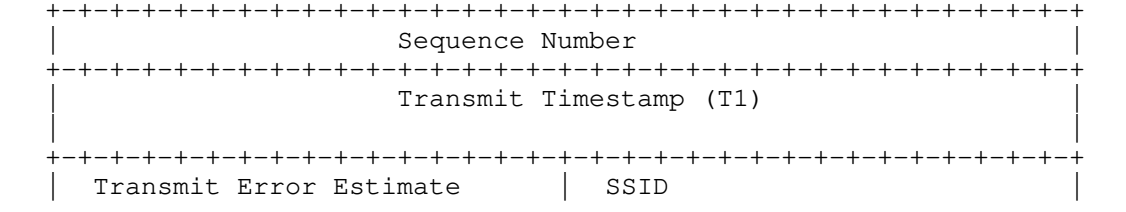
Figure 2: Example Provisioning Model

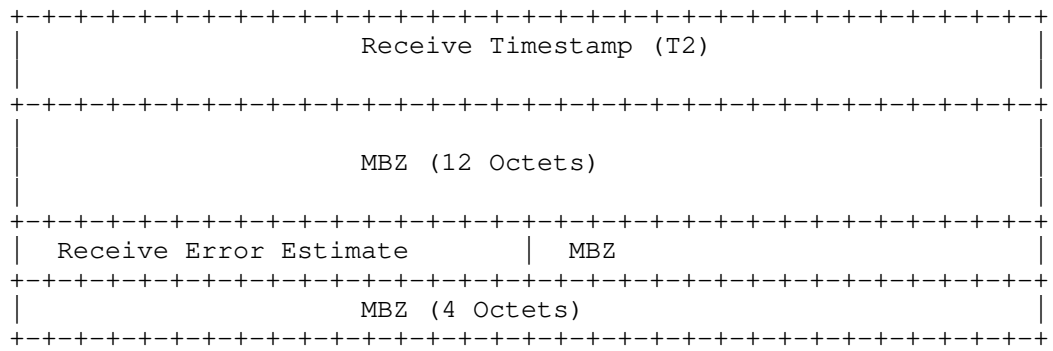
Example of PLM mode is loopback mode. The values for Timestamp Label and SRv6 Endpoint Behaviour may be provisioned as described in Section 6. Example of Timestamp Format is 64-bit PTPv2 [IEEE1588]. Example of Timestamp Offset is 16 and 32 bytes for the PLM test packet formats defined in this document. Example threshold values configured for generating notifications are: Missed Packet Count (N), Delay Exceeded Threshold and Packet Count (T/M) and Packet Loss Threshold (XofY), as described in Section 7.

The mechanisms to provision the Session-Sender and Session-Reflector are outside the scope of this document.

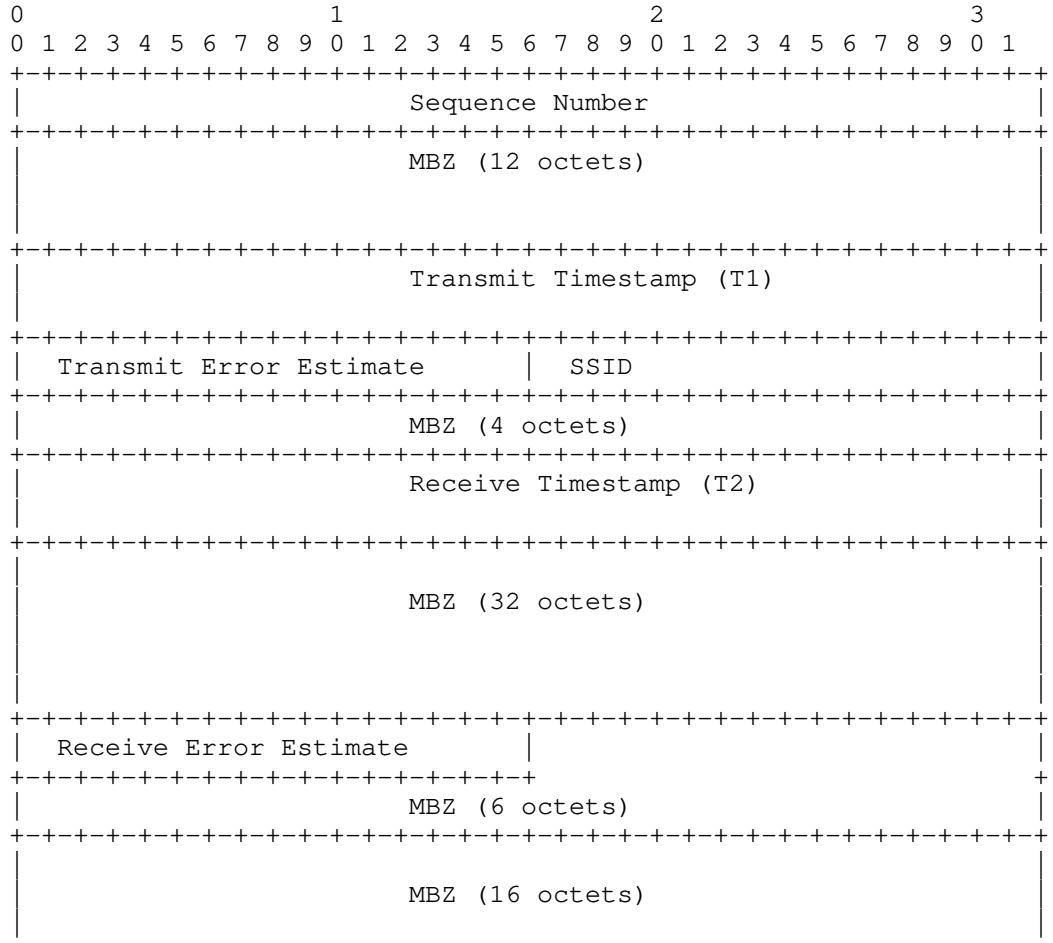
4. PLM Test Packet Formats

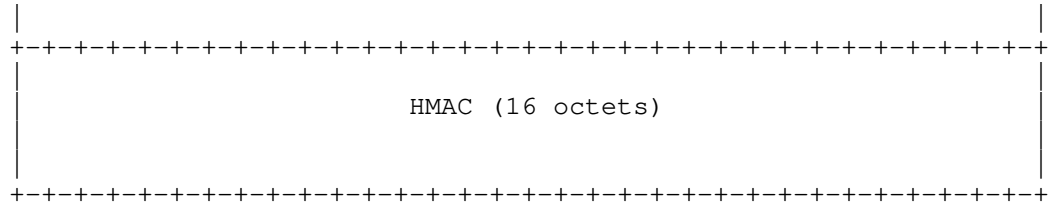
The PLM test packet formats for unauthenticated and authenticated modes are defined in this document as shown in Figure 3 those have the transmit and receive timestamps at the same locations as the base STAMP test packets to leverage the existing hardware support for STAMP.





PLM Test Packet Format in Unauthenticated Mode





PLM Test Packet Format in Authenticated Mode

Figure 3: PLM Test Packet Formats

Sequence Number is the sequence number of the PLM test packet according to its transmit order. It starts with zero and is incremented by one for each subsequent PLM test packet.

SSID (16-bits): PLM Sender Session Identifier. Uses the procedure for SSID defined in [RFC8762].

Transmit Timestamp and Transmit Error Estimate are the Session-Sender's transmit timestamp and error estimate for the PLM test packet, respectively.

Receive Timestamp and Receive Error Estimate are the Session-Reflector's receive timestamp and error estimate, respectively.

The timestamp and error estimate fields follow the definition and formats defined in Section 4.1.2 in [RFC8762]. The timestamp format used by default is 64-bit PTPv2 [IEEE1588].

HMAC: The use of the HMAC field is described in Section 4.4 of [RFC8762].

MBZ: Must be Zero. It MUST be all zeroed on the transmission and MUST be ignored on receipt.

5. PLM Procedure

For performance and liveness monitoring of an end-to-end SR path including SR Policy, PLM test packets in loopback mode are used.

For SR Policy, the PLM test packets are transmitted using the Segment List (SL) of the Candidate-Path [I-D.ietf-spring-segment-routing-policy]. When a Candidate-Path has more than one Segment Lists, multiple PLM test packets are sent, one using each Segment List. The PLM return test packets are received by the Session-Sender via IP/UDP [RFC0768] return path by default. The Segment List of the return SR path can be added in the PLM test

packet header to receive the return test packet on a specific path using the Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path].

5.1. PLM for SR-MPLS Policies

The PLM test packets are transmitted using the MPLS header for each Label Stack of the SR-MPLS Policy Candidate-Path(s) as shown in Figure 4. In case of IP/UDP return path, the MPLS header is removed by the Session-Reflector. The Label Stack can contain a reverse SR-MPLS path to receive the PLM return test packet on a specific path. In this case, the MPLS header will not be removed by the Session-Reflector.

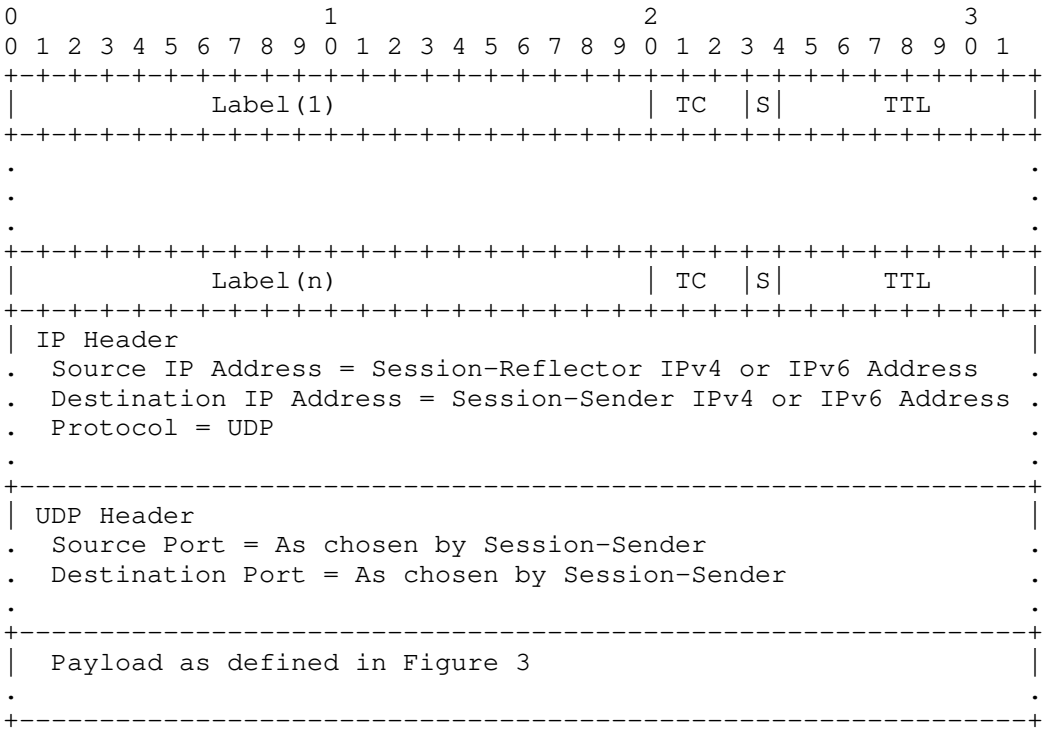


Figure 4: Example PLM Test Packet for SR-MPLS

5.2. PLM for SRv6 Policies

The PLM test packets for SRv6 data plane are transmitted using the Segment Routing Header (SRH) [RFC8754] for each Segment List of the SRv6 Policy Candidate-Path(s) as shown in Figure 5. In case of IP/UDP return path, the SRH is removed by the Session-Reflector. The

Segment List can contain a reverse SRv6 path to receive the PLM return test packet on a specific path. In this case, the SRH will not be removed by the Session-Reflector. When the PLM return test packet contains an SRH at the Session-Sender, the procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received PLM test packets.

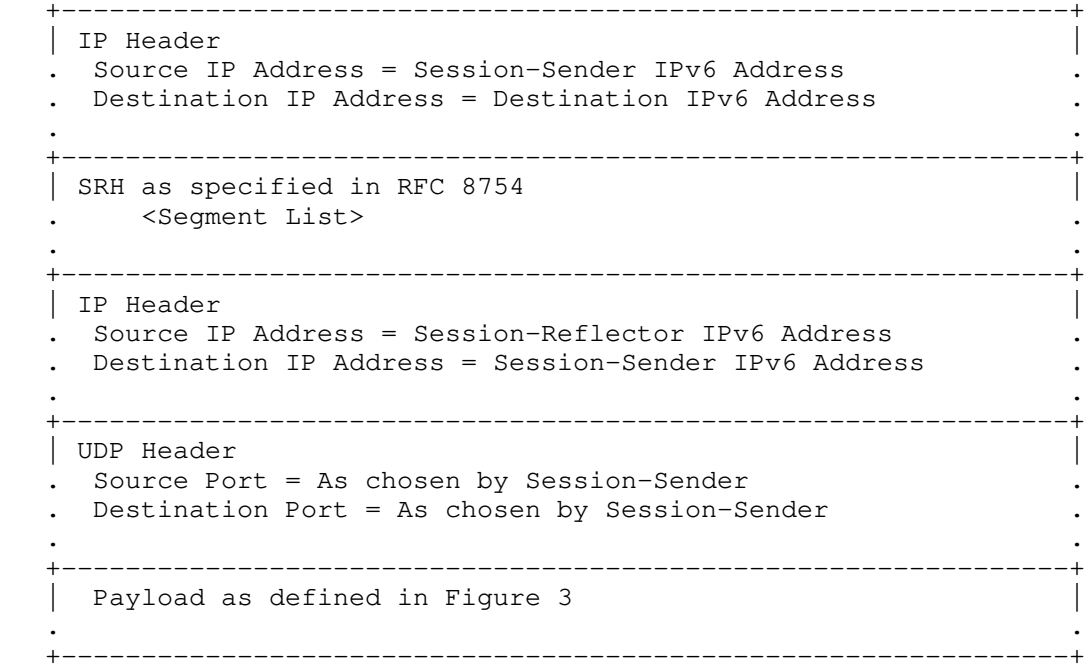


Figure 5: Example PLM Test Packet for SRv6

6. Enhanced PLM Procedure

The enhanced performance and liveness monitoring of an end-to-end SR path including SR Policy is defined using the PLM test packets in loopback mode enabled with network programming function.

6.1. Enhanced PLM with Timestamp Label for SR-MPLS Policies

In this document, two new Timestamp Labels are defined for SR-MPLS data plane to enable network programming function for "timestamp, pop and forward" the received test packet.

In the PLM test packets for SR-MPLS Policies, a Timestamp Label is added in the MPLS header as shown in Figure 6, to collect "Receive

Timestamp" field in the payload of the PLM test packet. The Label Stack for the reverse SR-MPLS path can be added after the Timestamp Label to receive the PLM return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Label, after timestamping the packet at a specific offset, the Session-Reflector pops the Timestamp Label and forwards the packet using the next label or IP header in the packet (just like the data packets for the regular traffic).

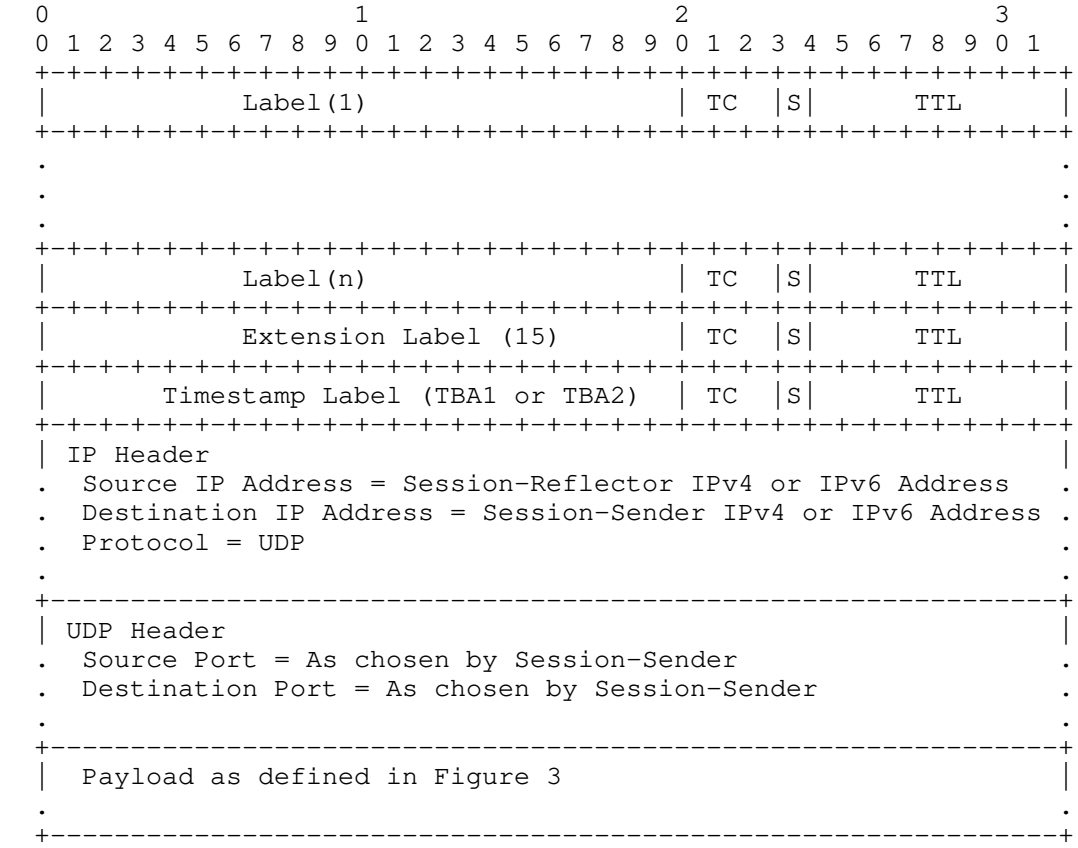


Figure 6: Example PLM Test Packet with Timestamp Label for SR-MPLS

6.1.1. Timestamp Label Allocation

The timestamp Labels for unauthenticated and authenticated modes can be allocated using one of the following methods:

- Labels (values TBA1 and TBA2) assigned by IANA from the "Extended Special-Purpose MPLS Values" [I-D.ietf-mpls-spl-terminology]. For

Label (value TBA1), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the unauthenticated mode, and Label (value TBA2) at byte-offset 32 from the start of the payload for the authenticated mode, both using the timestamp format 64-bit PTPv2.

- Labels allocated by a Controller from the global table of the Session-Reflector. The Controller provisions the labels on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.

- Labels allocated by the Session-Reflector. The signaling and IGP flooding extension for the labels (including timestamp offsets and timestamp formats) are outside the scope of this document.

6.1.2. Node Capability for Timestamp Label

The PLM Session-Sender needs to know if the Session-Reflector can process the Timestamp Label to avoid dropping PLM test packets. The signaling extension for this capability exchange is outside the scope of this document.

6.2. Enhanced PLM with Timestamp Endpoint Function for SRv6 Policies

The [I-D.ietf-spring-srv6-network-programming] defines SRv6 Endpoint Behaviours (EB) for SRv6 nodes. In this document, two new Timestamp Endpoint Behaviours are defined for Segment Routing Header (SRH) [RFC8754] to enable "Timestamp and Forward (TSF)" function for the received test packets.

In the PLM test packets for SRv6 Policies, Timestamp Endpoint Function (End.TSF) is carried with the target Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 7, to collect "Receive Timestamp" field in the payload of the PLM test packet. The Segment List for the reverse path can be added after the target SID to receive the PLM return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Endpoint (End.TSF) for the target SID which is local, after timestamping the packet at a specific offset, the Session-Reflector forwards the packet using the next SID or IP header in the packet (just like the data packets for the regular traffic).

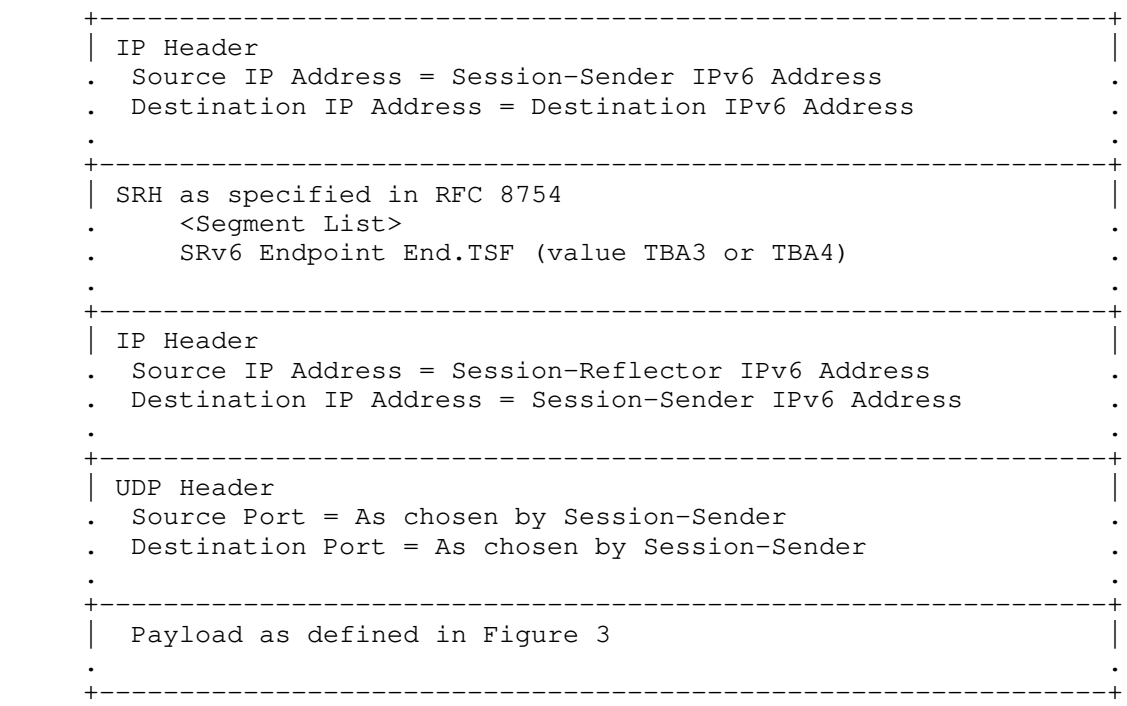


Figure 7: Example PLM Test Packet with Endpoint Function for SRv6

6.2.1. Timestamp Endpoint Function Assignment

The Timestamp Endpoint Functions for "Timestamp and Forward" can be signaled using one of the following methods:

- Timestamp Endpoint Functions (values TBA3 and TBA4) assigned by IANA from the "SRv6 Endpoint Behaviors Registry". For endpoint behaviour (value TBA3), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the unauthenticated mode, and endpoint behaviour (value TBA4) at byte-offset 32 from the start of the payload for the authenticated mode, both using the timestamp format 64-bit PTPv2.

- Timestamp Endpoint Functions assigned by a Controller. The Controller provisions the values on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.

- Timestamp Endpoint Functions assigned by the Session-Reflector. The signaling and IGP flooding extension for the endpoint

functions (including timestamp offsets and timestamp formats) are outside the scope of this document.

6.2.2. Node Capability for Timestamp Endpoint Function

The PLM Session-Sender needs to know if the Session-Reflector can process the Timestamp Endpoint Function to avoid dropping PLM test packets. The signaling extension for this capability exchange is outside the scope of this document.

7. ECMP Handling

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. The PLM test packets need to be sent to traverse different ECMP paths to monitor an end-to-end SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. In IPv4 header of the PLM test packets, sweeping of Destination Address from the 127/8 range can be used to exercise different IPv4 ECMP paths in both loopback modes as long as the forward and the return paths are SR-MPLS paths. In this case, the TTL field in the IPv4 header is set to 1.

The Flow Label field in the outer IPv6 header can also be used for sweeping to exercise different IPv6 ECMP paths.

8. Example PLM Failure Notifications

Liveness or connectivity success for an end-to-end SR path is initially notified as soon as one or more PLM return test packets are received at the Session-Sender.

Liveness or connectivity failure for an end-to-end SR path is notified when consecutive N number of PLM return test packets are not received at the Session-Sender, where N (Missed PLM Packet Count) is a locally provisioned value.

The round-trip packet loss for an end-to-end SR path is calculated using the Sequence Number in the PLM test packets. The packet loss metric is notified when X number of PLM test packets were lost out of last Y number of PLM test packets transmitted by the Session-Sender, where Threshold XofY is locally provisioned value.

Similarly, the delay metrics are notified, as an example, when consecutive M number of PLM test packets have measured delay values exceed user-configured threshold T, where M (Delay Exceeded Packet

Count) and T (Absolute and Percentage Delay Exceeded Threshold) are also locally provisioned values.

In both loopback modes, the timestamps T1 and T4 are used to measure round-trip delay. In loopback mode enabled with network programming function, the timestamps T1 and T2 are used to measure one-way delay.

In both loopback modes, a failure on the reverse direction path can cause the PLM return test packets to not reach the Session-Sender. This is also true in the case where the PLM return test packets were generated by the Session-Reflector e.g. to indicate Session-Sender of a failure on the forward direction path. As such, the test packet based methods have a limitation of false detection due to a reverse direction failure.

9. Security Considerations

The Performance and Liveness Monitoring is intended for deployment in the well-managed private and service provider networks. As such, it assumes that a node involved in a monitoring operation has previously verified the integrity of the path and the identity of the Session-Reflector.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the Session-Sender, of the timestamp fields in received PLM packets. The minimal state associated with these protocols also limits the extent of disruption that can be caused by a corrupt or invalid packet to a single test cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the test packets. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

The security considerations specified in [RFC8762] also apply to the procedures defined in this document.

10. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>). IANA is requested to allocate Timestamp Label value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	Timestamp Label for offset 16 for Unauthenticated Mode	This document
TBA2	Timestamp Label for offset 32 for Authenticated Mode	This document

IANA is requested to allocate, within the "SRv6 Endpoint Behaviors Registry" sub-registry belonging to the top-level "Segment Routing Parameters" registry [I-D.ietf-spring-srv6-network-programming], the following allocation:

Value	Endpoint Behavior	Reference
TBA3	End.TSF (Timestamp and Forward) for offset 16 for Unauthenticated Mode	This document
TBA4	End.TSF (Timestamp and Forward) for offset 32 for Authenticated Mode	This document

11. References

11.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[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>.

[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>.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

11.2. Informative References

[IEEE1588]
IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-mpls-spl-terminology]
Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-06 (work in progress), January 2021.

[I-D.ietf-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-ietf-pce-binding-label-sid-05 (work in progress), October 2020.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-05 (work in progress), January 2021.

Acknowledgments

The authors would like to thank Greg Mirsky, Mach Chen, Kireeti Kompella, and Adrian Farrel for providing the review comments.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Navin Vaghamshi
Reliance

Email: Navin.Vaghamshi@ril.com

Moses Nagarajah
Telstra

Email: Moses.Nagarajah@team.telstra.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
M. Chen
Huawei
B. Janssens
Colt

October 21, 2020

Performance Measurement Using TWAMP Light for Segment Routing Networks
draft-gandhi-spring-twamp-srpm-11

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the mechanisms defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP) Light) and its extensions for Performance Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks using probe messages. These protocols rely on control-channel signaling to establish a test-channel over an UDP path. The TWAMP Light [Appendix I in RFC5357] [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer IP networks by provisioning UDP paths and eliminates the need for control-channel signaling.

This document specifies procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the mechanisms defined in [RFC5357] (TWAMP Light) and its extensions for Performance Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths. Unless otherwise specified, the mechanisms defined in [RFC5357] are not modified by this document.

2. Conventions Used in This Document

2.1. 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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

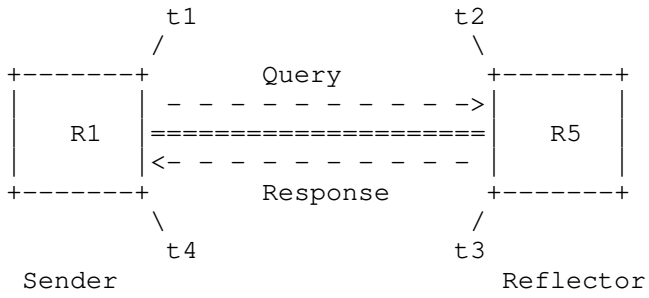
TC: Traffic Class.

TWAMP: Two-Way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a performance measurement probe query message and the reflector node R5 sends a probe response message for the query message received. The probe response message is typically sent to the sender node R1.

SR is enabled on nodes R1 and R5. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R5 (called tail-end).



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC5357] are used. For direct-mode and inferred-mode loss measurements, the probe messages defined in [I-D.gandhi-ippm-twamp-srpm] are used. For both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths, no PM state for delay or loss measurement need to be created on the reflector node R5.

Separate UDP destination port numbers are user-configured for delay and loss measurements. As specified in [RFC8545], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. The same destination UDP port is used for Links and SR Paths and the reflector is unaware if the query is for the Links or SR Paths. The number of UDP ports with PM functionality needs to be minimized due to limited hardware resources.

For Performance Measurement, probe query and response messages are sent as following:

- For delay measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Paths.

- For loss measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

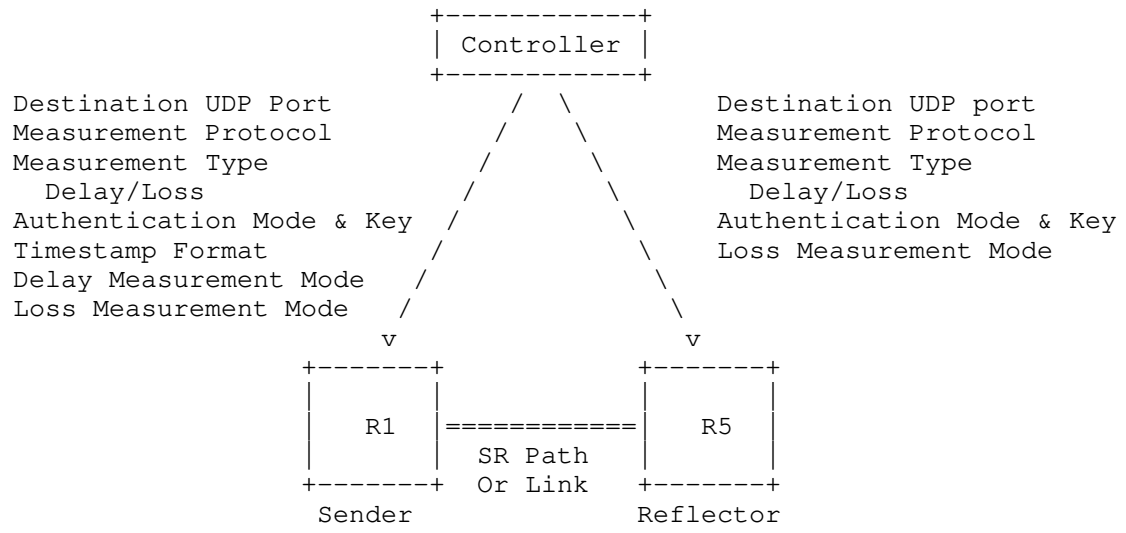


Figure 1: Example Provisioning Model

Example of Measurement Protocol is TWAMP Light, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

The mechanisms to provision the sender and reflector nodes are outside the scope of this document. The provisioning model is not used for signaling the PM parameters between the reflector and sender nodes in SR networks.

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC5357] are used for delay measurement for Links and end-to-end SR Paths including SR Policies. For loss measurement, the probe messages defined in [I-D.gandhi-ippm-twamp-srpm] are used.

4.1.1. Delay Measurement Query Message

The message content for delay measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in Section 4.1.2 of [RFC5357]. For symmetrical size query and response messages as defined in [RFC6038], the DM probe query message contains the payload format defined in Section 4.2.1 of [RFC5357].

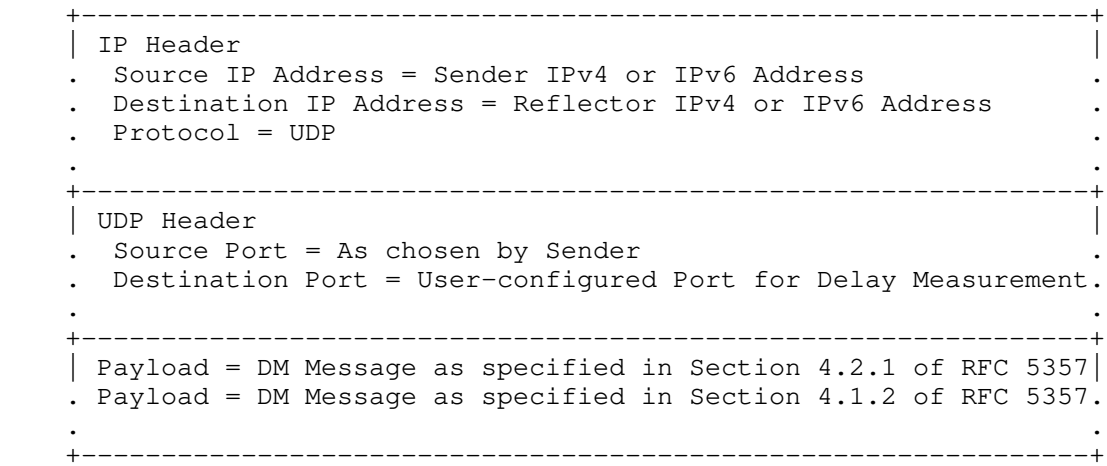


Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC5357]. It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for loss measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in [I-D.gandhi-ippm-twamp-srpm].

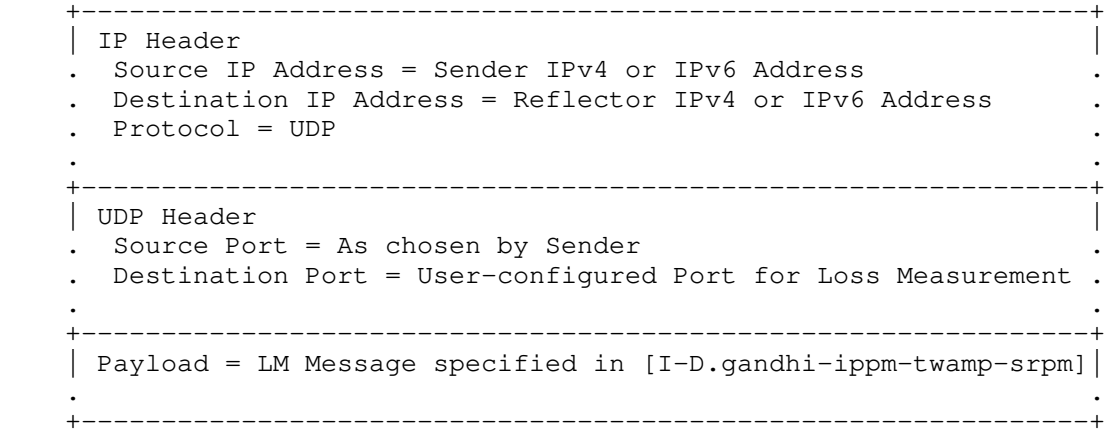


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement are used for Links which may be physical, virtual or LAG (bundle), LAG (bundle) member, numbered/ unnumbered Links. The probe messages are pre-routed over the Link for both delay and loss measurement. The local and remote IP addresses of the link are used as Source and Destination Addresses. They can also be IPv6 link local address as probe messages are pre-routed.

4.1.4. Probe Query for SR Policy

The performance delay and loss measurement for segment routing is applicable to both end-to-end SR-MPLS and SRv6 Policies.

The sender IPv4 or IPv6 address is used as the source address. The endpoint IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the loopback address from range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 is used as the destination address, respectively.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for performance measurement of an end-to-end SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

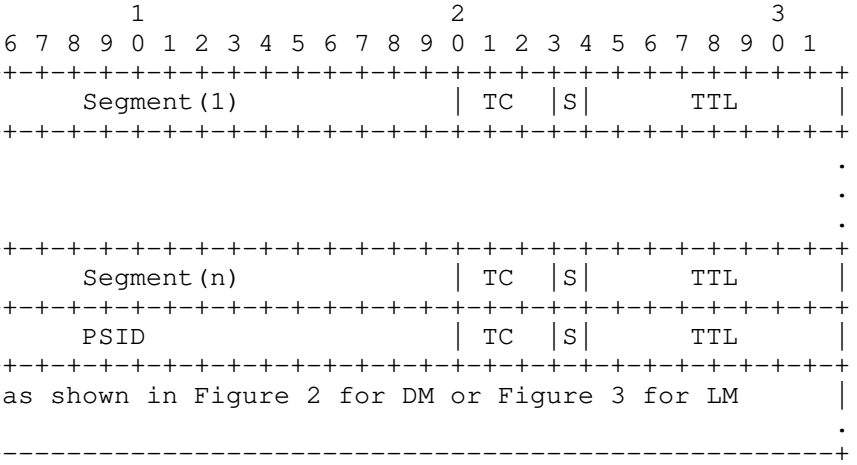


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. The SRv6 network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for performance measurement of an end-to-end SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe query messages.

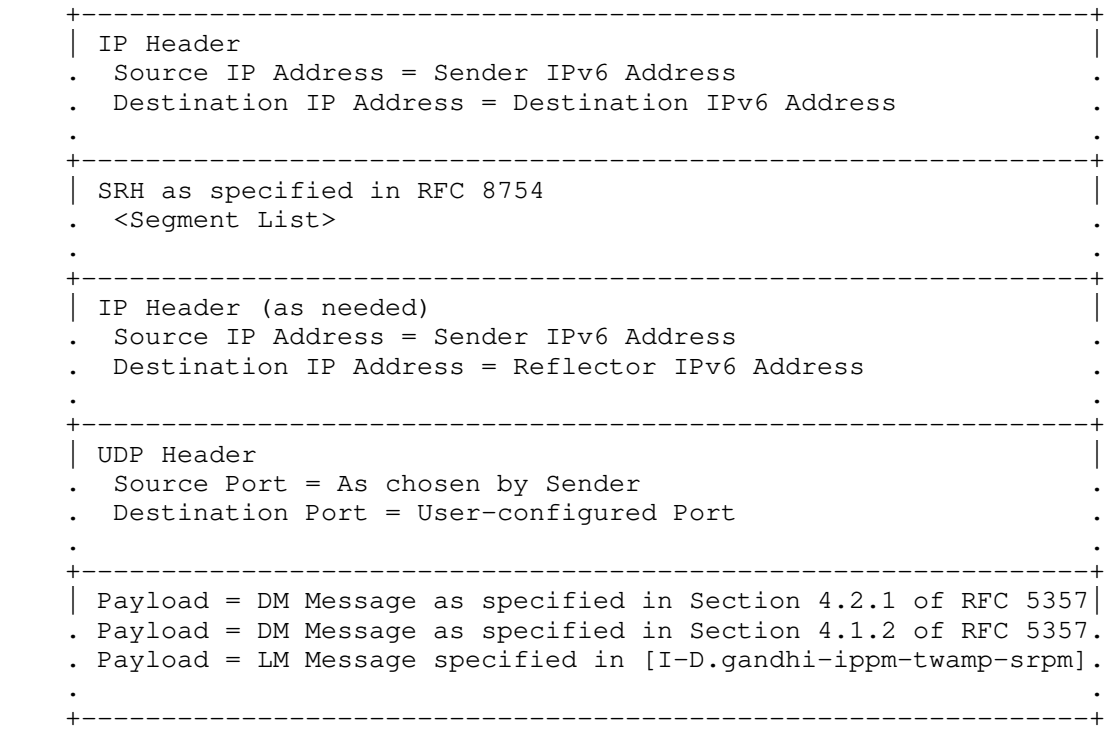


Figure 5: Example Probe Query Message for SRv6 Policy

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 6.

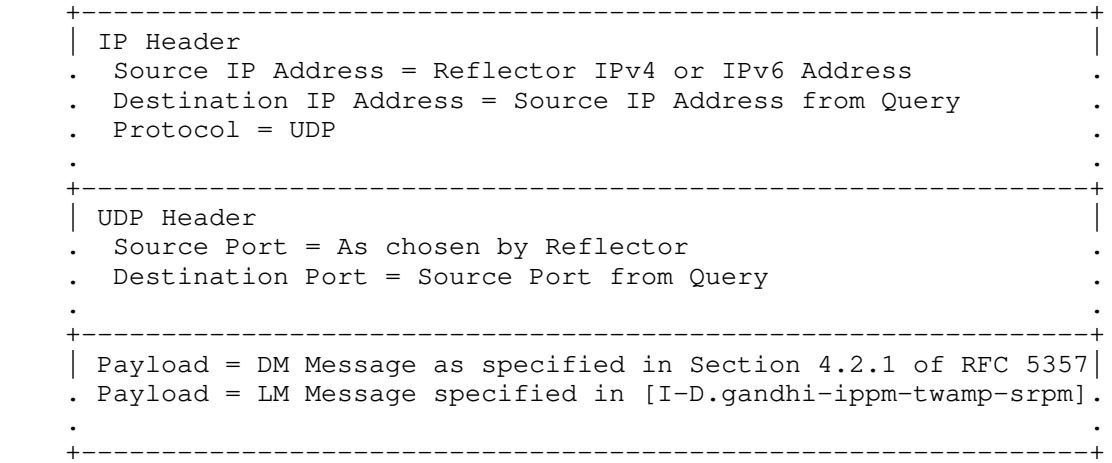


Figure 6: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way measurement mode, the probe response message as defined in Figure 6 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. However, only timestamps t1 and t2 are used to measure one-way delay as (t2 - t1).

4.2.2. Two-way Measurement Mode

In two-way measurement mode, when using a bidirectional path, the probe response message as defined in Figure 6 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. All four timestamps are used to measure two-way delay as ((t4 - t1) - (t3 - t2)).

For two-way measurement mode for Links, the probe response message is sent back on the incoming physical interface where the probe query message is received.

4.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message for two-way performance measurement of an end-to-end SR-MPLS Policy is shown in Figure 7.

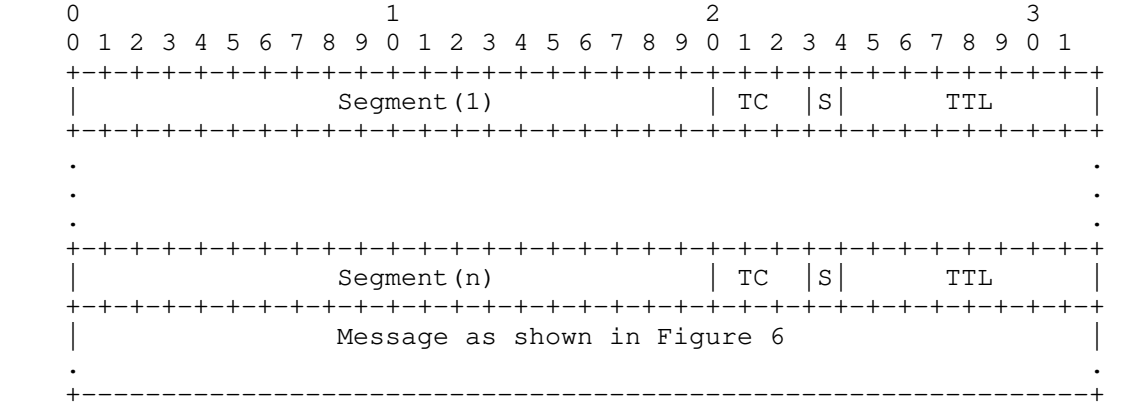


Figure 7: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR Policy in the probe query can be used to find the associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path] to send the probe response message for two-way measurement of SR Policy.

4.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way performance measurement of an end-to-end SRv6 Policy with SRH is shown in Figure 8. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe response messages.

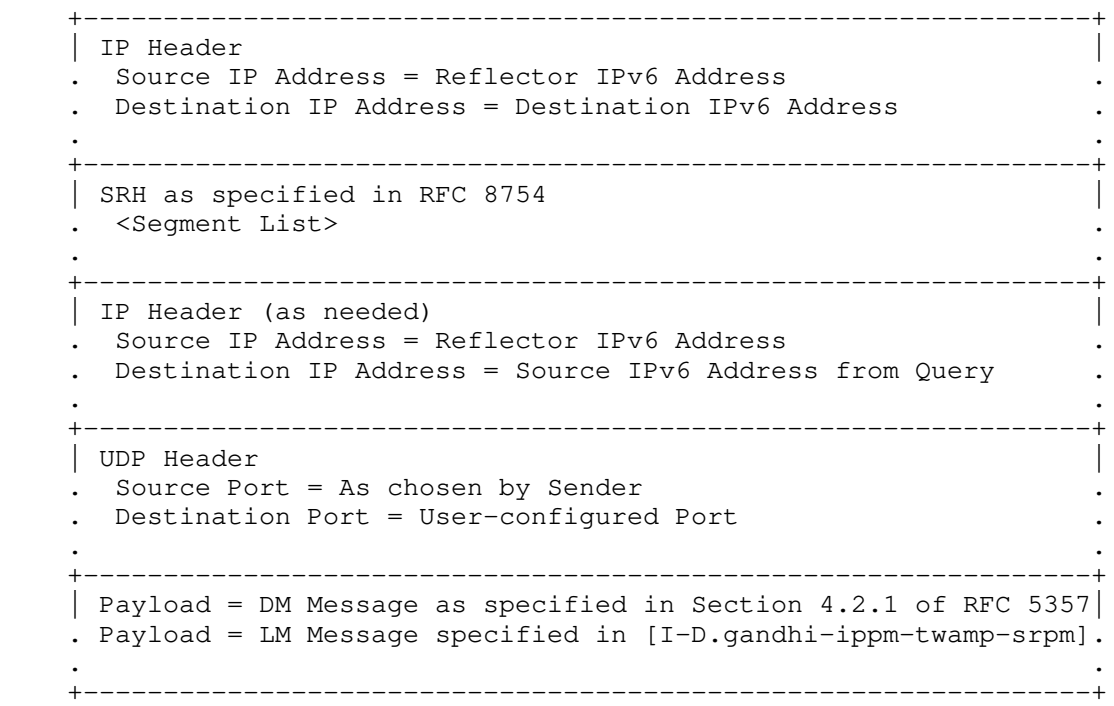


Figure 8: Example Probe Response Message for SRv6 Policy

4.2.3. Loopback Measurement Mode

The Loopback measurement mode can be used to measure round-trip delay for a bidirectional SR Path. The IP header of the probe query message contains the destination address equals to the sender address and the source address equals to the reflector address. Optionally, the probe query message can carry the reverse path information (e.g. reverse path label stack for SR-MPLS) as part of the SR header. The probe messages are not punted at the reflector node and it does not process them and generate response messages. The Sender Control Code is set to the default value of 0. In this mode, as the probe packet is not punted on the reflector node for processing, the querier copies the 'Sequence Number' in 'Session-Sender Sequence Number' directly. In this delay measurement mode, as per Reference Topology, the timestamps t1 and t4 are collected by the probes. Both these timestamps are used to measure round-trip delay as (t4 - t1).

4.3. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to TWAMP Light messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.3.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC5357]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC5357].

When using the Destination IPv4 Address from range 127/8, the TTL field in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.3.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.3.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR Path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy] or P2MP Transport [I-D.shen-spring-p2mp-transport-chain]).

The procedures for delay and loss measurement described in this document for P2P SR Policies are also equally applicable to the P2MP SR Policies. The procedure for one-way measurement is defined as following:

- The sender root node sends probe query messages using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 9.

- The probe query messages can contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].

- The Destination Address is set to the loopback address from range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 address.

- Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 9. This allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the delay and loss performance for each P2MP leaf node of the end-to-end P2MP SR Policy.

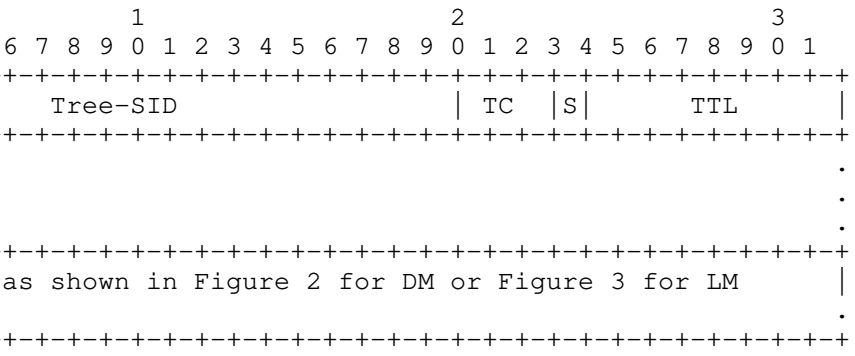


Figure 9: Example Probe Query with Tree-SID for SR-MPLS Policy

The probe query messages can also be sent using the scheme defined for P2MP Transport using Chain Replication that may contain Bud SID as defined in [I-D.shen-spring-p2mp-transport-chain].

The considerations for two-way mode for performance measurement for P2MP SR Policy (e.g. for bidirectional SR Path) are outside the scope of this document.

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the probe messages, sweeping of Destination Address from range 127/8 can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

Liveness monitoring is required for connectivity verification and continuity check in an SR network. The procedure defined in this document for delay measurement using the TWAMP Light probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that for one-way and two-way modes, the failure detection interval and scale for number of probe messages need to account for the processing of the probe query messages which need to be punted from the forwarding fast path (to slow path or control plane) and response messages need to be injected on the reflector node. This is improved by using the probes in loopback mode.

8. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far-end reflector node.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state

associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

This document does not require any IANA action.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[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>.

[I-D.gandhi-ippm-twamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "TWAMP Light Extensions for Segment Routing", draft-gandhi-ippm-twamp-srpm-00 (work in progress), October 2020.

10.2. Informative References

[IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

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

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

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

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

[RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <https://www.rfc-editor.org/info/rfc8186>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

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

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[I-D.ietf-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-00 (work in progress), July 2020.

[I-D.shen-spring-p2mp-transport-chain] Shen, Y., Zhang, Z., Parekh, R., Bidgoli, H., and Y. Kamite, "Point-to-Multipoint Transport Using Chain Replication in Segment Routing", draft-shen-spring-p2mp-transport-chain-02 (work in progress), April 2020.

[I-D.ietf-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.

[I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-24 (work in progress), October 2020.

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

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-03 (work in progress), September 2020.

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "PCEP Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-03 (work in progress), September 2020.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 20, 2021

A. Morton
AT&T Labs
R. Geib
Deutsche Telekom
L. Ciavattone
AT&T Labs

February 16, 2021

Metrics and Methods for One-way IP Capacity
draft-ietf-ippm-capacity-metric-method-06

Abstract

This memo revisits the problem of Network Capacity metrics first examined in RFC 5136. The memo specifies a more practical Maximum IP-layer Capacity metric definition catering for measurement purposes, and outlines the corresponding methods of measurement.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 20, 2021.

Copyright Notice

Table of Contents

1. Introduction

The IETF's efforts to define Network and Bulk Transport Capacity have been chartered and progressed for over twenty years. Over that time, the performance community has seen development of Informative definitions in [RFC3148] for Framework for Bulk Transport Capacity (BTC), RFC 5136 for Network Capacity and Maximum IP-layer Capacity, and the Experimental metric definitions and methods in [RFC8337], Model-Based Metrics for BTC.

This memo revisits the problem of Network Capacity metrics examined first in [RFC3148] and later in [RFC5136]. Maximum IP-Layer Capacity and [RFC3148] Bulk Transfer Capacity (goodput) are different metrics. Maximum IP-layer Capacity is like the theoretical goal for goodput. There are many metrics in [RFC5136], such as Available Capacity. Measurements depend on the network path under test and the use case. Here, the main use case is to assess the maximum capacity of the access network, with specific performance criteria used in the measurement.

This memo recognizes the importance of a definition of a Maximum IP-layer Capacity Metric at a time when access speeds have increased dramatically; a definition that is both practical and effective for the performance community's needs, including Internet users. The metric definition is intended to use Active Methods of Measurement [RFC7799], and a method of measurement is included.

The most direct active measurement of IP-layer Capacity would use IP packets, but in practice a transport header is needed to traverse address and port translators. UDP offers the most direct assessment possibility, and in the [copycat] measurement study to investigate whether UDP is viable as a general Internet transport protocol, the authors found that a high percentage of paths tested support UDP transport. A number of liaisons have been exchanged on this topic [LS-SG12-A] [LS-SG12-B], discussing the laboratory and field tests that support the UDP-based approach to IP-layer Capacity measurement.

This memo also recognizes the many updates to the IP Performance Metrics Framework [RFC2330] published over twenty years, and makes use of [RFC7312] for Advanced Stream and Sampling Framework, and [RFC8468] with IPv4, IPv6, and IPv4-IPv6 Coexistence Updates.

1.1. 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.

2. Scope and Goals

The scope of this memo is to define a metric and corresponding method to unambiguously perform Active measurements of Maximum IP-Layer Capacity, along with related metrics and methods.

The main goal is to harmonize the specified metric and method across the industry, and this memo is the vehicle that captures IETF consensus, possibly resulting in changes to the specifications of other Standards Development Organizations (SDO) (through each SDO's normal contribution process, or through liaison exchange).

A local goal is to aid efficient test procedures where possible, and to recommend reporting with additional interpretation of the results. Also, to foster the development of protocol support for this metric and method of measurement (all active testing protocols currently defined by the IPPM WG are UDP-based, meeting a key requirement of these methods). The supporting protocol development to measure this metric according to the specified method is a key future contribution to Internet measurement.

3. Motivation

As with any problem that has been worked for many years in various SDOs without any special attempts at coordination, various solutions for metrics and methods have emerged.

There are five factors that have changed (or begun to change) in the 2013-2019 time frame, and the presence of any one of them on the path requires features in the measurement design to account for the changes:

1. Internet access is no longer the bottleneck for many users.

2. Both transfer rate and latency are important to user's satisfaction.

3. UDP's growing role in Transport, in areas where TCP once dominated.

4. Content and applications moving physically closer to users.

5. Less emphasis on ISP gateway measurements, possibly due to less traffic crossing ISP gateways in future.

4. General Parameters and Definitions

This section lists the REQUIRED input factors to specify a Sender or Receiver metric.

- Src, the address of a host (such as the globally routable IP address).
- Dst, the address of a host (such as the globally routable IP address).
- MaxHops, the limit on the number of Hops a specific packet may visit as it traverses from the host at Src to the host at Dst (implemented in the TTL or Hop Limit).
- T0, the time at the start of measurement interval, when packets are first transmitted from the Source.
- I, the nominal duration of a measurement interval at the destination (default 10 sec)
- dt, the nominal duration of m equal sub-intervals in I at the destination (default 1 sec)
- dtn, a specific sub-interval, n, one of m sub-intervals in I
- Tmax, a maximum waiting time for test packets to arrive at the destination, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost), such that the distribution of one-way delay is not truncated.
- F, the number of different flows synthesized by the method (default 1 flow)
- flow, the stream of packets with the same n-tuple of designated header fields that (when held constant) result in identical treatment in a multi-path decision (such as the decision taken in load balancing). Note: The IPv6 flow label MAY be included in the flow definition when routers have complied with [RFC6438] guidelines at the Tunnel End Points (TEP), and the source of the measurement is a TEP.
- Type-P, the complete description of the test packets for which this assessment applies (including the flow-defining fields). Note that the UDP transport layer is one requirement for test packets specified below. Type-P is a parallel concept to "population of interest" defined in clause 6.1.1 of[Y.1540].

- o PM, a list of fundamental metrics, such as loss, delay, and reordering, and corresponding target performance threshold. At least one fundamental metric and target performance threshold MUST be supplied (such as One-way IP Packet Loss [RFC7680] equal to zero).

A non-Parameter which is required for several metrics is defined below:

- o T, the host time of the *first* test packet's *arrival* as measured at the destination Measurement Point, or MP(Dst). There may be other packets sent between source and destination hosts that are excluded, so this is the time of arrival of the first packet used for measurement of the metric.

Note that time stamp format and resolution, sequence numbers, etc. will be established by the chosen test protocol standard or implementation.

5. IP-Layer Capacity Singleton Metric Definitions

This section sets requirements for the following components to support the Maximum IP-layer Capacity Metric.

5.1. Formal Name

Type-P-One-way-IP-Capacity, or informally called IP-layer Capacity.

Note that Type-P depends on the chosen method.

5.2. Parameters

This section lists the REQUIRED input factors to specify the metric, beyond those listed in Section 4.

No additional Parameters are needed.

5.3. Metric Definitions

This section defines the REQUIRED aspects of the measurable IP-layer Capacity metric (unless otherwise indicated) for measurements between specified Source and Destination hosts:

Define the IP-layer capacity, C(T,dt,PM), to be the number of IP-layer bits (including header and data fields) in packets that can be transmitted from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length. The IP-layer

capacity depends on the Src and Dst hosts, the host addresses, and the path between the hosts.

The number of these IP-layer bits is designated n0[dtn,dtn+1] for a specific dt.

When the packet size is known and of fixed size, the packet count during a single sub-interval dt multiplied by the total bits in IP header and data fields is equal to n0[dtn,dtn+1].

Anticipating a Sample of Singletons, the interval dt SHOULD be set to a natural number m so that T+I = T + m*dt with dtn+1 - dtn = dt and with 1 <= n <= m.

Parameter PM represents other performance metrics [see section 5.4 below]; their measurement results SHALL be collected during measurement of IP-layer Capacity and associated with the corresponding dtn for further evaluation and reporting. Users SHALL specify the parameter Tmax as required by each metric's reference definition.

Mathematically, this definition can be represented as:

$$C(T,dt,PM) = \frac{(n0[dtn,dtn+1])}{dt}$$

Equation for IP-Layer Capacity

and:

- n0 is the total number of IP-layer header and payload bits that can be transmitted in Standard Formed packets [RFC8468] from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length, during the interval [T, T+I],

- C(T,dt,PM) the IP-Layer Capacity, corresponds to the value of n0 measured in any sub-interval ending at dtn (meaning T + n*dt), divided by the length of sub-interval, dt.

- PM represents other performance metrics [see section 5.4 below]; their measurement results SHALL be collected during measurement of IP-layer Capacity and associated with the corresponding dtn for further evaluation and reporting.

- o all sub-intervals MUST be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.

- o The bit rate of the physical interface of the measurement device must be higher than that of the link whose C(T,I,PM) is to be measured.

Measurements according to these definitions SHALL use the UDP transport layer. Standard Formed packets are specified in Section 5 of [RFC8468]. Some compression affects on measurement are discussed in Section 6 of [RFC8468], as well.

5.4. Related Round-Trip Delay and One-way Loss Definitions

RTD[dtn-1,dtn] is defined as a sample of the [RFC2681] Round-trip Delay between the Src host and the Dst host over the interval [T,T+I] (that contains equal non-overlapping intervals of dt). The "reasonable period of time" in [RFC2681] is the parameter Tmax in this memo. The statistics used to summarize RTD[dtn-1,dtn] MAY include the minimum, maximum, median, and mean, and the range = (maximum - minimum) is referred to below in Section 8.1 for load adjustment purposes.

OWL[dtn-1,dtn] is defined as a sample of the [RFC7680] One-way Loss between the Src host and the Dst host over the interval [T,T+I] (that contains equal non-overlapping intervals of dt). The statistics used to to summarize OWL[dtn-1,dtn] MAY include the lost packet count and the lost packet ratio.

Other metrics MAY be measured: one-way reordering, duplication, and delay variation.

5.5. Discussion

See the corresponding section for Maximum IP-Layer Capacity.

5.6. Reporting the Metric

The IP-Layer Capacity SHOULD be reported with at least single Megabit resolution, in units of Megabits per second (Mbps).

The Related Round Trip Delay and/or Loss metric measurements for the same Singleton SHALL be reported, also with meaningful resolution for the values measured.

Individual Capacity measurements MAY be reported in a manner consistent with the Maximum IP-Layer Capacity, see Section 9.

6. Maximum IP-Layer Capacity Metric Definitions (Statistic)

This section sets requirements for the following components to support the Maximum IP-layer Capacity Metric.

6.1. Formal Name

Type-P-One-way-Max-IP-Capacity, or informally called Maximum IP-layer Capacity.

Note that Type-P depends on the chosen method.

6.2. Parameters

This section lists the REQUIRED input factors to specify the metric, beyond those listed in Section 4.

No additional Parameters or definitions are needed.

6.3. Metric Definitions

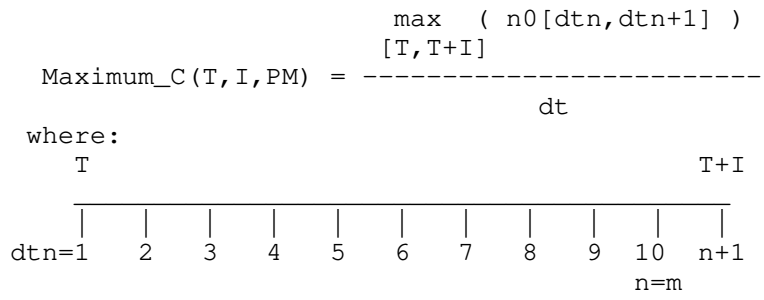
This section defines the REQUIRED aspects of the Maximum IP-layer Capacity metric (unless otherwise indicated) for measurements between specified Source and Destination hosts:

Define the Maximum IP-layer capacity, Maximum_C(T,I,PM), to be the maximum number of IP-layer bits n0[dtn,dtn+1] that can be transmitted in packets from the Src host and correctly received by the Dst host, over all dt length intervals in [T, T+I], and meeting the PM criteria. Equivalently the Maximum of a Sample of size m of C(T,I,PM) collected during the interval [T, T+I] and meeting the PM criteria.

The interval dt SHOULD be set to a natural number m so that T+I = T + m*dt with dtn+1 - dtn = dt and with 1 <= n <= m.

Parameter PM represents the other performance metrics (see Section 6.4 below) and their measurement results for the maximum IP-layer capacity. At least one target performance threshold (PM criterion) MUST be defined. If more than one metric and target performance threshold are defined, then the sub-interval with maximum number of bits transmitted MUST meet all the target performance thresholds. Users SHALL specify the parameter Tmax as required by each metric's reference definition.

Mathematically, this definition can be represented as:



Equation for Maximum Capacity

and:

- n0 is the total number of IP-layer header and payload bits that can be transmitted in Standard Formed packets from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length, during the interval [T, T+I],

- Maximum_C(T,I,PM) the Maximum IP-Layer Capacity, corresponds to the maximum value of n0 measured in any sub-interval ending at dtn (meaning T + n*dt), divided by the constant length of all sub-intervals, dt.

- PM represents the other performance metrics (see Section 5.4) and their measurement results for the maximum IP-layer capacity. At least one target performance threshold (PM criterion) MUST be defined.

- all sub-intervals MUST be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.

- The bit rate of the physical interface of the measurement systems must be higher than that of the link whose Maximum_C(T,I,PM) is to be measured (the bottleneck link).

In this definition, the m sub-intervals can be viewed as trials when the Src host varies the transmitted packet rate, searching for the maximum n0 that meets the PM criteria measured at the Dst host in a test of duration, I. When the transmitted packet rate is held constant at the Src host, the m sub-intervals may also be viewed as trials to evaluate the stability of n0 and metric(s) in the PM list over all dt-length intervals in I.

Measurements according to these definitions SHALL use the UDP transport layer.

6.4. Related Round-Trip Delay and One-way Loss Definitions

RTD[dtn,dtn+1] and OWL[dtn,dtn+1] are defined in Section 5.4. Here, the test intervals are increased to match the capacity samples, RTD[T,I] and OWL[T,I].

The interval dtn,dtn+1 where Maximum_C[T,I,PM] occurs is the reporting sub-interval for RTD[T,I] and OWL[T,I].

Other metrics MAY be measured: one-way reordering, duplication, and delay variation.

6.5. Discussion

If traffic conditioning (e.g., shaping, policing) applies along a path for which Maximum_C(T,I,PM) is to be determined, different values for dt SHOULD be picked and measurements be executed during multiple intervals [T, T+I]. A single constant interval dt SHOULD be chosen so that is an integer multiple of increasing values k times serialisation delay of a path MTU at the physical interface speed where traffic conditioning is expected. This should avoid taking configured burst tolerance singletons as a valid Maximum_C(T,I,PM) result.

A Maximum_C(T,I,PM) without any indication of bottleneck congestion, be that an increasing latency, packet loss or ECN marks during a measurement interval I, is likely to underestimate Maximum_C(T,I,PM).

6.6. Reporting the Metric

The IP-Layer Capacity SHOULD be reported with at least single Megabit resolution, in units of Megabits per second (Mbps) (which is 1,000,000 bits per second to avoid any confusion).

The Related Round Trip Delay and/or Loss metric measurements for the same Singleton SHALL be reported, also with meaningful resolution for the values measured.

When there are demonstrated and repeatable Capacity modes in the Sample, then the Maximum IP-Layer Capacity SHALL be reported for each mode, along with the relative time from the beginning of the stream that the mode was observed to be present. Bimodal Maxima have been observed with some services, sometimes called a "turbo mode" intending to deliver short transfers more quickly, or reduce the

initial buffering time for some video streams. Note that modes lasting less than dt duration will not be detected.

Some transmission technologies have multiple methods of operation that may be activated when channel conditions degrade or improve, and these transmission methods may determine the Maximum IP-Layer Capacity. Examples include line-of-sight microwave modulator constellations, or cellular modem technologies where the changes may be initiated by a user moving from one coverage area to another. Operation in the different transmission methods may be observed over time, but the modes of Maximum IP-Layer Capacity will not be activated deterministically as with the "turbo mode" described in the paragraph above.

7. IP-Layer Sender Bit Rate Singleton Metric Definitions

This section sets requirements for the following components to support the IP-layer Sender Bitrate Metric. This metric helps to check that the sender actually generated the desired rates during a test, and measurement takes place at the Src host to network path interface (or as close as practical). It is not a metric for path performance.

7.1. Formal Name

Type-P-IP-Sender-Bit-Rate, or informally called IP-layer Sender Bitrate.

Note that Type-P depends on the chosen method.

7.2. Parameters

This section lists the REQUIRED input factors to specify the metric, beyond those listed in Section 4.

- S, the duration of the measurement interval at the Source
- st, the nominal duration of N sub-intervals in S (default = 0.05 seconds)

S SHALL be longer than I, primarily to account for on-demand activation of the path, or any preamble to testing required, and the delay of the path.

st SHOULD be much smaller than the sub-interval dt. The st parameter does not have relevance when the Source is transmitting at a fixed rate throughout S.

7.3. Metric Definition

This section defines the REQUIRED aspects of the IP-layer Sender Bitrate metric (unless otherwise indicated) for measurements at the specified Source on packets addressed for the intended Destination host and matching the required Type-P:

Define the IP-layer Sender Bit Rate, B(S,st), to be the number of IP-layer bits (including header and data fields) that are transmitted from the Source during one contiguous sub-interval, st, during the test interval S (where S SHALL be longer than I), and where the fixed-size packet count during that single sub-interval st also provides the number of IP-layer bits in any interval: n0[stn-1,stn].

Measurements according to these definitions SHALL use the UDP transport layer. Any feedback from Dst host to Src host received by Src host during an interval [stn-1,stn] MUST NOT result in an adaptation of the Src host traffic conditioning during this interval (rate adjustment occurs on st boundaries).

7.4. Discussion

Both the Sender and Receiver or (source and destination) bit rates SHOULD be assessed as part of an IP-layer Capacity measurement.

7.5. Reporting the Metric

The IP-Layer Sender Bit Rate SHALL be reported with meaningful resolution, in units of Megabits per second.

Individual IP-Layer Sender Bit Rate measurements are discussed further in Section 9.

8. Method of Measurement

The architecture of the method REQUIRES two cooperating hosts operating in the roles of Src (test packet sender) and Dst (receiver), with a measured path and return path between them.

The duration of a test, parameter I, MUST be constrained in a production network, since this is an active test method and it will likely cause congestion on the Src to Dst host path during a test.

8.1. Load Rate Adjustment Algorithm

A table SHALL be pre-built defining all the offered load rates that will be supported (R1 through Rn, in ascending order, corresponding to indexed rows in the table). Each rate is defined as datagrams of

size ss, sent as a burst of count cc, every time interval tt (default for tt is 1ms, a likely system tick-interval). While it is advantageous to use datagrams of as large a size as possible, it may be prudent to use a slightly smaller maximum that allows for secondary protocol headers and/or tunneling without resulting in IP-layer fragmentation. Selection of a new rate is indicated by a calculation on the current row, Rx. For example:

"Rx+1": the sender uses the next higher rate in the table.

"Rx-10": the sender uses the rate 10 rows lower in the table.

At the beginning of a test, the sender begins sending at rate R1 and the receiver starts a feedback timer at interval F (while awaiting inbound datagrams). As datagrams are received they are checked for sequence number anomalies (loss, out-of-order, duplication, etc.) and the delay range is measured (one-way or round-trip). This information is accumulated until the feedback timer F expires and a status feedback message is sent from the receiver back to the sender, to communicate this information. The accumulated statistics are then reset by the receiver for the next feedback interval. As feedback messages are received back at the sender, they are evaluated to determine how to adjust the current offered load rate (Rx).

If the feedback indicates that no sequence number anomalies were detected AND the delay range was below the lower threshold, the offered load rate is increased. If congestion has not been confirmed up to this point, the offered load rate is increased by more than one rate (e.g., Rx+10). This allows the offered load to quickly reach a near-maximum rate. Conversely, if congestion has been previously confirmed, the offered load rate is only increased by one (Rx+1).

If the feedback indicates that sequence number anomalies were detected OR the delay range was above the upper threshold, the offered load rate is decreased. Also, if congestion is now confirmed by the current feedback message being processed, then the offered load rate is decreased by more than one rate (e.g., Rx-30). This one-time reduction is intended to compensate for the fast initial ramp-up. In all other cases, the offered load rate is only decreased by one (Rx-1).

If the feedback indicates that there were no sequence number anomalies AND the delay range was above the lower threshold, but below the upper threshold, the offered load rate is not changed. This allows time for recent changes in the offered load rate to stabilize, and the feedback to represent current conditions more accurately.

Lastly, the method for inferring congestion is that there were sequence number anomalies AND/OR the delay range was above the upper threshold for two consecutive feedback intervals. The algorithm described above is also illustrated in ITU-T Rec. Y.1540, 2020 version[Y.1540], in Annex B, and implemented in the Appendix on Load Rate Adjustment Pseudo Code in this memo.

All the values used above are relevant to searches in the 1 Mbps to 10 Gbps capacity range. Searches can accommodate higher rate capacities by changing the rates in the pre-built table.

8.2. Measurement Qualification or Verification

It is of course necessary to calibrate the equipment performing the IP-layer Capacity measurement, to ensure that the expected capacity can be measured accurately, and that equipment choices (processing speed, interface bandwidth, etc.) are suitably matched to the measurement range.

When assessing a Maximum rate as the metric specifies, artificially high (optimistic) values might be measured until some buffer on the path is filled. Other causes include bursts of back-to-back packets with idle intervals delivered by a path, while the measurement interval (dt) is small and aligned with the bursts. The artificial values might result in an un-sustainable Maximum Capacity observed when the method of measurement is searching for the Maximum, and that would not do. This situation is different from the bi-modal service rates (discussed under Reporting), which are characterized by a multi-second duration (much longer than the measured RTT) and repeatable behavior.

There are many ways that the Method of Measurement could handle this false-max issue. The default value for measurement of singletons (dt = 1 second) has proven to a be of practical value during tests of this method, allows the bimodal service rates to be characterized, and it has an obvious alignment with the reporting units (Mbps).

Another approach comes from Section 24 of RFC 2544[RFC2544] and its discussion of Trial duration, where relatively short trials conducted as part of the search are followed by longer trials to make the final determination. In the production network, measurements of singletons and samples (the terms for trials and tests of Lab Benchmarking) must be limited in duration because they may be service-affecting. But there is sufficient value in repeating a sample with a fixed sending rate determined by the previous search for the Max IP-layer Capacity, to qualify the result in terms of the other performance metrics measured at the same time.

A qualification measurement for the search result is a subsequent measurement, sending at a fixed 99.x % of the Max IP-layer Capacity for I, or an indefinite period. The same Max Capacity Metric is applied, and the Qualification for the result is a sample without packet loss or a growing minimum delay trend in subsequent singletons (or each dt of the measurement interval, I). Samples exhibiting losses or increasing queue occupation require a repeated search and/ or test at reduced fixed sender rate for qualification.

Here, as with any Active Capacity test, the test duration must be kept short. 10 second tests for each direction of transmission are common today. The default measurement interval specified here is I = 10 seconds). In combination with a fast search method and user-network coordination, the concerns raised in RFC 6815[RFC6815] are alleviated. The method for assessing Max IP Capacity is different from classic [RFC2544] methods: they use short term load adjustment and are sensitive to loss and delay, like other congestion control algorithms used on the Internet every day.

8.3. Measurement Considerations

In general, the wide-spread measurements that this memo encourages will encounter wide-spread behaviors. The bimodal IP Capacity behaviors already discussed in Section 6.6 are good examples.

In general, it is RECOMMENDED to locate test endpoints as close to the intended measured link(s) as practical (this is not always possible for reasons of scale; there is a limit on number of test endpoints coming from many perspectives, management and measurement traffic for example). The testing operator MUST set a value for the MaxHops parameter, based on the expected path length. This parameter can keep measurement traffic from straying too far beyond the intended path.

The path measured may be state-full based on many factors, and the Parameter "Time of day" when a test starts may not be enough information. Repeatable testing may require the time from the beginning of a measured flow, and how the flow is constructed including how much traffic has already been sent on that flow when a state-change is observed, because the state-change may be based on time or bytes sent or both.

Many different traffic shapers and on-demand access technologies may be encountered, as anticipated in [RFC7312], and play a key role in measurement results. Methods MUST be prepared to provide a short preamble transmission to activate on-demand access, and to discard the preamble from subsequent test results.

Conditions which might be encountered during measurement, where packet losses may occur independently from the measurement sending rate:

1. Congestion of an interconnection or backbone interface may appear as packet losses distributed over time in the test stream, due to much higher rate interfaces in the backbone.

2. Packet loss due to use of Random Early Detection (RED) or other active queue management may or may not affect the measurement flow if competing background traffic (other flows) are simultaneously present.

3. There may be only small delay variation independent of sending rate under these conditions, too.

4. Persistent competing traffic on measurement paths that include shared transmission media may cause random packet losses in the test stream.

It is possible to mitigate these conditions using the flexibility of the load-rate adjusting algorithm described in Section 8.1 above (tuning specific parameters).

If the measurement flow burst duration happens to be on the order of or smaller than the burst size of a shaper or a policer in the path, then the line rate might be measured rather than the bandwidth limit imposed by the shaper or policer. If this condition is suspected, alternate configurations SHOULD be used.

In general, results depend on the sending stream characteristics; the measurement community has known this for a long time, and needs to keep it front of mind. Although the default is a single flow (F=1) for testing, use of multiple flows may be advantageous for the following reasons:

1. the test hosts may be able to create higher load than with a single flow, or parallel test hosts may be used to generate 1 flow each.

2. there may be link aggregation present (flow-based load balancing) and multiple flows are needed to occupy each member of the aggregate.

3. access policies may limit the IP-Layer Capacity depending on the Type-P of packets, possibly reserving capacity for various stream types.

Each flow would be controlled using its own implementation of the Load Adjustment (Search) Algorithm.

As testing continues, implementers should expect some evolution in the methods. The ITU-T has published a Supplement (60) to the Y-series of Recommendations, "Interpreting ITU-T Y.1540 maximum IP-layer capacity measurements", [Y.Sup60], which is the result of continued testing with the metric and method described here.

8.4. Running Code

This section is for the benefit of the Document Shepherd's form, and will be deleted prior to final review.

Much of the development of the method and comparisons with existing methods conducted at IETF Hackathons and elsewhere have been based on the example udpst Linux measurement tool (which is a working reference for further development) [udpst]. The current project:

- is a utility that can function as a client or server daemon
- requires a successful client-initiated setup handshake between cooperating hosts and allows firewalls to control inbound unsolicited UDP which either go to a control port [expected and w/ authentication] or to ephemeral ports that are only created as needed. Firewalls protecting each host can both continue to do their job normally. This aspect is similar to many other test utilities available.
- is written in C, and built with gcc (release 9.3) and its standard run-time libraries
- allows configuration of most of the parameters described in Sections 4 and 7.
- supports IPv4 and IPv6 address families.
- supports IP-layer packet marking.

9. Reporting Formats

The singleton IP-Layer Capacity results SHOULD be accompanied by the context under which they were measured.

- timestamp (especially the time when the maximum was observed in dtn)
- source and destination (by IP or other meaningful ID)

- other inner parameters of the test case (Section 4)
- outer parameters, such as "test conducted in motion" or other factors belonging to the context of the measurement
- result validity (indicating cases where the process was somehow interrupted or the attempt failed)
- a field where unusual circumstances could be documented, and another one for "ignore/mask out" purposes in further processing

The Maximum IP-Layer Capacity results SHOULD be reported in the format of a table with a row for each of the test Phases and Number of Flows. There SHOULD be columns for the phases with number of flows, and for the resultant Maximum IP-Layer Capacity results for the aggregate and each flow tested.

As mentioned in Section 6.6, bi-modal (or multi-modal) maxima SHALL be reported for each mode separately.

Phase, # Flows	Max IP-Layer Capacity, Mbps	Loss Ratio	RTT min, max, msec
Search,1	967.31	0.0002	30, 58
Verify,1	966.00	0.0000	30, 38

Maximum IP-layer Capacity Results

Static and configuration parameters:

The sub-interval time, dt, MUST accompany a report of Maximum IP-Layer Capacity results, and the remaining Parameters from Section 4, General Parameters.

The PM list metrics corresponding to the sub-interval where the Maximum Capacity occurred MUST accompany a report of Maximum IP-Layer Capacity results, for each test phase.

The IP-Layer Sender Bit rate results SHOULD be reported in the format of a table with a row for each of the test Phases, sub-intervals (st) and Number of Flows. There SHOULD be columns for the phases with number of flows, and for the resultant IP-Layer Sender Bit rate results for the aggregate and each flow tested.

Phase, Flow or Aggregate	st, sec	Sender Bit Rate, Mbps	??
Search,1	0.00 - 0.05	345	__
Search,2	0.00 - 0.05	289	__
Search,Agg	0.00 - 0.05	634	__

IP-layer Sender Bit Rate Results

Static and configuration parameters:

The subinterval time, st, MUST accompany a report of Sender IP-Layer Bit Rate results.

Also, the values of the remaining Parameters from Section 4, General Parameters, MUST be reported.

9.1. Configuration and Reporting Data Formats

As a part of the multi-Standards Development Organization (SDO) harmonization of this metric and method of measurement, one of the areas where the Broadband Forum (BBF) contributed its expertise was in the definition of an information model and data model for configuration and reporting. These models are consistent with the metric parameters and default values specified as lists is this memo. [TR-471] provides the Information model that was used to prepare a full data model in related BBF work. The BBF has also carefully considered topics within its purview, such as placement of measurement systems within the access architecture. For example, timestamp resolution requirements that influence the choice of the test protocol are provided in Table 2 of [TR-471].

10. Security Considerations

Active metrics and measurements have a long history of security considerations. The security considerations that apply to any active measurement of live paths are relevant here. See [RFC4656] and [RFC5357].

When considering privacy of those involved in measurement or those whose traffic is measured, the sensitive information available to potential observers is greatly reduced when using active techniques which are within this scope of work. Passive observations of user traffic for measurement purposes raise many privacy issues. We refer the reader to the privacy considerations described in the Large Scale

Measurement of Broadband Performance (LMAP) Framework [RFC7594], which covers active and passive techniques.

There are some new considerations for Capacity measurement as described in this memo.

1. Cooperating source and destination hosts and agreements to test the path between the hosts are REQUIRED. Hosts perform in either the Src or Dst roles.

2. A REQUIRED user client-initiated setup handshake between cooperating hosts and allows firewalls to control inbound unsolicited UDP which either go to a control port [expected and w/authentication] or to ephemeral ports that are only created as needed. Firewalls protecting each host can both continue to do their job normally.

3. Integrity protection for feedback messages conveying measurements is RECOMMENDED.

4. Hosts MUST limit the number of simultaneous tests to avoid resource exhaustion and inaccurate results.

5. Senders MUST be rate-limited. This can be accomplished using the pre-built table defining all the offered load rates that will be supported (Section 8.1). The recommended load-control search algorithm results in "ramp up" from the lowest rate in the table.

6. Service subscribers with limited data volumes who conduct extensive capacity testing might experience the effects of Service Provider controls on their service. Testing with the Service Provider's measurement hosts SHOULD be limited in frequency and/or overall volume of test traffic.

The exact specification of these features is left for the future protocol development.

11. IANA Considerations

This memo makes no requests of IANA.

12. Acknowledgments

Thanks to Joachim Fabini, Matt Mathis, J.Ignacio Alvarez-Hamelin, Wolfgang Balzer, Frank Brockners, Greg Mirsky and Martin Duke for their extensive comments on the memo and related topics.

13. Appendix - Load Rate Adjustment Pseudo Code

The following is a pseudo-code implementation of the algorithm described in Section 8.1.

```
Rx = 1  # The current sending rate (equivalent to a row of the table)
seqErr = 0  # Measured count of any of Loss or Reordering impairments
delay = 0 # Measured Range of Round Trip Time, RTT, ms
lowThresh = 30 # Low threshold on the Range of RTT, ms
upperThresh = 90 # Upper threshold on the Range of RTT, ms
hSpeedTresh = 1Gbps # Threshold for transition between sending rate step
 sizes (such as 1 Mbps and 100 Mbps)
slowAdjCount = 0 # Measured Number of consecutive status reports
 indicating loss and/or delay variation above upperThresh
slowAdjThresh = 2 # Threshold on slowAdjCount used to infer congestion.
 Use values >1 to avoid misinterpreting transient loss
highSpeedDelta = 10 # The number of rows to move in a single adjustment
 when initially increasing offered load (to ramp-up quickly)
maxLoadRates = 2000 # Maximum table index (rows)


if ( seqErr == 0 && delay < lowThresh ) {
        if ( Rx < hSpeedTresh && slowAdjCount < slowAdjThresh ) {
                        Rx += highSpeedDelta;
                        slowAdjCount = 0;
        } else {
                        if ( Rx < maxLoadRates - 1 )
                                        Rx++;
        }
} else if ( seqErr > 0 || delay > upperThresh ) {
        slowAdjCount++;
        if ( Rx < hSpeedTresh && c->slowAdjCount == slowAdjThresh ) {
                        if ( Rx > highSpeedDelta * 3 )
                                        Rx -= highSpeedDelta * 3;
                        else
                                        Rx = 0;
        } else {
                        if ( Rx > 0 )
                                        Rx--;
        }
}
```

14. References

14.1. Normative References

[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>.

[RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <https://www.rfc-editor.org/info/rfc2330>.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <https://www.rfc-editor.org/info/rfc2681>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[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>.

[RFC8468] Morton, A., Fabini, J., Elkins, N., Ackermann, M., and V. Hegde, "IPv4, IPv6, and IPv4-IPv6 Coexistence: Updates for the IP Performance Metrics (IPPM) Framework", RFC 8468, DOI 10.17487/RFC8468, November 2018, <https://www.rfc-editor.org/info/rfc8468>.

14.2. Informative References

[copycat] Edleine, K., Kuhlewind, K., Trammell, B., and B. Donnet, "copycat: Testing Differential Treatment of New Transport Protocols in the Wild (ANRW '17)", July 2017, <https://irtf.org/anrw/2017/anrw17-final5.pdf>.

[LS-SG12-A] 12, I. S., "LS - Harmonization of IP Capacity and Latency Parameters: Revision of Draft Rec. Y.1540 on IP packet transfer performance parameters and New Annex A with Lab Evaluation Plan", May 2019, <https://datatracker.ietf.org/liaison/1632/>.

[LS-SG12-B] 12, I. S., "LS on harmonization of IP Capacity and Latency Parameters: Consent of Draft Rec. Y.1540 on IP packet transfer performance parameters and New Annex A with Lab & Field Evaluation Plans", March 2019, <https://datatracker.ietf.org/liaison/1645/>.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

[RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, DOI 10.17487/RFC3148, July 2001, <https://www.rfc-editor.org/info/rfc3148>.

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <https://www.rfc-editor.org/info/rfc5136>.

[RFC6815] Bradner, S., Dubray, K., McQuaid, J., and A. Morton, "Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful", RFC 6815, DOI 10.17487/RFC6815, November 2012, <https://www.rfc-editor.org/info/rfc6815>.

[RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <https://www.rfc-editor.org/info/rfc7312>.

[RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <https://www.rfc-editor.org/info/rfc7594>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8337] Mathis, M. and A. Morton, "Model-Based Metrics for Bulk Transport Capacity", RFC 8337, DOI 10.17487/RFC8337, March 2018, <https://www.rfc-editor.org/info/rfc8337>.

[TR-471] Morton, A., "Broadband Forum TR-471: IP Layer Capacity Metrics and Measurement", July 2020, <https://www.broadband-forum.org/technical/download/TR-471.pdf>.

[udpst] udpst Project Collaborators, "UDP Speed Test Open Broadband project", December 2020, <https://github.com/BroadbandForum/obudpst>.

[Y.1540] Y.1540, I. R., "Internet protocol data communication service - IP packet transfer and availability performance parameters", December 2019, <https://www.itu.int/rec/T-REC-Y.1540-201912-I/en>.

[Y.Sup60] Morton, A., "Recommendation Y.Sup60, (04/20) Interpreting ITU-T Y.1540 maximum IP-layer capacity measurements", June 2020, <https://www.itu.int/rec/T-REC-Y.Sup60/en>.

Authors' Addresses

Al Morton
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Phone: +1 732 420 1571
Fax: +1 732 368 1192
Email: acm@research.att.com

Ruediger Geib
Deutsche Telekom
Heinrich Hertz Str. 3-7
Darmstadt 64295
Germany

Phone: +49 6151 5812747
Email: Ruediger.Geib@telekom.de

Len Ciavattone
AT&T Labs
200 Laurel Avenue South
Middletown,, NJ 07748
USA

Email: lencia@att.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2021

H. Song
Futurewei
B. Gafni
Nvidia
T. Zhou
Z. Li
Huawei
F. Brockners
Cisco
S. Bhandari, Ed.
Thoughtspot
R. Sivakolundu
Cisco
T. Mizrahi, Ed.
Huawei

February 17, 2021

In-situ OAM Direct Exporting
draft-ietf-ippm-ioam-direct-export-03

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) is used for recording and collecting operational and telemetry information. Specifically, IOAM allows telemetry data to be pushed into data packets while they traverse the network. This document introduces a new IOAM option type called the Direct Export (DEX) option, which is used as a trigger for IOAM data to be directly exported without being pushed into in-flight data packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 21, 2021.

Table of Contents

1. Introduction

IOAM [I-D.ietf-ippm-ioam-data] is used for monitoring traffic in the network, and for incorporating IOAM data fields into in-flight data packets.

IOAM makes use of four possible IOAM options, defined in [I-D.ietf-ippm-ioam-data]: Pre-allocated Trace Option, Incremental Trace Option, Proof of Transit (POT) Option, and Edge-to-Edge Option.

This document defines a new IOAM option type (also known as an IOAM type) called the Direct Export (DEX) option. This option is used as a trigger for IOAM nodes to export IOAM data to a receiving entity

(or entities). A "receiving entity" in this context can be, for example, an external collector, analyzer, controller, decapsulating node, or a software module in one of the IOAM nodes.

This draft has evolved from combining some of the concepts of PBT-I from [I-D.song-ippm-postcard-based-telemetry] with immediate exporting from [I-D.ietf-ippm-ioam-flags].

2. Conventions

2.1. Requirement 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.

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

DEX: Direct EXporting

3. The Direct Exporting (DEX) IOAM Option Type

3.1. Overview

The DEX option is used as a trigger for exporting telemetry data to a receiving entity (or entities).

This option is incorporated into data packets by an IOAM encapsulating node, and removed by an IOAM decapsulating node, as illustrated in Figure 1. The option can be read but not modified by transit nodes. Note: the terms IOAM encapsulating, decapsulating and transit nodes are as defined in [I-D.ietf-ippm-ioam-data].

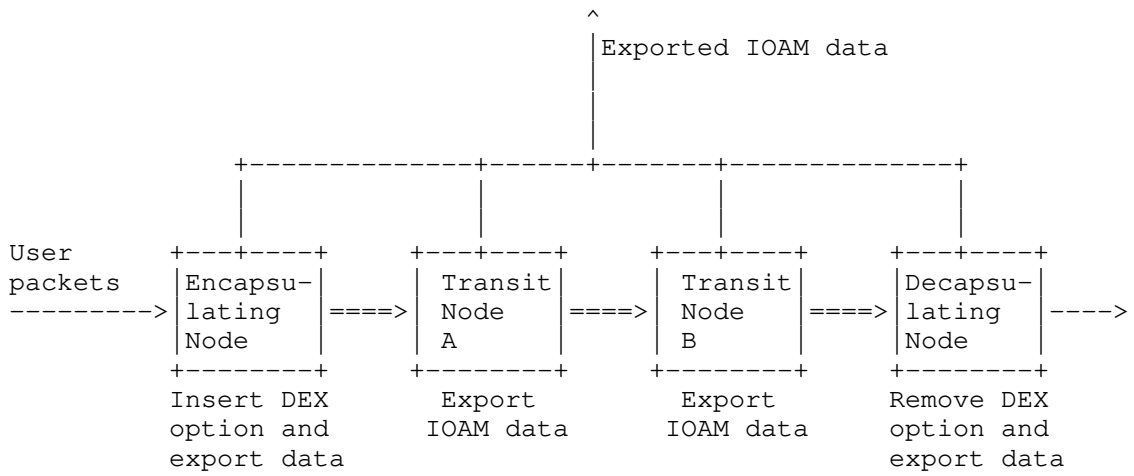


Figure 1: DEX Architecture

The DEX option is used as a trigger to export IOAM data. The trigger applies to transit nodes, the decapsulating node, and the encapsulating node:

- An IOAM encapsulating node configured to incorporate the DEX option encapsulates the packet with the DEX option, and MAY export the requested IOAM data immediately. The IOAM encapsulating node is the only type of node allowed to push the DEX option.

- A transit node that processes a packet with the DEX option MAY export the requested IOAM data.

- An IOAM decapsulating node that processes a packet with the DEX option MAY export the requested IOAM data, and MUST decapsulate the IOAM header.

As in [I-D.ietf-ippm-ioam-data], the DEX option may be incorporated into all or a subset of the traffic that is forwarded by the encapsulating node. Moreover, IOAM nodes MAY export data for all traversing packets that carry the DEX option, or MAY selectively export data only for a subset of these packets.

The DEX option specifies which data fields should be exported, as specified in Section 3.2. The format and encapsulation of the packet that contains the exported data is not within the scope of the current document. For example, the export format can be based on [I-D.spiegel-ippm-ioam-rawexport].

A transit IOAM node that does not support the DEX option SHOULD ignore it. A decapsulating node that does not support the DEX option MUST remove it, along with any other IOAM options carried in the packet if such exist.

3.2. The DEX Option Format

The format of the DEX option is depicted in Figure 2. The length of the DEX option is either 8 octets or 16 octets, as the Flow ID and the Sequence Number fields (summing up to 8 octets) are optional. It is assumed that the lower layer protocol indicates the length of the DEX option, thus indicating whether the two optional fields are present.

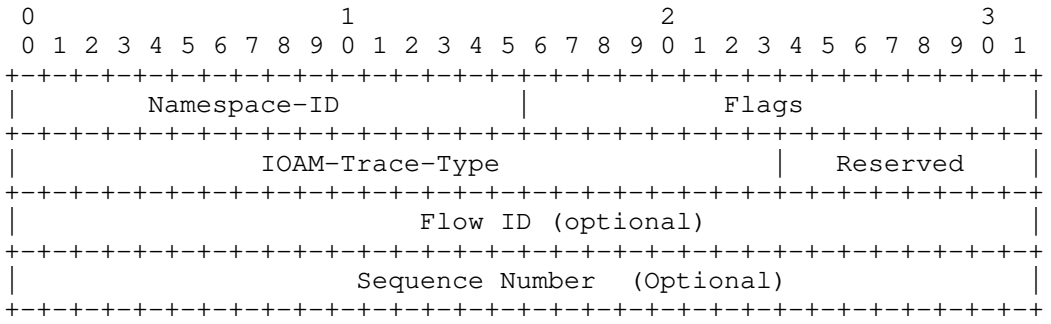


Figure 2: DEX Option Format

Namespace-ID A 16-bit identifier of the IOAM namespace, as defined in [I-D.ietf-ippm-ioam-data].

Flags A 16-bit field, comprised of 16 one-bit subfields. Flags are allocated by IANA, as defined in Section 4.2.

IOAM-Trace-Type A 24-bit identifier which specifies which data fields should be exported. The format of this field is as defined in [I-D.ietf-ippm-ioam-data]. Specifically, bit 23, which corresponds to the Checksum Complement data field, should be assigned to be zero by the IOAM encapsulating node, and ignored by transit and decapsulating nodes. The reason for this is that the Checksum Complement is intended for in-flight packet modifications and is not relevant for direct exporting.

Reserved	This field SHOULD be ignored by the receiver.
Flow ID	A 32-bit flow identifier. If the actual Flow ID is shorter than 32 bits, it is zero padded in its most significant bits. The field is set at the encapsulating node. The Flow ID can be uniformly assigned by a central controller or algorithmically generated by the encapsulating node. The latter approach cannot guarantee the uniqueness of Flow ID, yet the conflict probability is small due to the large Flow ID space. The Flow ID can be used to correlate the exported data of the same flow from multiple nodes and from multiple packets.
Sequence Number	A 32-bit sequence number starting from 0 and increasing by 1 for each following monitored packet from the same flow at the encapsulating node. The Sequence Number, when combined with the Flow ID, provides a convenient approach to correlate the exported data from the same user packet.

4. IANA Considerations

4.1. IOAM Type

The "IOAM Type Registry" was defined in Section 7.2 of [I-D.ietf-ippm-ioam-data]. IANA is requested to allocate the following code point from the "IOAM Type Registry" as follows:

TBD-type	IOAM Direct Export (DEX) Option Type

If possible, IANA is requested to allocate code point 4 (TBD-type).

4.2. IOAM DEX Flags

IANA is requested to define an "IOAM DEX Flags" registry. This registry includes 16 flag bits. Allocation should be performed based on the "RFC Required" procedure, as defined in [RFC8126].

5. Performance Considerations

The DEX option triggers exported packets to be exported to a receiving entity (or entities). In some cases this may impact the receiving entity's performance, or the performance along the paths leading to it.

Therefore, rate limiting may be enabled so as to ensure that direct exporting is used at a rate that does not significantly affect the

network bandwidth, and does not overload the receiving entity (or the source node in the case of loopback). It should be possible to use each DEX on a subset of the data traffic, and to load balance the exported data among multiple receiving entities.

6. Security Considerations

The security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data]. Specifically, an attacker may try to use the functionality that is defined in this document to attack the network.

An attacker may attempt to overload network devices by injecting synthetic packets that include the DEX option. Similarly, an on-path attacker may maliciously incorporate the DEX option into transit packets, or maliciously remove it from packets in which it is incorporated.

Forcing DEX, either in synthetic packets or in transit packets may overload the receiving entity (or entities). Since this mechanism affects multiple devices along the network path, it potentially amplifies the effect on the network bandwidth and on the receiving entity's load.

The amplification effect of DEX may be worse in wide area networks in which there are multiple IOAM domains. For example, if DEX is used in IOAM domain 1 for exporting IOAM data to a receiving entity, then the exported packets of domain 1 can be forwarded through IOAM domain 2, in which they are subject to DEX. The exported packets of domain 2 may in turn be forwarded through another IOAM domain (or through domain 1), and theoretically this recursive amplification may continue infinitely.

In order to mitigate the attacks described above, it should be possible for IOAM-enabled devices to limit the exported IOAM data to a configurable rate.

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their affect. This is a fundamental assumption with respect to the security aspects of IOAM, as further discussed in [I-D.ietf-ippm-ioam-data].

7. References

7.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[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>.

[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>.

7.2. Informative References

[I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-ietf-ippm-ioam-flags-03 (work in progress), October 2020.

[I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

[I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-04 (work in progress), November 2020.

[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>.

Appendix A. Hop Limit and Hop Count in Direct Exporting

In order to help correlate and order the exported packets, it is possible to include the Hop_Lim/Node_ID data field in exported packets; if the IOAM-Trace-Type [I-D.ietf-ippm-ioam-data] has the Hop_Lim/Node_ID bit set, then exported packets include the Hop_Lim/

Node_ID data field, which contains the TTL/Hop Limit value from a lower layer protocol.

An alternative approach was considered during the design of this document, according to which a 1-octet Hop Count field would be included in the DEX header (presumably by claiming some space from the Flags field). The Hop Limit would starts from 0 at the encapsulating node and be incremented by each IOAM transit node that supports the DEX option. In this approach the Hop Count field value would also be included in the exported packet.

The main advantage of the Hop_Lim/Node_ID approach is that it provides information about the current hop count without requiring each transit node to modify the DEX option, thus simplifying the data plane functionality of Direct Exporting. The main advantage of the Hop Count approach that was considered is that it counts the number of IOAM-capable nodes without relying on the lower layer TTL, especially when the lower layer cannot prvide the accurate TTL information, e.g., Layer 2 Ethernet or hierarchical VPN. The Hop Count approach would also explicitly allow to detect a case where an IOAM-capable node fails to export packets. It would also be possible to use a flag to indicate an optional Hop Count field, which enables to control the tradeoff. On one hand it would address the use cases that the Hop_Lim/Node_ID cannot cover, and on the other hand it would not require transit switches to update the option if it was not supported or disabled. For the sake of simplicity the Hop Count approach was not pursued, and this field is not incorporated in the DEX header.

Authors' Addresses

Haoyu Song
Futurewei
2330 Central Expressway
Santa Clara 95050
USA

Email: haoyu.song@huawei.com

Barak Gafni
Nvidia
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@nvidia.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Zhenbin Li
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari (editor)
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
U.S.A.

Email: sramesh@cisco.com

Tal Mizrahi (editor)
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2021

T. Mizrahi
Huawei
F. Brockners
Cisco
S. Bhandari, Ed.
Thoughtspot
R. Sivakolundu
C. Pignataro
Cisco
A. Kfir
B. Gafni
Nvidia
M. Spiegel
Barefoot Networks
J. Lemon
Broadcom
February 17, 2021

In-situ OAM Flags
draft-ietf-ippm-ioam-flags-04

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document presents new flags in the IOAM Trace Option headers. Specifically, the document defines the Loopback and Active flags.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 21, 2021.

Table of Contents

1. Introduction

IOAM [I-D.ietf-ippm-ioam-data] is used for monitoring traffic in the network by incorporating IOAM data fields into in-flight data packets.

IOAM data may be represented in one of four possible IOAM options: Pre-allocated Trace Option, Incremental Trace Option, Proof of Transit (POT) Option, and Edge-to-Edge Option. This document defines two new flags in the Pre-allocated and Incremental Trace options: the Loopback and Active flags.

2. Conventions

2.1. 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.

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

3. New IOAM Trace Option Flags

This document defines two new flags in the Pre-allocated and Incremental Trace options:

Bit 1 "Loopback" (L-bit). Loopback mode is used to send a copy of a packet back towards the source, as further described in Section 4.

Bit 2 "Active" (A-bit). When set, this indicates that this is an active IOAM packet, where "active" is used in the sense defined in [RFC7799], rather than a data packet. The packet may be an IOAM probe packet, or a replicated data packet (the second and third use cases of Section 5).

4. Loopback in IOAM

Loopback is used for triggering each transit device along the path to loop back a copy of the data packet. Loopback allows an IOAM encapsulating node to trace the path to a given destination, and to receive per-hop data about both the forward and the return path. Loopback is intended to provide an accelerated alternative to Traceroute, that allows the encapsulating node to receive responses from multiple transit nodes along the path in less then one round-trip-time, and by sending a single packet.

Loopback can be used only if a return path from transit nodes and destination nodes towards the source (encapsulating node) exists. Specifically, loopback is only applicable in encapsulations in which the identity of the encapsulating node is available in the encapsulation header. If an encapsulating node receives a looped

back packet that was not originated from the current encapsulating node, the packet is dropped.

The encapsulating node either generates synthetic packets with an IOAM trace option that has the loopback flag set, or sets the loopack flag in a subset of the in-transit data packets. Loopback is used either proactively or on-demand, i.e., when a failure is detected. The encapsulating node also needs to ensure that sufficient space is available in the IOAM header for loopback operation, which includes transit nodes adding trace data on the original path and then again on the return path.

An IOAM trace option that has the loopback bit set MUST have the value '1' in the most significant bit of IOAM-Trace-Type, and '0' in the rest of the bits of IOAM-Trace-Type. Thus, every transit node that processes this trace option only adds a single data field, which is the Hop_Lim and node_id data field. The reason for allowing a single data field per hop is to minimize the impact of amplification attacks.

A loopback bit that is set indicates to the transit nodes processing this option that they are to create a copy of the received packet and send the copy back to the source of the packet. In this context the source is the IOAM encapsulating node, and it is assumed that the source address is available in the encapsulation header. Thus, the source address of the original packet is used as the destination address in the copied packet. The address of the node performing the copy operation is used as the source address. The IOAM transit node pushes the required data field *after* creating the copy of the packet, in order to allow any egress-dependent information to be set based on the egress of the copy rather than the original packet. The copy is also truncated, i.e., any payload that resides after the IOAM option(s) is removed before transmitting the looped back packet back towards the encapsulating node. The original packet continues towards its destination. The L-bit MUST be cleared in the copy of the packet that a node sends back towards the source.

On its way back towards the source, the copied packet is processed like any other packet with IOAM information, including adding any requested data at each transit node (assuming there is sufficient space).

Once the return packet reaches the IOAM domain boundary, IOAM decapsulation occurs as with any other packet containing IOAM information. Note that the looped back packet does not have the L-bit set. The IOAM encapsulating node that initiated the original loopback packet recognizes a received packet as an IOAM looped-back packet by checking the Node ID in the Hop_Lim/node_id field that

corresponds to the first hop. If the Node ID matches the current IOAM node, it indicates that this is a looped back packet that was initiated by the current IOAM node, and processed accordingly. If there is no match in the Node ID, the packet is processed like a conventional IOAM-encapsulated packet.

Note that an IOAM encapsulating node may either be an endpoint (such as an IPv6 host), or a switch/router that pushes a tunnel encapsulation onto data packets. In both cases, the functionality that was described above avoids IOAM data leaks from the IOAM domain. Specificallly, if an IOAM looped-back packet reaches an IOAM boundary node that is not the IOAM node that initiated the loopback, the node does not process the packet as a loopback; the IOAM encapsulation is removed, and since the packet does not have any payload it is terminated. In either case, when the packet reaches the IOAM boundary its IOAM encapsulation is removed, preventing IOAM information from leaking out from the IOAM domain.

5. Active Measurement with IOAM

Active measurement methods [RFC7799] make use of synthetically generated packets in order to facilitate the measurement. This section presents use cases of active measurement using the IOAM Active flag.

The active flag indicates that a packet is used for active measurement. An IOAM decapsulating node that receives a packet with the Active flag set in one of its Trace options must terminate the packet. The active flag is intended to simplify the implementation of decapsulating nodes by indicating that the packet should not be forwarded further. It is not intended as a replacement for existing active OAM protocols, which may run in higher layers and make use of the active flag.

An example of an IOAM deployment scenario is illustrated in Figure 1. The figure depicts two endpoints, a source and a destination. The data traffic from the source to the destination is forwarded through a set of network devices, including an IOAM encapsulating node, which incorporates one or more IOAM options, a decapsulating node, which removes the IOAM options, optionally one or more transit nodes. The IOAM options are encapsulated in one of the IOAM encapsulation types, e.g., [I-D.ietf-sfc-ioam-nsh], or [I-D.ietf-ippm-ioam-ipv6-options].

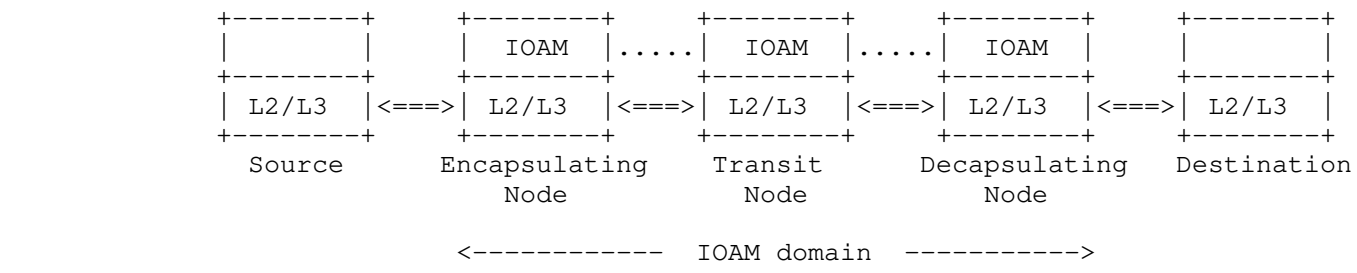


Figure 1: Network using IOAM.

This draft focuses on three possible use cases of active measurement using IOAM. These use cases are described using the example of Figure 1.

- Endpoint active measurement: synthetic probe packets are sent between the source and destination, traversing the IOAM domain. Since the probe packets are sent between the endpoints, these packets are treated as data packets by the IOAM domain, and do not require special treatment at the IOAM layer. Specifically, the active flag is not used in this case, and the IOAM layer needs not be aware that an active measurement mechanism is used at a higher layer.

- IOAM active measurement using probe packets within the IOAM domain: probe packets are generated and transmitted by the IOAM encapsulating node, and are expected to be terminated by the decapsulating node. IOAM data related to probe packets may be exported by one or more nodes along its path, by an exporting protocol that is outside the scope of this document (e.g., [I-D.spiegel-ippm-ioam-rawexport]). Probe packets include a Trace Option which has its Active flag set, indicating that the decapsulating node must terminate them.

- IOAM active measurement using replicated data packets: probe packets are created by the encapsulating node by selecting some or all of the en route data packets and replicating them. A selected data packet that is replicated, and its (possibly truncated) copy is forwarded with one or more IOAM option, while the original packet is forwarded normally, without IOAM options. To the extent possible, the original data packet and its replica are forwarded through the same path. The replica includes a Trace Option that has its Active flag set, indicating that the decapsulating node should terminate it. It should be noted that the current document defines the role of the active flag in allowing the decapsulating

node to terminate the packet, but the replication functionality in this context is outside the scope of this document.

6. IANA Considerations

IANA is requested to allocate the following bits in the "IOAM Trace Flags Registry" as follows:

Bit 1 "Loopback" (L-bit)

Bit 2 "Active" (A-bit)

Note that bit 0 is the most significant bit in the Flags Registry.

7. Performance Considerations

Each of the flags that are defined in this document may have performance implications. When using the loopback mechanism a copy of the data packet is sent back to the sender, thus generating more traffic than originally sent by the endpoints. Using active measurement with the active flag requires the use of synthetic (overhead) traffic.

Each of the mechanisms that use the flags above has a cost in terms of the network bandwidth, and may potentially load the node that analyzes the data. Therefore, it MUST be possible to use each of the mechanisms on a subset of the data traffic; an encapsulating node needs to be able to set the Loopback and Active flag selectively, in a way that considers the effect on the network performance. Similarly, transit and decapsulating nodes need to be able to selectively loop back packets with the Loopback flag, and to selectively export packets. Specifically, rate limiting can be enabled so as to ensure that the mechanisms are used at a rate that does not significantly affect the network bandwidth, and does not overload the receiving entity (or the source node in the case of loopback).

8. Security Considerations

The security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data]. Specifically, an attacker may try to use the functionality that is defined in this document to attack the network.

An attacker may attempt to overload network devices by injecting synthetic packets that include an IOAM Trace Option with one or more of the flags defined in this document. Similarly, an on-path

attacker may maliciously set one or more of the flags of transit packets.

- Loopback flag: an attacker that sets this flag, either in synthetic packets or transit packet, can potentially cause an amplification, since each device along the path creates a copy of the data packet and sends it back to the source. The attacker can potentially leverage the loopback flag for a Distributed Denial of Service (DDoS) attack, as multiple devices send looped-back copies of a packet to a single source.

- Active flag: the impact of synthetic packets with the active flag is no worse than synthetic data packets in which the Active flag is not set. By setting the active flag in en route packets an attacker can prevent these packets from reaching their destination, since the packet is terminated by the decapsulating device; however, note that an on-path attacker may achieve the same goal by changing the destination address of a packet. Another potential threat is amplification; if an attacker causes transit switches to replicate more packets than they are intended to replicate, either by setting the Active flag or by sending synthetic packets, then traffic is amplified, causing bandwidth degredation. As mentioned in Section 5, the specification of the replication mechanism is not within the scope of this document. A specification that defines the replication functionality should also address the security aspects of this mechanism.

Some of the security threats that were discussed in this document may be worse in a wide area network in which there are nested IOAM domains. For example, if there are two nested IOAM domains that use loopback, then a looped-back copy in the outer IOAM domain may be forwarded through another (inner) IOAM domain and may be subject to loopback in that (inner) IOAM domain, causing the amplification to be worse than in the conventional case.

In order to mitigate the attacks described above, as described in Section 7 it should be possible for IOAM-enabled devices to selectively apply the mechanisms that use the flags defined in this document to a subset of the traffic, and to limit the performance of synthetically generated packets to a configurable rate; specifically, network devices should be able to limit the rate of: (i) looped-back traffic (at transit nodes), (ii) replicated active packets (at encapsulating nodes), (iii) packets that are exported to a collector (from either encapsulating nodes or transit nodes), and (iv) synthetically generated packets (at encapsulating nodes).

Furthermore, as defined in Section 4, transit nodes that process a packet with the Loopback flag only add a single data field, and

truncate any payload that follows the IOAM option(s), thus significanly limiting the possible impact of an amplification attack.

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their affect. This is a fundamental assumtion with respect to the security aspects of IOAM, as further discussed in [I-D.ietf-ippm-ioam-data].

9. References

9.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.

[I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-05 (work in progress), December 2020.

[I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-04 (work in progress), November 2020.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

Authors' Addresses

Tal Mizrahi
Huawei
Israel

Email: tal.mizrahi.phd@gmail.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari (editor)
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
U.S.A.

Email: sramesh@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Aviv Kfir
Nvidia

Email: avivk@nvidia.com

Barak Gafni
Nvidia
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@nvidia.com

Mickey Spiegel
Barefoot Networks
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mspiegel@barefootnetworks.com

Jennifer Lemon
Broadcom
270 Innovation Drive
San Jose, CA 95134
US

Email: jennifer.lemon@broadcom.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2021

S. Bhandari
Thoughtspot
F. Brockners
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JMPC
T. Mizrahi
Huawei Network.IO Innovation Lab
A. Kfir
B. Gafni
Mellanox Technologies, Inc.
P. Lapukhov
Facebook
M. Spiegel
Barefoot Networks, an Intel company
S. Krishnan
Kaloom
R. Asati
Cisco
M. Smith
February 21, 2021

In-situ OAM IPv6 Options
draft-ietf-ippm-ioam-ipv6-options-05

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in IPv6.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in the IPv6 [RFC8200] and discusses deployment options for networks that use IPv6-encapsulated IOAM data fields. These options have distinct deployment considerations; for example, the IOAM domain can either be between hosts, or be between IOAM encapsulating and decapsulating network nodes that forward traffic, such as routers.

2. Conventions

2.1. 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.

2.2. Abbreviations

Abbreviations used in this document:

E2E: Edge-to-Edge

IOAM: In-situ Operations, Administration, and Maintenance

ION: IOAM Overlay Network

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

3. In-situ OAM Metadata Transport in IPv6

In-situ OAM in IPv6 is used to enhance diagnostics of IPv6 networks. It complements other mechanisms designed to enhance diagnostics of IPv6 networks, such as the IPv6 Performance and Diagnostic Metrics Destination Option described in [RFC8250].

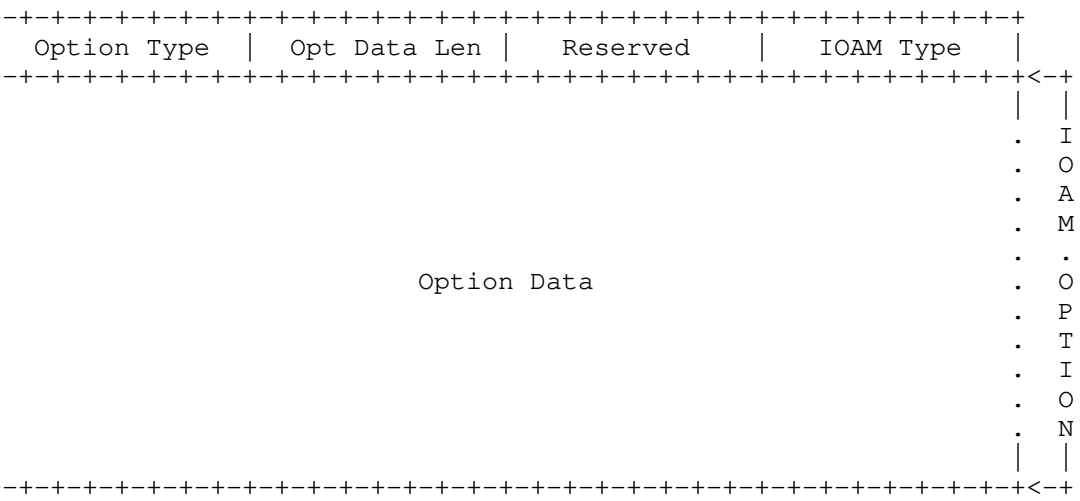
IOAM data fields can be encapsulated in "option data" fields using two types of extension headers in IPv6 packets - either Hop-by-Hop Options header or Destination options header. Deployments select one of these extension header types depending on how IOAM is used, as described in section 4 of [I-D.ietf-ippm-ioam-data]. Multiple

options with the same Option Type MAY appear in the same Hop-by-Hop Options or Destination Options header, with distinct content.

In order for IOAM to work in IPv6 networks, IOAM MUST be explicitly enabled per interface on every node within the IOAM domain. Unless a particular interface is explicitly enabled (i.e., explicitly configured) for IOAM, a router MUST drop packets that contain extension headers carrying IOAM data-fields. This is the default behavior and is independent of whether the Hop-by-Hop options or Destination options are used to encode the IOAM data. This ensures that IOAM data does not unintentionally get forwarded outside the IOAM domain.

An IPv6 packet carrying IOAM data in an Extension header can have other extension headers, compliant with [RFC8200].

IPv6 Hop-by-Hop and Destination Option format for carrying in-situ OAM data fields:



Option Type: 8-bit option type identifier as defined inSection 6.

Opt Data Len: 8-bit unsigned integer. Length of this option, in octets, not including the first 2 octets.

Reserved: 8-bit field MUST be set to zero upon transmission and ignored upon reception.

IOAM Type: 8-bit field as defined in section 7.2 in [I-D.ietf-ippm-ioam-data].

Option Data: Variable-length field. Option-Type-specific data.

In-situ OAM Option-Types are inserted as Option data as follows:

1. Pre-allocated Trace Option: The in-situ OAM Preallocated Trace Option-Type defined in [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in the Hop-by-Hop extension header:

   Option Type: 001xxxxx 8-bit identifier of the IOAM type of option. xxxxx=TBD.

   IOAM Option-Type: IOAM Pre-allocated Trace Option-Type.

2. Incremental Trace Option: The in-situ OAM Incremental Trace Option-Type defined in [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in the Hop-by-Hop extension header:

   Option Type: 001xxxxx 8-bit identifier of the IOAM type of option. xxxxx=TBD.

   IOAM Option-Type: IOAM Incremental Trace Option-Type.

3. Proof of Transit Option: The in-situ OAM POT Option-Type defined in [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in the Hop-by-Hop extension header:

   Option Type: 001xxxxx 8-bit identifier of the IOAM type of option. xxxxx=TBD.

   IOAM Option-Type: IOAM POT Option-Type.

4. Edge to Edge Option: The in-situ OAM E2E option defined in [I-D.ietf-ippm-ioam-data] is represented as an IPv6 option in Destination extension header:

   Option Type: 000xxxxx 8-bit identifier of the IOAM type of option. xxxxx=TBD.

   IOAM Option-Type: IOAM E2E Option-Type.

5. Direct Export (DEX) Option: The in-situ OAM Direct Export Option-Type defined in [I-D.ietf-ippm-ioam-direct-export] is represented as an IPv6 option in the Hop-by-Hop extension header:

Option Type: 000xxxxx 8-bit identifier of the IOAM type of option. xxxxx=TBD.

IOAM Option-Type: IOAM Direct Export (DEX) Option-Type.

All the in-situ OAM IPv6 options defined here have alignment requirements. Specifically, they all require 4n alignment. This ensures that fields specified in [I-D.ietf-ippm-ioam-data] are aligned at a multiple-of-4 offset from the start of the Hop-by-Hop and Destination Options header. In addition, to maintain IPv6 extension header 8-octet alignment and avoid the need to add or remove padding at every hop, the Trace-Type for Incremental Trace Option in IPv6 MUST be selected such that the IOAM node data length is a multiple of 8-octets.

IPv6 options can have a maximum length of 255 octets. Consequently, the total lenght of IOAM Option-Types including all data fields is also limited to 255 octets when encapsulated into IPv6.

4. IOAM Deployment In IPv6 Networks

4.1. Considerations for IOAM deployment in IPv6 networks

IOAM deployments in IPv6 networks should take the following considerations and requirements into account:

C1 It is desirable that the addition of IOAM data fields neither changes the way routers forward packets nor the forwarding decisions the routers take. Packets with added OAM information should follow the same path within the domain that an identical packet without OAM information would follow, even in the presence of ECMP. Such behavior is particularly important for deployments where IOAM data fields are only added "on-demand", e.g., to provide further insights in case of undesired network behavior for certain flows. Implementations of IOAM SHOULD ensure that ECMP behavior for packets with and without IOAM data fields is the same.

C2 Given that IOAM data fields increase the total size of a packet, the size of a packet including the IOAM data could exceed the PMTU. In particular, the incremental trace IOAM Hop-by-Hop (HbH) Option, which is intended to support hardware implementations of IOAM, changes Option Data Length en-route. Operators of an IOAM domain SHOULD ensure that the addition of OAM information does not lead to fragmentation of the packet, e.g., by configuring the MTU of transit routers and switches to a sufficiently high value. Careful control of the MTU in a network is one of the reasons why IOAM is considered a domain-specific feature (see also

[I-D.ietf-ippm-ioam-data]). In addition, the PMTU tolerance range in the IOAM domain should be identified (e.g., through configuration) and IOAM encapsulation operations and/or IOAM data field insertion (in case of incremental tracing) should not be performed if it exceeds the packet size beyond PMTU.

C3 Packets with IOAM data or associated ICMP errors, should not arrive at destinations that have no knowledge of IOAM. For exmample, if IOAM is used in in transit devices, misleading ICMP errors due to addition and/or presence of OAM data in a packet could confuse the host that sent the packet if it did not insert the OAM information.

C4 OAM data leaks can affect the forwarding behavior and state of network elements outside an IOAM domain. IOAM domains SHOULD provide a mechanism to prevent data leaks or be able to ensure that if a leak occurs, network elements outside the domain are not affected (i.e., they continue to process other valid packets).

C5 The source that inserts and leaks the IOAM data needs to be easy to identify for the purpose of troubleshooting, due to the high complexity of troubleshooting a source that inserted the IOAM data and did not remove it when the packet traversed across an Autonomous System (AS). Such a troubleshooting process might require coordination between multiple operators, complex configuration verification, packet capture analysis, etc.

C6 Compliance with [RFC8200] requires OAM data to be encapsulated instead of header/option insertion directly into in-flight packets using the original IPv6 header.

4.2. IOAM domains bounded by hosts

For deployments where the IOAM domain is bounded by hosts, hosts will perform the operation of IOAM data field encapsulation and decapsulation. IOAM data is carried in IPv6 packets as Hop-by-Hop or Destination options as specified in this document.

4.3. IOAM domains bounded by network devices

For deployments where the IOAM domain is bounded by network devices, network devices such as routers form the edge of an IOAM domain. Network devices will perform the operation of IOAM data field encapsulation and decapsulation.

4.4. Deployment options

This section lists out possible deployment options that can be employed to meet the requirements listed in Section 4.1.

4.4.1. IPv6-in-IPv6 encapsulation

The "IPv6-in-IPv6" approach preserves the original IP packet and add an IPv6 header including IOAM data fields in an extension header in front of it, to forward traffic within and across an IOAM domain. The overlay network formed by the additional IPv6 header with the IOAM data fields included in an extension header is referred to as IOAM Overlay Network (ION) in this document.

The following steps should be taken to perform an IPv6-in-IPv6 approach:

1. The source address of the outer IPv6 header is that of the IOAM encapsulating node. The destination address of the outer IPv6 header is the same as the inner IPv6 destination address, i.e., the destination address of the packet does not change.

2. To simplify debugging in case of leaked IOAM data fields, consider a new IOAM E2E destination option to identify the Source IOAM domain (AS, v6 prefix). Insert this option into the IOAM destination options EH attached to the outer IPv6 header. This additional information would allow for easy identification of an AS operator that is the source of packets with leaked IOAM information. Note that leaked packets with IOAM data fields would only occur in case a router would be misconfigured.

3. All the IOAM options are defined with type "00" - skip over this option and continue processing the header. Presence of these options must not cause packet drops in network elements that do not understand the option. In addition, [I-D.ietf-6man-hbh-header-handling] should be considered.

4.4.2. IP-in-IPv6 encapsulation with ULA

The "IP-in-IPv6 encapsulation with ULA" [RFC4193] approach can be used to apply IOAM to either an IPv6 or an IPv4 network. In addition, it fulfills requirement C4 (avoid leaks) by using ULA for the ION. Similar to the IPv6-in-IPv6 encapsulation approach above, the original IP packet is preserved. An IPv6 header including IOAM data fields in an extension header is added in front of it, to forward traffic within and across the IOAM domain. IPv6 addresses for the ION, i.e. the outer IPv6 addresses are assigned from the ULA space. Addressing and routing in the ION are to be configured so

that the IP-in-IPv6 encapsulated packets follow the same path as the original, non-encapsulated packet would have taken. This would create an internal IPv6 forwarding topology using the IOAM domain's interior ULA address space which is parallel with the forwarding topology that exists with the non-IOAM address space (the topology and address space that would be followed by packets that do not have supplemental IOAM information). Establishment and maintenance of the parallel IOAM ULA forwarding topology could be automated, e.g., similar to how LDP [RFC5036] is used in MPLS to establish and maintain an LSP forwarding topology that is parallel to the network's IGP forwarding topology.

Transit across the ION could leverage the transit approach for traffic between BGP border routers, as described in [RFC1772], "A.2.3 Encapsulation". Assuming that the operational guidelines specified in Section 4 of [RFC4193] are properly followed, the probability of leaks in this approach will be almost close to zero. If the packets do leak through IOAM egress device misconfiguration or partial IOAM egress device failure, the packets' ULA destination address is invalid outside of the IOAM domain. There is no exterior destination to be reached, and the packets will be dropped when they encounter either a router external to the IOAM domain that has a packet filter that drops packets with ULA destinations, or a router that does not have a default route.

4.4.3. x-in-IPv6 Encapsulation that is used Independently

In some cases it is desirable to monitor a domain that uses an overlay network that is deployed independently of the need for IOAM, e.g., an overlay network that runs Geneve-in-IPv6, or VXLAN-in-IPv6. In this case IOAM can be encapsulated in as an extension header in the tunnel (outer) IPv6 header. Thus, the tunnel encapsulating node is also the IOAM encapsulating node, and the tunnel end point is also the IOAM decapsulating node.

5. Security Considerations

This document describes the encapsulation of IOAM data fields in IPv6. Security considerations of the specific IOAM data fields for each case (i.e., Trace, Proof of Transit, and E2E) are described and defined in [I-D.ietf-ippm-ioam-data].

As this document describes new options for IPv6, these are similar to the security considerations of [RFC8200] and the weakness documented in [RFC8250].

6. IANA Considerations

This draft requests the following IPv6 Option Type assignments from the Destination Options and Hop-by-Hop Options sub-registry of Internet Protocol Version 6 (IPv6) Parameters.

http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2

Hex Value	Binary Value act	chg	rest	Description	Reference
TBD_1_0	00	0	TBD_1	IOAM	[This draft]
TBD_1_1	00	1	TBD_1	IOAM	[This draft]

7. Acknowledgements

The authors would like to thank Tom Herbert, Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, Mark Smith, Andrew Yourtchenko and Justin Iurman for the comments and advice. For the IPv6 encapsulation, this document leverages concepts described in [I-D.kitamura-ipv6-record-route]. The authors would like to acknowledge the work done by the author Hiroshi Kitamura and people involved in writing it.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.

[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>.

[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>.

8.2. Informative References

[I-D.ietf-6man-hbh-header-handling]
Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options Extension Header", March 2016.

[I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.

[RFC1772] Rekhter, Y. and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, DOI 10.17487/RFC1772, March 1995, <https://www.rfc-editor.org/info/rfc1772>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <https://www.rfc-editor.org/info/rfc8250>.

Authors' Addresses

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Frank Brockners
Cisco Systems, Inc.
Kaiserswerther Str. 115,
RATINGEN, NORDRHEIN-WESTFALEN 40880
Germany

Email: fbrockne@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mickey.spiegel@intel.com

Suresh Krishnan
Kaloom

Email: suresh@kaloom.com

Rajiv Asati
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: rajiva@cisco.com

Mark Smith
PO BOX 521
HEIDELBERG, VIC 3084
AU

Email: markzzzsmith+id@gmail.com

Network Working Group
Internet-Draft
Updates: 8762 (if approved)
Intended status: Standards Track
Expires: May 19, 2021

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

November 15, 2020

Simple Two-way Active Measurement Protocol Optional Extensions
draft-ietf-ippm-stamp-option-tlv-10

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 19, 2021.

Table of Contents

1. Introduction

Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] defined 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]

TLV Type-Length-Value

SSID STAMP Session Identifier

SSU Synchronization Supply Unit

STAMP Simple Two-way Active Measurement Protocol

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-long non-zero unsigned integer. SSID generation policy is implementation-specific. [I-D.gont-numeric-ids-generation] thoroughly analyzes common algorithms for identifier generation and their vulnerabilities. For example, an implementation can use algorithms described in Section 7.1 of [I-D.gont-numeric-ids-generation]. 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.

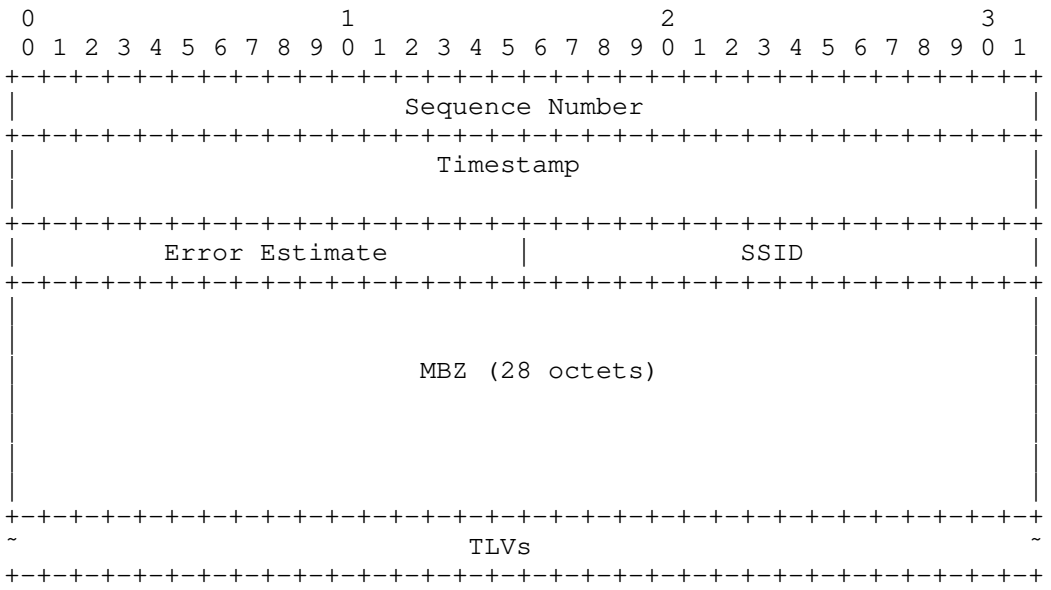


Figure 1: The format of an extended STAMP Session-Sender test packet in unauthenticated mode

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 packet(s). The means of provisioning the STAMP Session identification is outside the scope of this specification. A conforming implementation of STAMP Session-Reflector MUST copy the SSID value from the received test packet and put it into the reflected packet, as displayed in Figure 2.

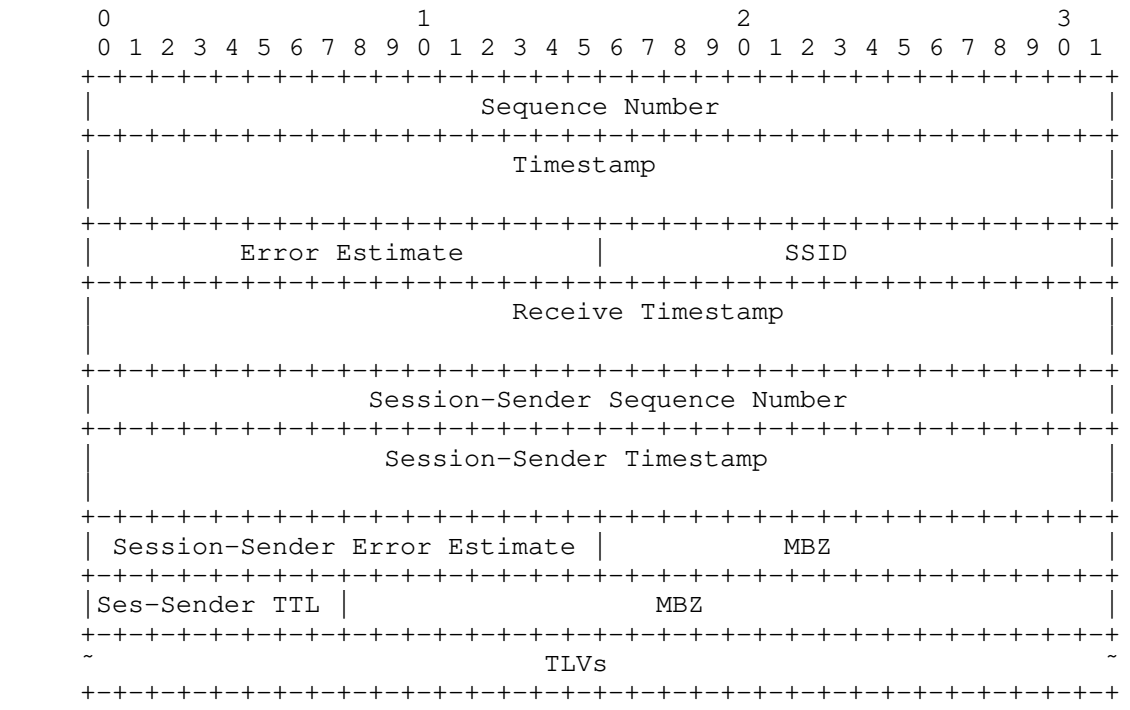


Figure 2: The format of an extended STAMP Session-Reflector test packet in unauthenticated mode

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.

Location of the SSID field in the authenticated mode is shown in Figure 3 and Figure 4.

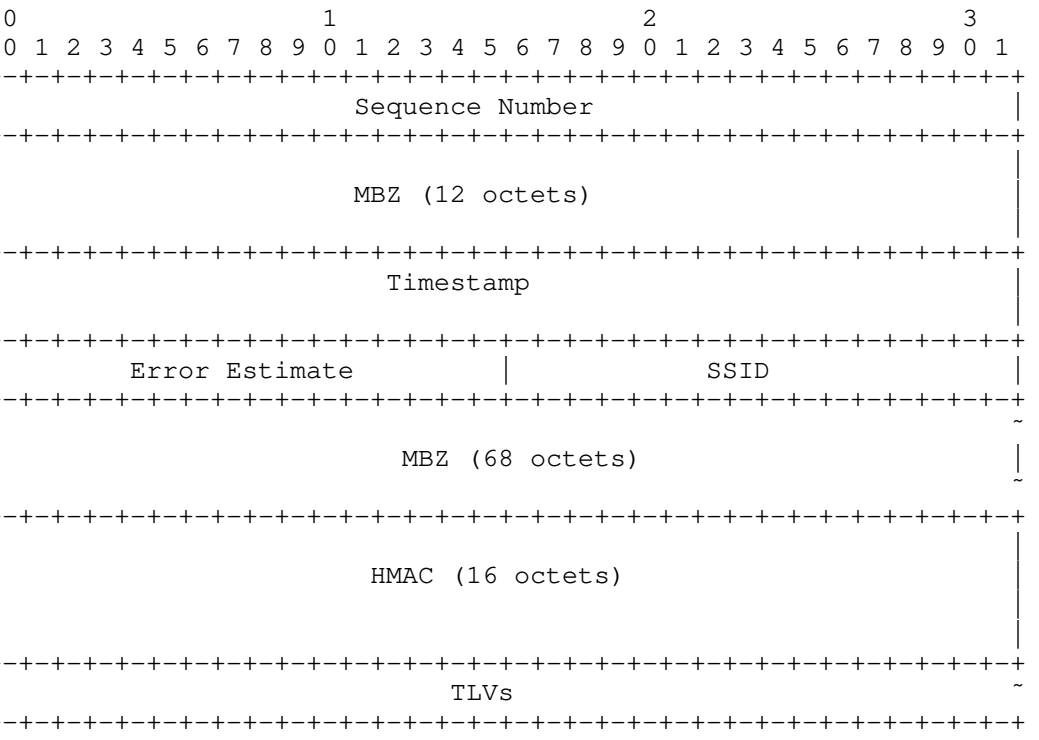
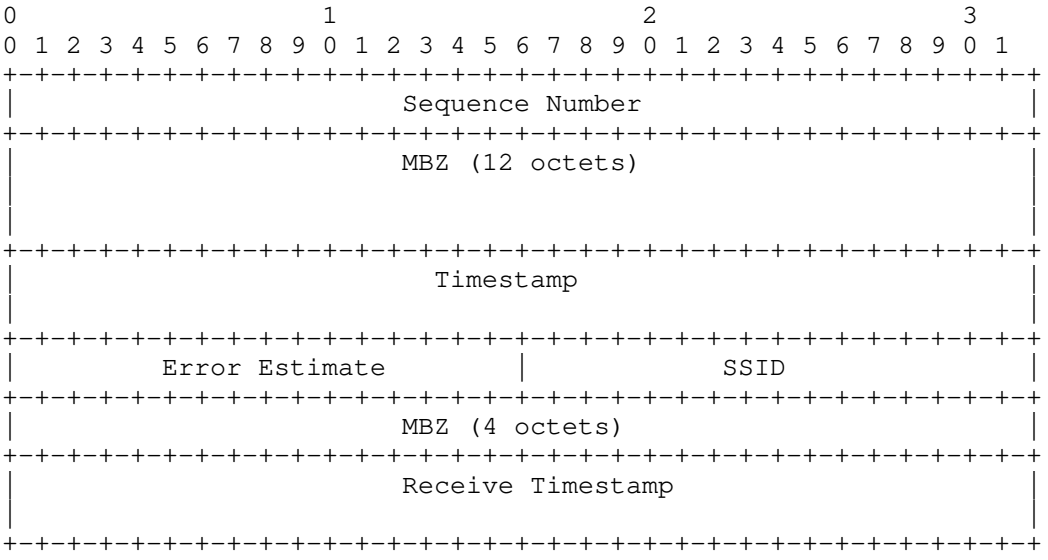


Figure 3: Base STAMP Session-Sender test packet format in authenticated mode



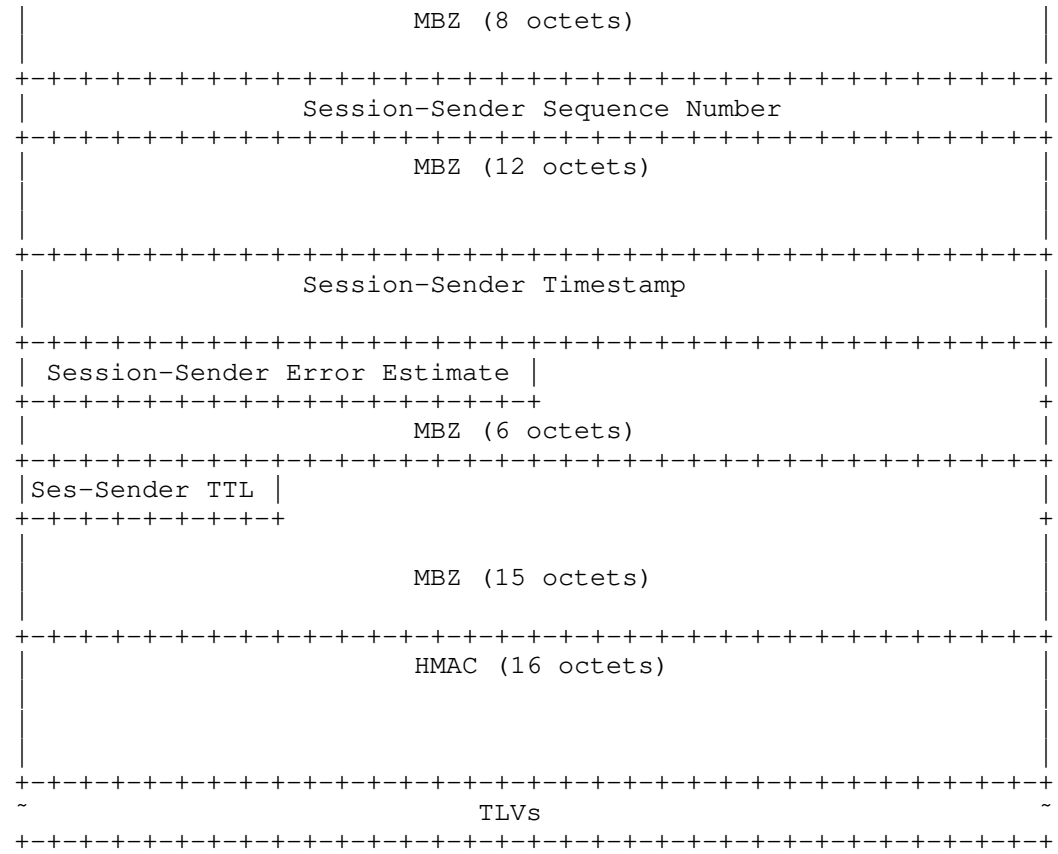


Figure 4: Base STAMP Session-Reflector test packet format in authenticated mode

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-long STAMP TLV Flags field, a one-octet-long Type field, and a two-octet-long Length field that is equal to the length of the Value field in octets. If a Type value for TLV or sub-TLV is in the range for Vendor Private Use, the Length MUST be at least 4, and the first four octets MUST be that vendor's Structure of Management Information (SMI) Private Enterprise Code, as recorded in IANA's SMI Private Enterprise Codes sub-registry, 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.

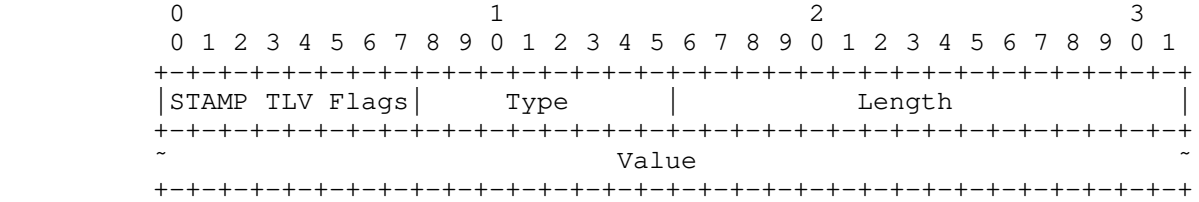


Figure 5: TLV Format in a STAMP Extended Packet

where fields are defined as the following:

- STAMP TLV Flags - eight-bit-long field. Detailed format and interpretation of flags defined in this specification is below.

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

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

- Value - a variable-length field. Its interpretation and encoding is determined by the value of the Type field.

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

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

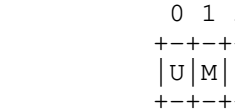


Figure 6: STAMP TLV Flags Format

where fields are defined as the following:

- U (Unrecognized) is 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) is 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) is 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 includes one or more TLVs. The node MUST compare the value in the Length field of the UDP header and the length of the base STAMP test packet in the mode, unauthenticated or authenticated based on the configuration of the particular STAMP test session. If the difference between the two values is larger than the length of 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 [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 into 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

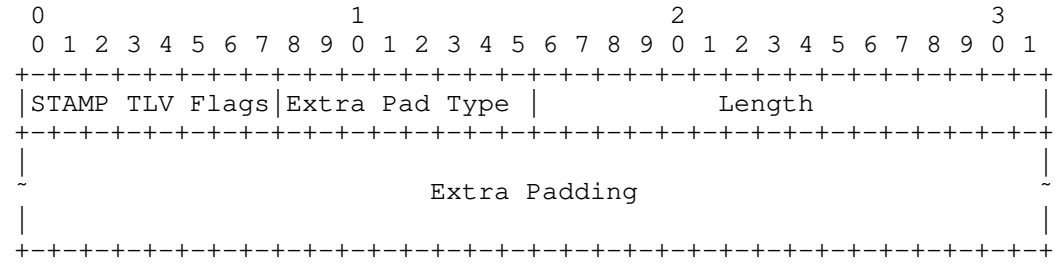


Figure 7: Extra Padding TLV

where fields are defined as the following:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- Extra Padding Type - is a one-octet-long field, value TBA1 allocated by IANA Section 5.1.
- Length - two-octet-long field equal to the length of the Extra Padding field in octets.
- Extra Padding - SHOULD be filled by a sequence of a pseudo-random numbers. The field MAY be filled with all zeros. An implementation MUST control the type of filling 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 of larger size 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 STAMP TLV Flags, Type, and Length. 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.

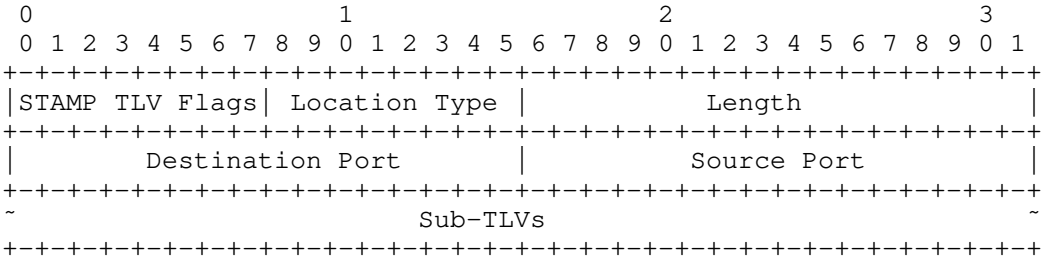


Figure 8: Location TLV

where fields are defined as the following:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.

- Location Type - is a one-octet-long field, value TBA2 allocated by IANA Section 5.1.

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

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

- Source Port - two-octet-long 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-TLV for the Location TLV are defined in this specification (type values are assigned according to Table 5):

- Source MAC Address sub-TLV - is a 12-octet-long sub-TLV. The Type value is TBA9. The value of the Length field MUST equal to 8. The Value field is an 8-octet-long MBZ field that MUST be zeroed on transmission and ignored on receipt.

- Source EUI-48 Address sub-TLV - is a 12-octet-long sub-TLV that includes the EUI-48 source MAC address. The Type value is TBA10. The value of the Length field MUST equal to 8.

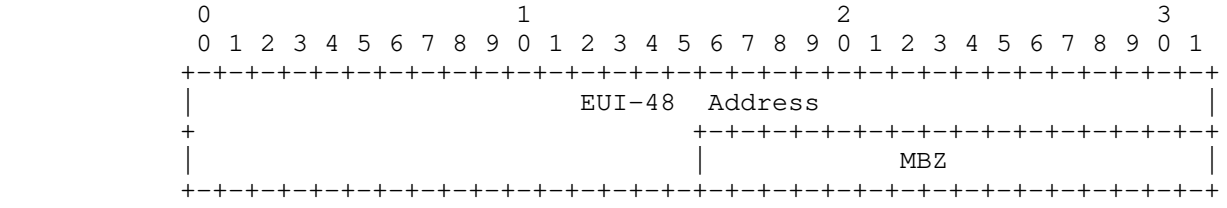


Figure 9: The Value Field of the Source EUI-48 Address sub-TLV

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

* The EUI-48 is a six-octet-long field.

* Two-octet-ling MBZ field MUST be zeroed on transmission and ignored on receipt.

- Source EUI-64 Address sub-TLV - is a 12-octet-long sub-TLV that includes the EUI-64 source MAC address. The Type value is TBA11. The value of the Length field MUST equal to 8. The Value field consists of an eight-octet-long EUI-64 field.

- Destination IP Address sub-TLV - is a 20-octet-long sub-TLV. The Type value is TBA12. The value of the Length field MUST equal to 16. The Value field consists of a 16-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt

- Destination IPv4 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv4 destination address. The Type value is TBA13. The value of the Length field MUST equal to 16.

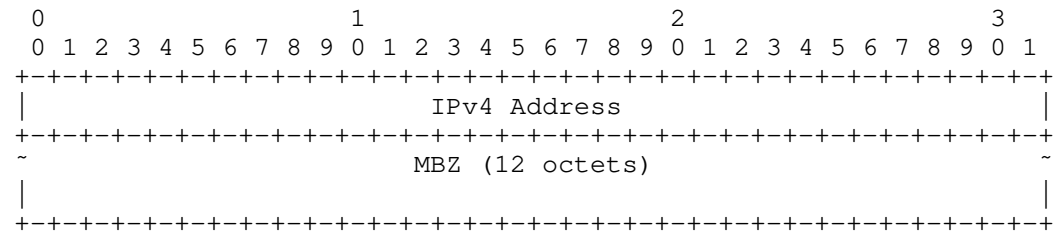


Figure 10: IPv4 Address in a Sub-TLV's Value Field

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

* The IPv4 Address is a four-octet-long field.

* 12-octet-long MBZ field MUST be zeroed on transmit and ignored on receipt.

- Destination IPv6 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv6 destination address. The Type value is TBA14. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long IP v6 Address field.

- Source IP Address sub-TLV - is a 20-octet-long sub-TLV. The Type value is TBA15. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt

- Source IPv4 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv4 source address. The Type value is TBA16. The value of the Length field MUST equal to 16. The Value field consists of the following fields (Figure 10):

  * The IPv4 Address is a four-octet-long field.

  * 12-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt.

- Source IPv6 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv6 source address. The Type value is TBA17. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long 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 the Location TLV of the size equal to the size of Location TLV in the received packet in the reflected 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 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 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 cause 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.

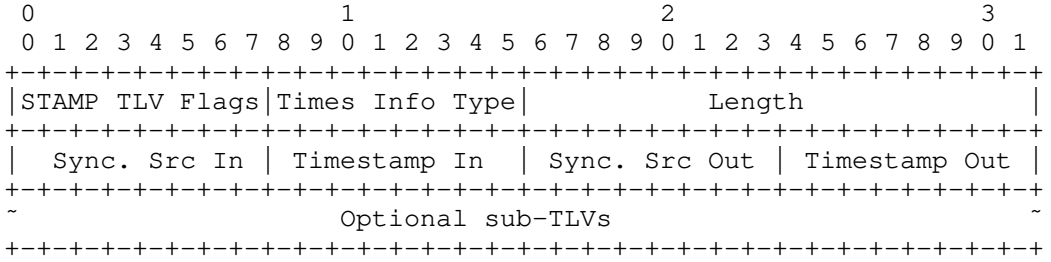


Figure 11: Timestamp Information TLV

where fields are defined as the following:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.

- Timestamp Information Type - is a one-octet-long field, value TBA3 allocated by IANA Section 5.1.

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

- Sync Src In - one-octet-long field that characterizes the source of clock synchronization at the ingress of a Session-Reflector. There are several methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905]. The value is one of those listed in Table 7.

- Timestamp In - one-octet-long 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 software API from a local wall clock, or from a remote clock (the latter is referred to as "control plane"). The value is one of those listed in Table 9.

- Sync Src Out - one-octet-long field that characterizes the source of clock synchronization at the egress of the Session-Reflector. The value is one of those listed in Table 7.

- Timestamp Out - one-octet-long field that characterizes the method by which the egress of the Session-Reflector obtained the timestamp T3. The value is one of those listed in Table 9.

- Optional sub-TLVs - 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.

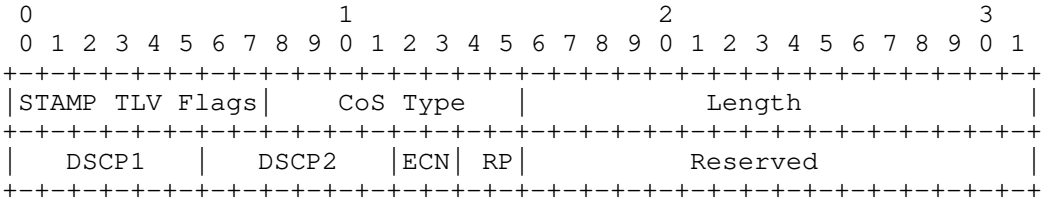


Figure 12: Class of Service TLV

where fields are defined as the following:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.

- CoS (Class of Service) Type - is a one-octet-long field, value TBA4 allocated by IANA Section 5.1.

- Length - two-octet-long 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) - is a two-bit-long field. A Session-Sender MUST set the value of the RP field to 0 on transmission.

- Reserved - 16-bit-long field, 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-Reflectorset 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.

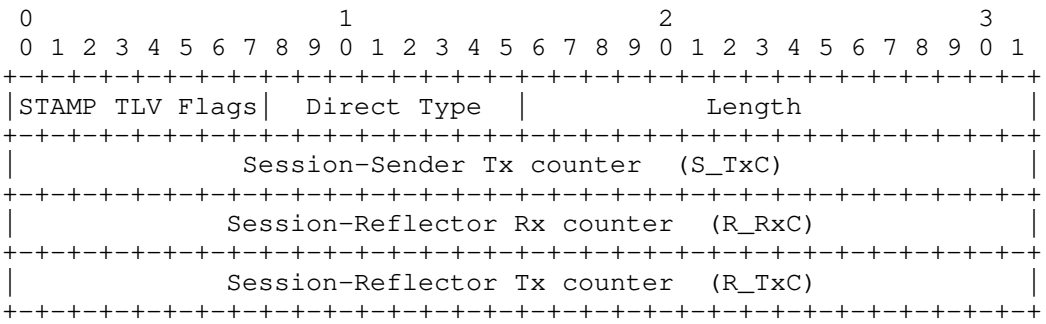


Figure 13: Direct Measurement TLV

where fields are defined as the following:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.

- Direct (Measurement) Type - is a one-octet-long field, value TBA5 allocated by IANA Section 5.1.

- Length - two-octet-long field equals the length of the Value field in octets. The Length field value MUST equal 12 octets.

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

- Session-Reflector Rx counter (R_RxC) is a four-octet-long field. 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) is a four-octet-long field. 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.

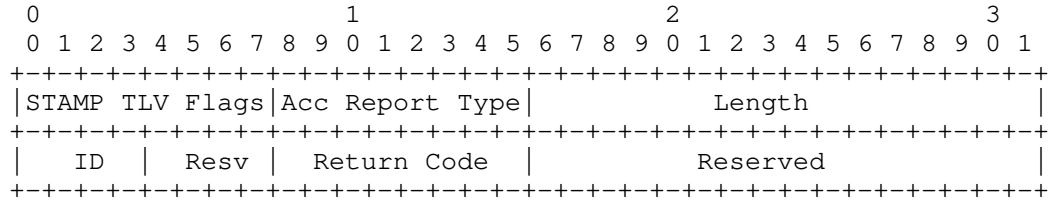


Figure 14: Access Report TLV

where fields are defined as follows:

- STAMP TLV Flags - is an eight-bit-long field. Its format presented in Figure 6.

- Access Report Type - is a one-octet-long field, value TBA6 allocated by IANA Section 5.1.

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

- ID (Access ID) - four-bit-long field that identifies the access network, e.g., 3GPP (Radio Access Technologies specified by 3GPP) or Non-3GPP (accesses that are not specified by 3GPP) [TS23501]. The value is one of those listed below:

  * 1 - 3GPP Network

  * 2 - Non-3GPP Network

  All other values are invalid and the TLV that contains it MUST be discarded.

- Resv - four-bit-long field, MUST be zeroed on transmission and ignored on receipt.

- Return Code - one-octet-long field that identifies the report signal, e.g., available or unavailable. The value is supplied to the STAMP end-point through some mechanism that is outside the scope of this document. The value is one of those listed in Section 5.6.

- Reserved - two-octet-long field, MUST be zeroed on transmission and ignored on receipt.

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

The Session-Sender MUST also arm a retransmission timer after sending a test packet that includes the Access Report TLV. This timer MUST be disarmed upon 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 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 in the Timestamp field only an "SW Local" (see Table 9) timestamp. 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 measuring network delay by minimizing the impact of egress queuing delays on the measurement.

A STAMP Session-Sender MAY include the Follow-up Telemetry TLV to request information from the Session-Reflector. The Session-Sender MUST set the Follow-up Telemetry Type and Length fields to their

appropriate values. The Sequence Number and Timestamp fields MUST be zeroed on transmission by the Session-Sender and ignored by the Session-Reflector upon receipt of the STAMP test packet that includes the Follow-up Telemetry TLV. The Session-Reflector MUST validate the Length value of the STAMP test packet. If the value of the Length field is invalid, the Session-Reflector MUST zero the Sequence Number and Timestamp fields and set the M flag in the STAMP TLV Flags field in the reflected packet. If the Session-Reflector is in stateless mode (defined in Section 4.2 [RFC8762]), it MUST zero the Sequence Number and Timestamp fields.

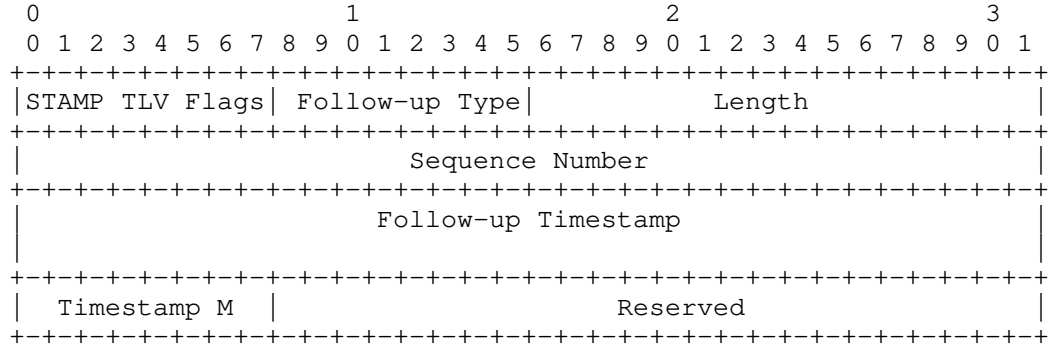


Figure 15: Follow-up Telemetry TLV

where fields are defined as follows:

- STAMP TLV Flags - is an eight-bit-long field. Its format presented in Figure 6.

- Follow-up (Telemetry) Type - is a one-octet-long field, value TBA7 allocated by IANA Section 5.1.

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

- Sequence Number - four-octet-long field indicating the sequence number of the last packet reflected in the same STAMP-test session. Since the Session-Reflector runs in the stateful mode (defined in Section 4.2 [RFC8762]), it is the Session-Reflector's Sequence Number of the previous reflected packet.

- Follow-up Timestamp - eight-octet-long 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 [RFC8762]. It carries the timestamp when the reflected packet with the specified sequence number was sent.

- Timestamp M(ode) - one-octet-long field that characterizes the method by which the entity that transmits a reflected STAMP packet obtained the Follow-up Timestamp. The value is one of those listed in Table 9.

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

4.8. HMAC TLV

The STAMP authenticated mode protects the integrity of data collected in 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 Section 4.2.2 and Section 4.3.2 [RFC8762]) compatible with this specification MUST additionally authenticate the option 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 in this section, further below. The HMAC TLV MAY be used to protect the integrity of STAMP extensions in STAMP unauthenticated mode. An implementation of STAMP extensions MUST provide controls to enable the integrity protection of STAMP extensions in STAMP unauthenticated mode.

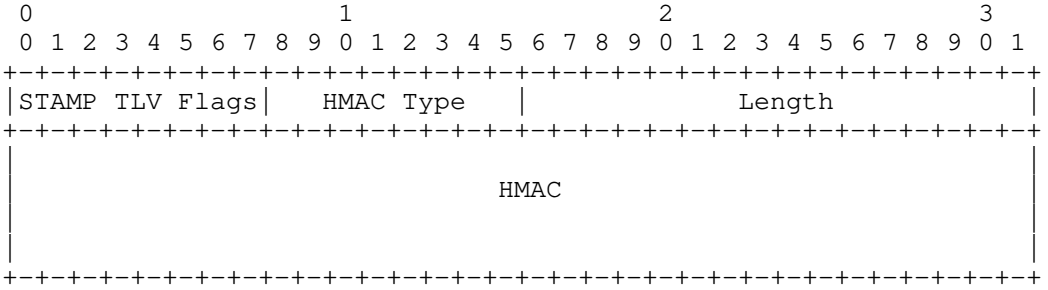


Figure 16: HMAC TLV

where fields are defined as follows:

- STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.

- HMAC Type - is a one-octet-long field, value TBA8 allocated by IANA Section 5.1.

- Length - two-octet-long field, set equal to 16 octets.

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

As defined in [RFC8762], STAMP uses HMAC-SHA-256 truncated to 128 bits ([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. 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

5.1. STAMP TLV Registry

IANA is requested to create the STAMP TLV Type registry. All code points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. The remaining code points are allocated according to Table 1:

Value	Description	Reference
0	Reserved	This document
1- 175	Unassigned	This document
176 - 239	Unassigned	This document
240 - 251	Experimental	This document
252 - 254	Private Use	This document
255	Reserved	This document

Table 1: STAMP TLV Type Registry

This document defines the following new values in the IETF Review range of the STAMP TLV Type registry:

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

Table 2: STAMP TLV Types

5.2. STAMP TLV Flags Sub-registry

IANA is requested to create the STAMP TLV Flags sub-registry as part of the STAMP TLV Type registry. The registration procedure is "IETF Review" [RFC8126]. Flags are 8 bits. This document defines the following bit positions in the STAMP TLV Flags sub-registry:

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

Table 3: STAMP TLV Flags

5.3. Sub-TLV Type Sub-registry

IANA is requested to create the sub-TLV Type sub-registry as part of the STAMP TLV Type registry. All code points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. The remaining code points are allocated according to Table 4:

Value	Description	Reference
0	Reserved	This document
1- 175	Unassigned	This document
176 - 239	Unassigned	This document
240 - 251	Experimental	This document
252 - 254	Private Use	This document
255	Reserved	This document

Table 4: Location Sub-TLV Type Sub-registry

This document defines the following new values in the IETF Review range of the Location sub-TLV Type sub-registry:

Value	Description	TLV Used	Reference
TBA9	Source MAC Address	Location	This document
TBA10	Source EUI-48 Address	Location	This document
TBA11	Source EUI-64 Address	Location	This document
TBA12	Destination IP Address	Location	This document
TBA13	Destination IPv4 Address	Location	This document
TBA14	Destination IPv6 Address	Location	This document
TBA15	Source IP Address	Location	This document
TBA16	Source IPv4 Address	Location	This document
TBA17	Source IPv6 Address	Location	This document

Table 5: STAMP sub-TLV Types

5.4. Synchronization Source Sub-registry

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

the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 6:

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

Table 6: Synchronization Source Sub-registry

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

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

Table 7: Synchronization Sources

5.5. Timestamping Method Sub-registry

IANA is requested to create the Timestamping Method sub-registry as part of the STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 8:

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

Table 8: Timestamping Method Sub-registry

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

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

Table 9: Timestamping Methods

5.6. Return Code Sub-registry

IANA is requested to create the Return Code sub-registry as part of the STAMP TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 128 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 10:

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

Table 10: Return Code Sub-registry

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

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

Table 11: 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 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;
- validate the Length field value.

As this specification defined 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 verifies the set of CoS values that are 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. Acknowledgments

Authors 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 the most valuable suggestions. The authors greatly appreciate comments and thoughtful suggestions received from Martin Duke.

8. Contributors

The following people contributed text to this document:

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

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

9. References

9.1. Normative References

[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>.

9.2. Informative References

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

[I-D.gont-numeric-ids-generation]
Gont, F. and I. Arce, "On the Generation of Transient Numeric Identifiers", draft-gont-numeric-ids-generation-04 (work in progress), July 2019.

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

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

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

Henrik Nydell
Accedian Networks

Email: hnydell@accedian.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

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

Email: adi@apple.com

Ernesto Ruffini
OutSys
via Caracciolo, 65
Milano 20155
Italy

Email: eruffini@outsys.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

Z. Li
China Mobile
M. Chen
Huawei
G. Mirsky
ZTE Corp.

November 02, 2020

Performance Measurement on LAG
draft-li-ippm-pm-on-lag-03

Abstract

This document defines extensions to One-way Active Measurement Protocol (OWAMP), Two-way Active Measurement Protocol (TWAMP), and Simple Two-Way Active Measurement Protocol (STAMP) to implement performance measurement on every member link of a Link Aggregation Group (LAG). With the measured metrics of each member links of a LAG, it enables operators to enforce performance metric based traffic steering policy among the member links.

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 6, 2021.

Table of Contents

1. Problem Statement

Link Aggregation Group (LAG), as defined in [IEEE802.1AX], provides mechanisms to combine multiple physical links into a single logical link. This logical link provides higher bandwidth and better resiliency, because if one of the physical member links fails, the

aggregate logical link can continue to forward traffic over the remaining operational physical member links.

Normally, when forwarding traffic over a LAG, a hash based or the like mechanism is used to load balance the traffic among member links of the LAG. In some cases, the link delays of the member links are different because the member links are over different transport paths. To provide low delay service to time sensitive traffic, we have to know the link delay of each member link of a LAG and then steer traffic accordingly. This requires a solution that could measure the performance metrics of each member link of a LAG.

However, when using One-way Active Measurement Protocol (OWAMP) [RFC4656], Two-way Active Measurement Protocol (TWAMP) [RFC5357], or Simple Two-Way Active Measurement Protocol (STAMP) [RFC8762] to measure the performance of a LAG, the LAG is treated as a single logical link/path. The measured metrics reflect the performance of one member link or an average of some/all member links of the LAG.

In addition, for LAG, using passive or hybrid methods (like alternative marking[RFC8321] or iOAM [I-D.ietf-ippm-ioam-data]) can only monitor the link crossed by traffic. Means the measured metrics only reflect the performance of some member links or an average of some/all member links of the LAG as well. Therefore, in order to measure every link of a LAG, using active methods would be more appropriate.

This document defines extensions to OWAMP [RFC4656], TWAMP [RFC5357] or STAMP [RFC8762] to implement performance measurement on every member link of a LAG.

2. Micro Session on LAG

This document intends to address the scenario (e.g., Figure 1) where two hosts (A and B) are directly connected by a LAG (e.g., the LAG is consisted by three links). The purpose is to measure the performance of each link of the LAG.

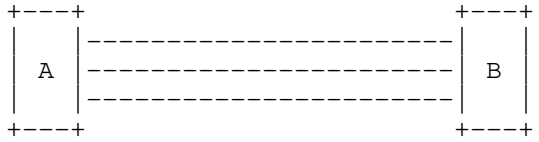


Figure 1: PM for LAG

To measure performance metrics of every member link of a LAG, multiple sessions (one session for each member link) need to be

established between the two hosts that are connected by the LAG. These sessions are called micro sessions in the remainder of this document.

All micro sessions of a LAG share the same Sender Address, Receiver Address. As for the Sender Port and Receiver Port, the micro sessions may share the same Sender Port and Receiver Port pair, or each micro session is configured with different Sender Port and Receiver Port pair. But from simplifying operation point of view, the former is recommended.

In addition, with micro sessions, there needs a way to correlate a session with a member link. For example, when receives a Control or Test packet, the Server/Reflector/Receiver needs to know from which member link the packet is received, and then correlate the packet with a micro session. This is different from the existing OWAMP [RFC4656], TWAMP [RFC5357], or STAMP [RFC8762].

This document defines new command types to indicate that a session is a micro session, the details are described in Section 3 and 4 of this document. For a micro session, on receiving of a Control/Test packet, the receiver uses the receiving link to correlate the packet with a particular session. In addition, Test packets may need to carry the member link information for validation checking. For example, when a Session-Sender receives a Test packet, it may need to check whether the Test packet is from the expected member link.

3. Mirco OWAMP Session

This document assumes that the OWAMP Server and the OWAMP Receiver of an OWAMP micro session are at the same host.

3.1. Micro OWAMP-Control

To support micro OWAMP session, a new command, which is referred to as Request-OW-Micro-Session (TBD1), is defined in this document. The Request-OW-Micro-Session command is based on the OWAMP Request-Session command, and uses the message format as described in Section 3.5 of OWAMP [RFC4656]. Test session creation of micro OWAMP session follows the same procedure as defined in Section 3.5 of OWAMP [RFC4656] with the following additions:

When a OWAMP Server receives a Request-OW-Micro-Session command, if the Session is accepted, the OWAMP Server MUST build an association between the session and the member link from which the Request-Session message is received.

3.2. Micro OWAMP-Test

Micro OWAMP-Test reuses the OWAMP-Test packet format and procedures as defined in Section 4 of OWAMP [RFC4656] with the following additions:

The micro OWAMP Sender MUST send the micro OWAMP-Test packets over the member link with which the session is associated. When receives a Test packet, the micro OWAMP receiver MUST use the member link from which the Test packet is received to correlate the micro OWAMP session. If there is no such a session, the Test packet MUST be discarded.

4. Mirco TWAMP Session

As above, this document assumes that the TWAMP Server and the TWAMP Session-Reflector of a micro OWAMP session are at the same host.

4.1. Micro TWAMP-Control

To support micro TWAMP session, a new command, which is referred to as Request-TW-Micro-Session (TBD2), is defined in this document. The Request-TW-Micro-Session command is based on the TWAMP Request-Session command, and uses the message format as described in Section 3.5 of TWAMP [RFC5357]. Test session creation of micro TWAMP session follows the same procedure as defined in Section 3.5 of TWAMP [RFC5357] with the following additions:

When a micro TWAMP Server receives a Request-TW-Micro-Session command, if the micro TWAMP Session is accepted, the micro TWAMP Server MUST build an association between the session and the member link from which the Request-Session message is received.

4.2. Micro TWAMP-Test

The micro TWAMP-Test protocol is based on the TWAMP-Test protocol [RFC5357] with the following extensions.

4.2.1. Sender Behavior

In addition to inheriting the TWAMP sender behavior as defined Section 4.1 of [RFC5357], the micro TWAMP Session-Sender MUST send the micro TWAMP-Test packets over the member link with which the session is associated.

When sending Test packet, the micro TWAMP Session-Sender MUST put the Sender member link identifier that is associated with the micro TWAMP session in the Sender Member Link ID. If the Session-Sender knows

the Reflector member link identifier, it MUST put it in the Reflector Member Link ID fields (see Figure 2 and Figure 3). Otherwise, the Reflector Member Link ID field MUST be set to zero.

The Sender member link identifier is used by the Session-Sender to check whether a reflected Test packet is received from the member link that associates to the correct micro TWAMP session. Therefore, it is carried in the Sender Member Link ID field of a Test packet and sent to the Session-Reflector. Then it will be sent back by the Session-Reflector with the reflected Test packet.

The Reflector member link identifier carried in the Reflector Member Link ID field is used by the Session-Receiver to check whether a Test packet is received from the member link that associates to the correct micro TWAMP session. Means that the Session-Sender has to learns the Reflector member link identifier. Once the Session-Sender learns the Reflector member link identifier, it MUST put the identifier in the Reflector Member Link ID field (see Figure 2 or Figure 3) of the Test packets that will be sent to the Session-Reflector. The Reflector member link identifier can be obtained from pre-configuration or learned through control plane or data plane (e.g., learned from a reflected Test packet). How to abtain/learn the Reflector member link identifier is out of the scope of this document.

When receives a reflected Test packet, the micro TWAMP Session-Sender MUST use the receiving member link to correlate the reflected Test packet to a micro TWAMP session. If there is no such a session, the reflected Test packet MUST be discarded. If a matched session exists, the Session-Sender MUST use the identifier carried in the Sender Member Link ID field to validate whether the reflected Test packet is correctly transmitted over the expected member link. If the validation is failed, the Test packet MUST be discarded.

4.2.1.1. Packet Format and Content

The micro TWAMP Session-Sender packet format is based on the TWAMP Session-Sender packet format as defined in Section 4.1.2 of [RFC5357]. In addition, in order to carry the LAG member link identifier, two new fields (Sender and Reflector Member Link ID) are added. The formats are as below:

For unauthenticated mode:

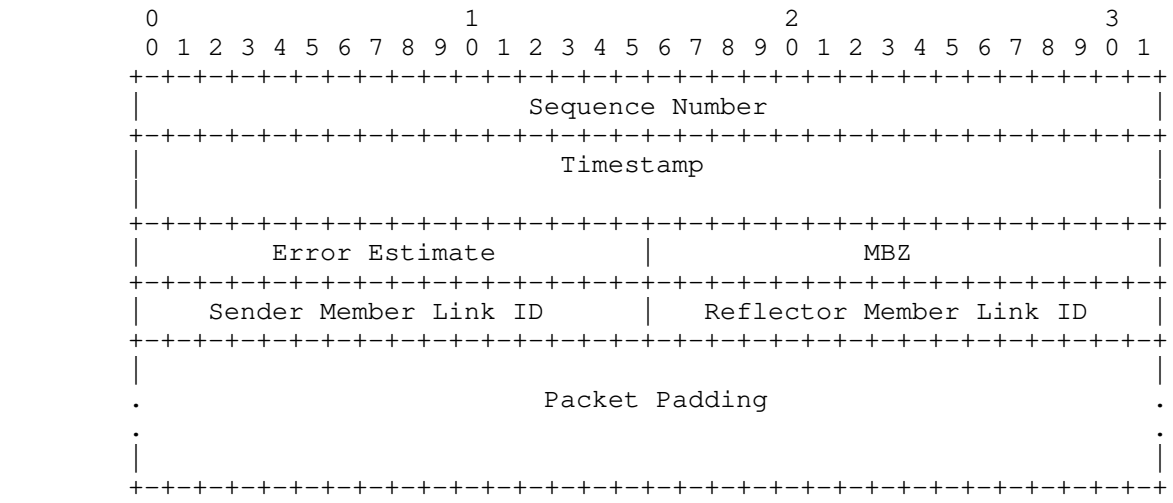


Figure 2: Session-Sender Packet format in Unauthenticated Mode

For authenticated mode:

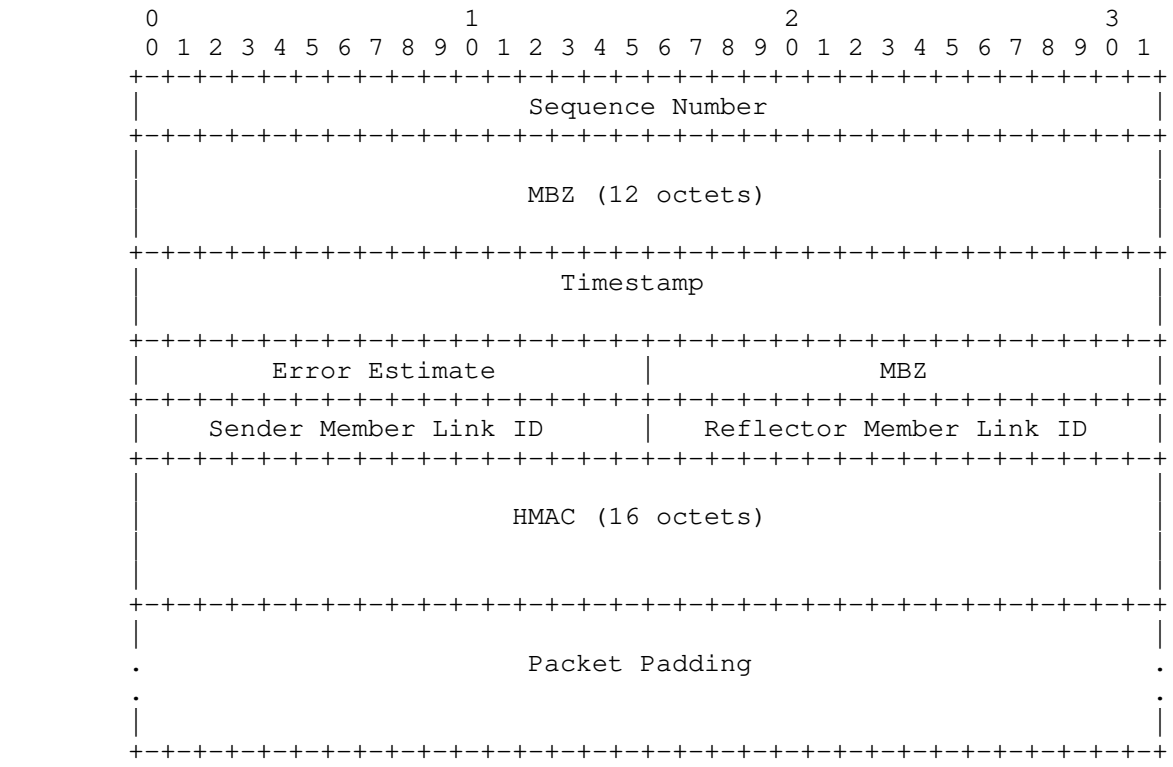


Figure 3: Session-Sender Packet Format in Authenticated Mode

Except for the Sender/Reflector Member Link ID field, all the other fields are the same as defined in Section 4.1.2 of TWAMP [RFC5357], which is originally defined in Section 4.1.2 of OWAMP [RFC4656]. Therefore, it follows the same procedure and guidelines as defined in Section 4.1.2 of TWAMP [RFC5357].

Sender Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Sender side. The value of the Sender Member Link ID MUST be unique at the Session-Sender.

Reflector Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Reflector side. The value of the Reflector Member ID MUST be unique at the Session-Reflector.

4.2.2. Reflector Behavior

The micro TWAMP Session-Reflector inherits the behaviors of a TWAMP Session-Reflector as defined in Section 4.2 of [RFC5357].

In addition, when receives a Test packet, the micro TWAMP Session-Reflector MUST use the receiving member link to correlate the Test packet to a micro TWAMP session. If there is no such a session, the Test packet MUST be discarded. If Reflector Member Link ID is not zero, the Reflector MUST use the Reflector member link identifier to check whether it associates with the receiving member link. If it does not, the Test packet MUST be discarded.

When sends a response to the received Test packet, the micro TWAMP Session-Sender MUST copy the Sender member link identifier from the received Test packet and put it in the Sender Member Link ID field of the reflected Test packet (see Figure 4 and Figure 5). In addition, the micro TWAMP Session-Reflector MUST fill the Reflector Member Link ID field (see Figure 2 or Figure 3) of the reflected Test packet with the member link identifier that are associated with the micro TWAMP session.

4.2.2.1. Packet Format and Content

The micro TWAMP Session-Reflector packet format is based on the TWAMP Session-Reflector packet format as defined in Section 4.2.1 of [RFC5357]. In addition, in order to carry the LAG member link identifier, two new fields (Sender and Reflector Member Link ID) are added. The formats are as below:

For unauthenticated mode:

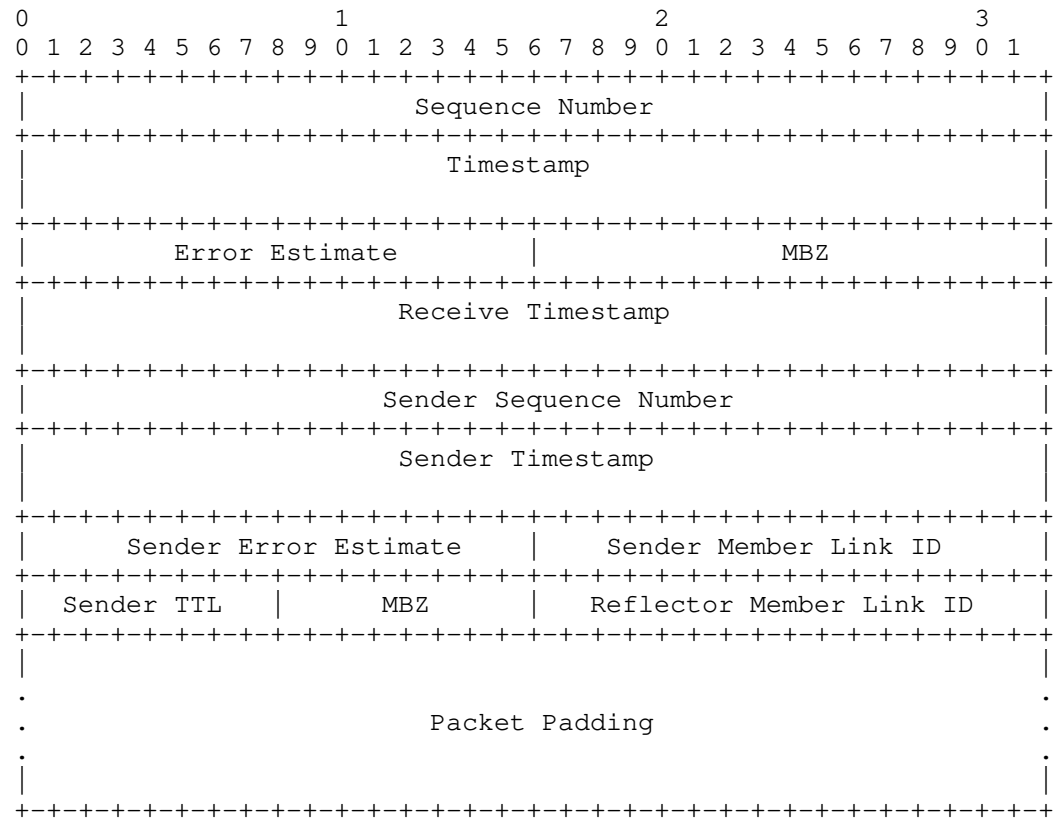
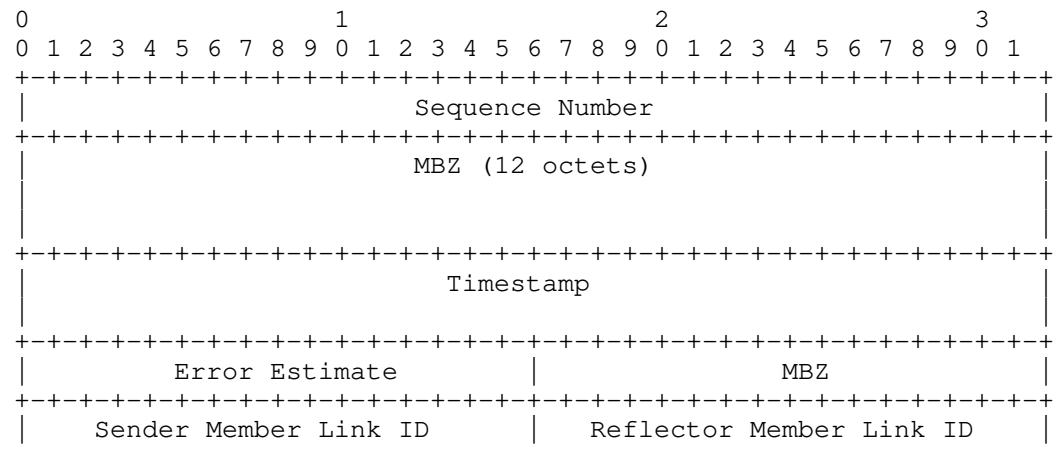


Figure 4: Session-Reflector Packet Format in Unauthenticated Mode

For authenticated and encrypted modes:



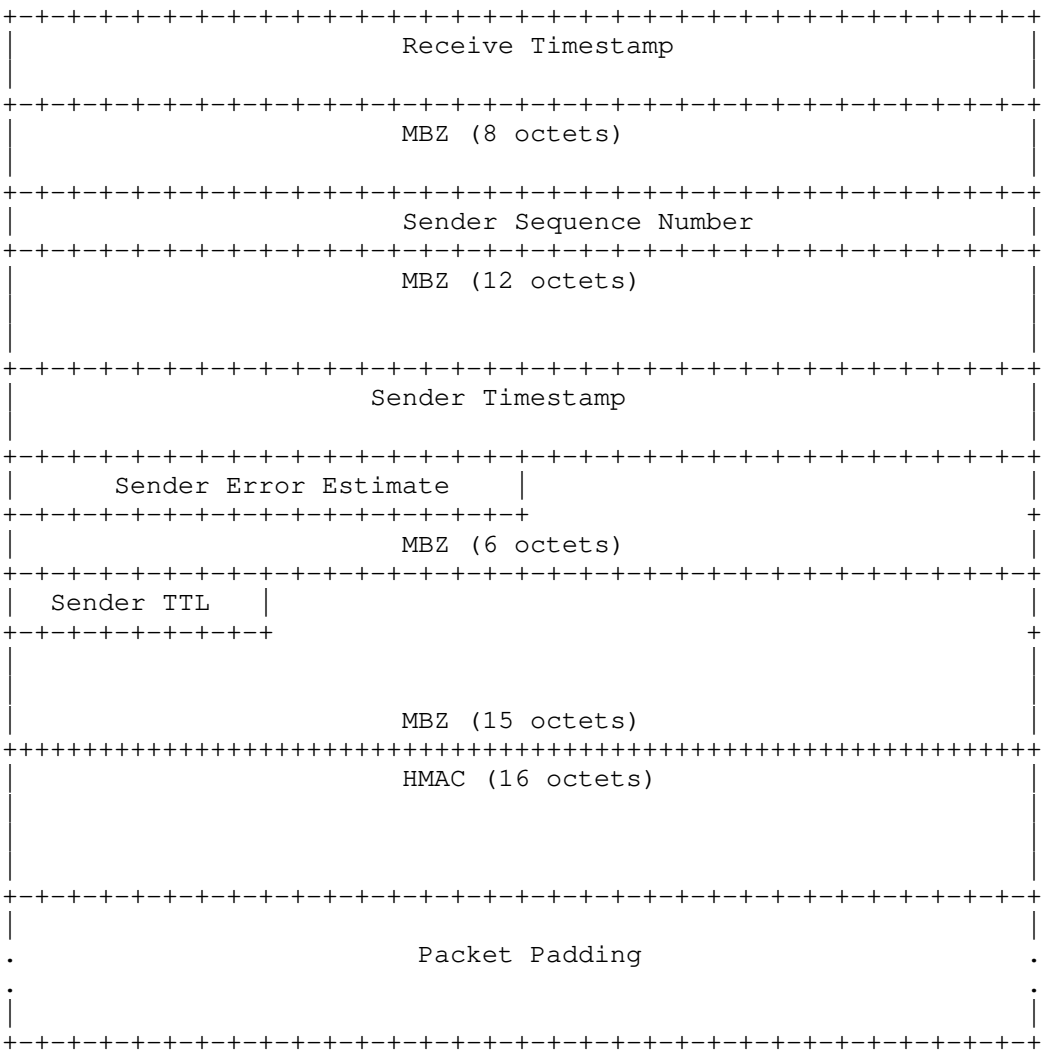


Figure 5: Session-Reflector Packet Format in Authenticated Mode

Except for the Sender/Reflector Member Link ID field, all the other fields are the same as defined in Section 4.2.1 of TWAMP [RFC5357]. Therefore, it follows the same procedure and guidelines as defined in Section 4.2.1 of TWAMP [RFC5357].

Sender Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Sender side. The value of the Sender Member Link ID MUST be unique at the Session-Sender.

Reflector Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Reflector side. The value of the Reflector Member ID MUST be unique at the Session-Reflector.

5. Mirco STAMP Session

5.1. Micro STAMP-Test

The micro STAMP-Test protocol is based on the STAMP-Test protocol [RFC8762] and [I-D.ietf-ippm-stamp-option-tlv] with the following extensions.

5.1.1. Session-Sender Packet Format

The micro STAMP Session-Sender Test packet formats are based on the STAMP Session-Sender Test packet formats and with some extensions, two new fields (Sender and Reflector Member Link ID) are added. The formats are as follows:

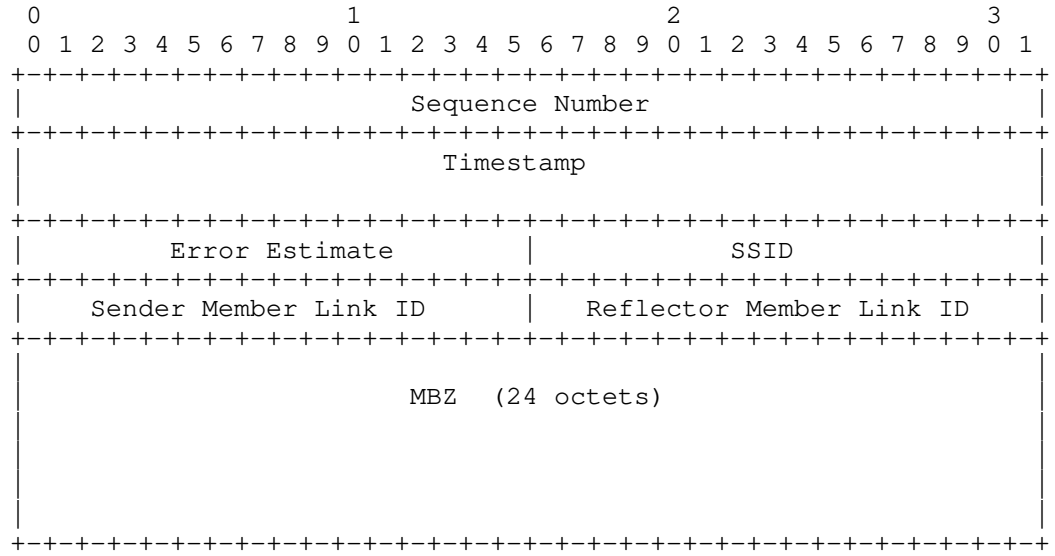


Figure 6: Session-Sender Test Packet in Unauthenticated Mode

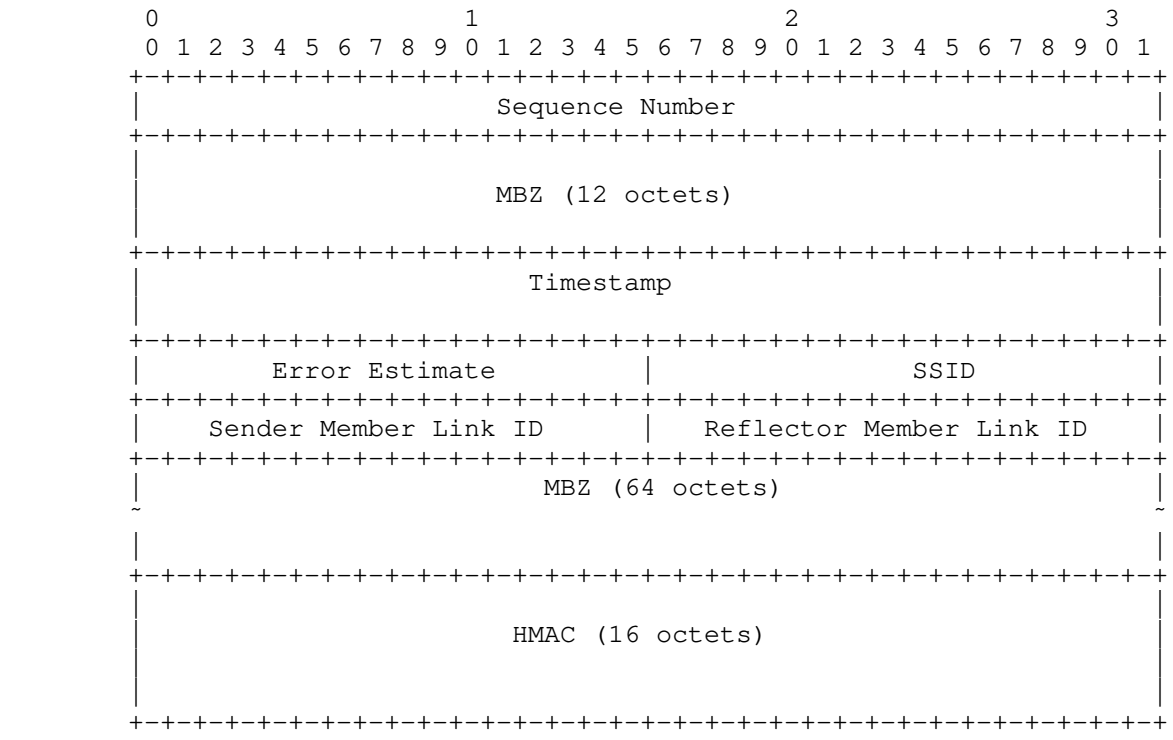


Figure 7: Session-Sender Test Packet in Authenticated Mode

Except for the Sender/Reflector Member Link ID fields, all the other fields are as defined in STAMP [RFC8762] and [I-D.ietf-ippm-stamp-option-tlv].

Sender Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Sender side. The value of the Sender Member Link ID MUST be unique at the Session-Sender.

Reflector Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Reflector side. The value of the Reflector Member ID MUST be unique at the Session-Reflector.

5.1.2. Session-Reflector Packet Format

The micro STAMP Session-Reflector Test packet formats are based on the STAMP Session-Reflector Test packet formats with some minor extensions, two new fields (Sender and Reflector Member Link ID) are added. The formats are as follows:

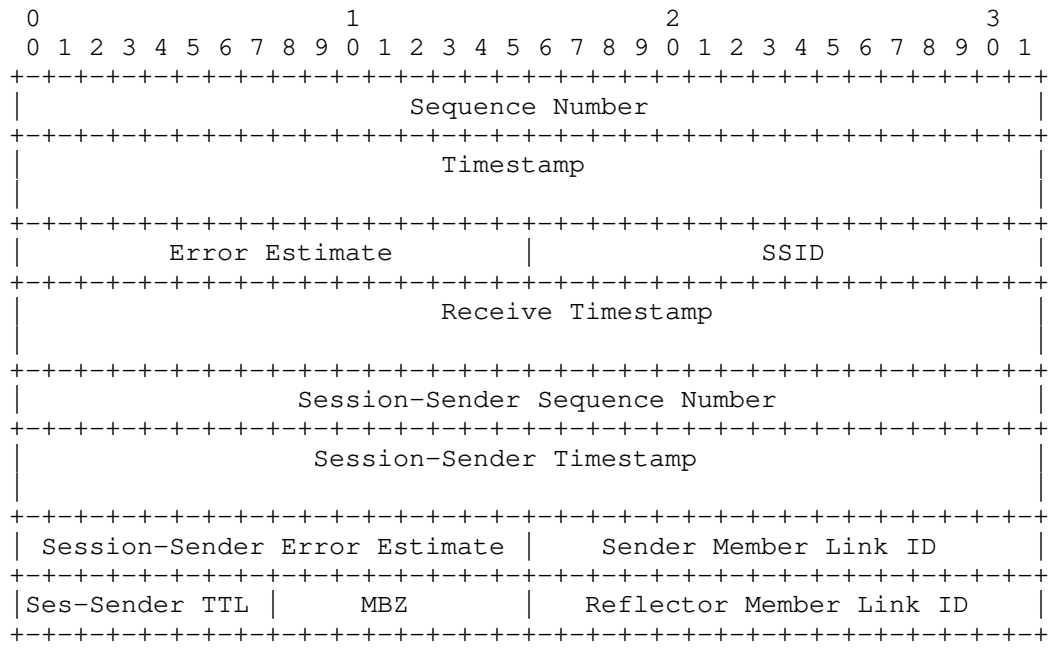


Figure 8: Session-Reflector Test Packet in Unauthenticated Mode

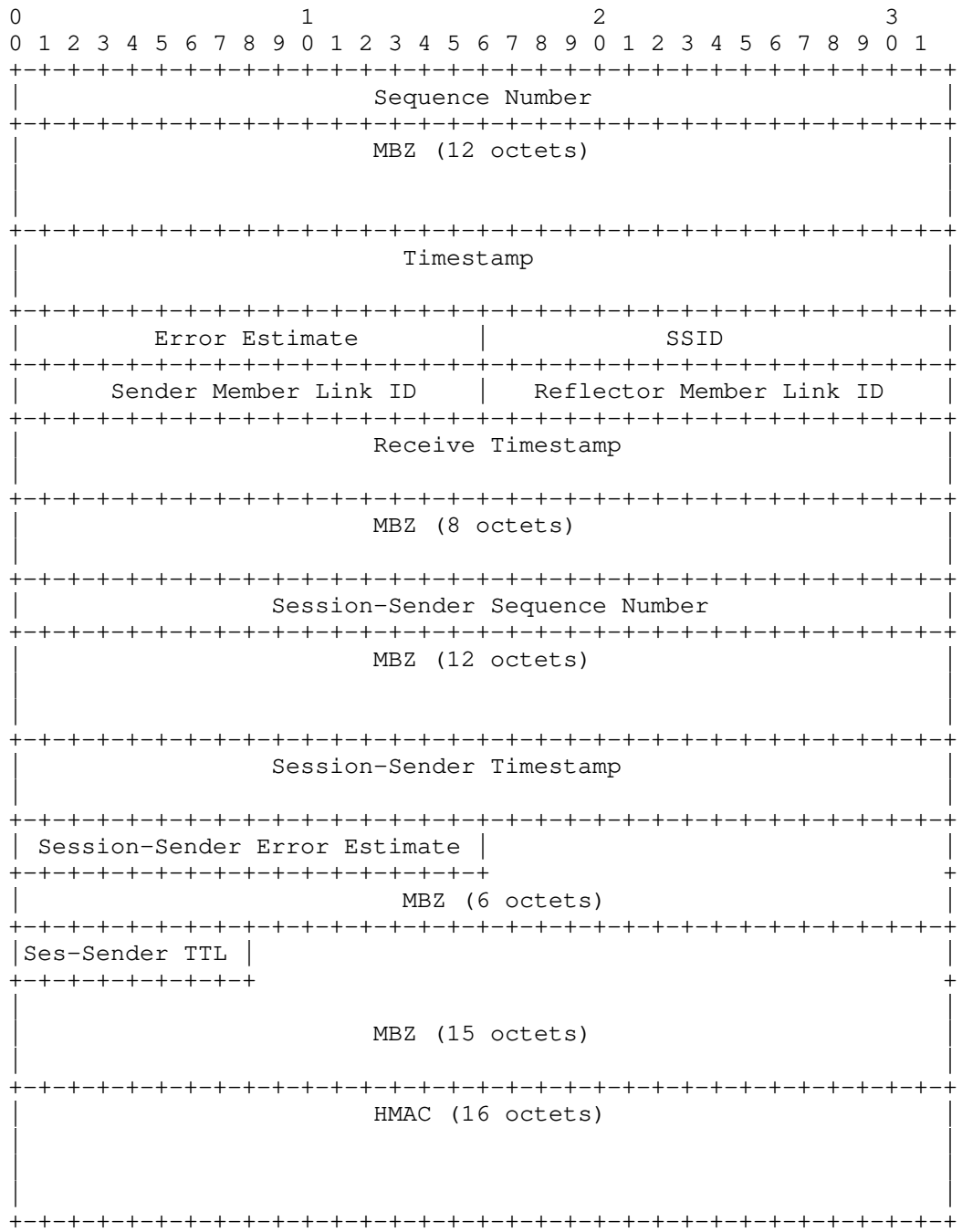


Figure 9: Session-Reflector Test Packet in Authenticated Mode

Except for the Sender/Reflector Member Link ID fields, all the other fields are as defined in STAMP [RFC8762] and [I-D.ietf-ippm-stamp-option-tlv].

Sender Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Sender side. The value of the Sender Member Link ID MUST be unique at the Session-Sender.

Reflector Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Reflector side. The value of the Reflector Member ID MUST be unique at the Session-Reflector.

5.1.3. Micro STAMP-Test Procedures

The micro STAMP-Test reuses the procedures as defined in Section 4 of STAMP [RFC8762] with the following additions:

The micro STAMP Session-Sender MUST send the micro STAMP-Test packets over the member link with which the session is associated.

The configuration and management of the mapping between a micro STAMP session and the Sender/Reflector member link identifiers are outside the scope of this document.

When sending a Test packet, the micro STAMP Session-Sender MUST set the Sender Member Link ID field with the member link identifier that is associated with the micro STAMP session. If the Session-Sender knows the Reflector member link identifier, it MUST put it in the Reflector Member Link ID field (see Figure 6 or Figure 7). Otherwise, the Reflector Member Link ID field MUST be set to zero.

The Sender member link identifier is used by the Session-Sender to check whether a reflected Test packet is received from the member link that associates with the correct micro STAMP session. The Reflector member link identifier is used by the Session-Receiver to check whether a Test packet is received from the member link that associates with the correct micro STAMP session.

The Reflector member link identifier can be obtained from pre-configuration or learned through data plane (e.g., learned from a reflected Test packet). How to abtain/learn the Reflector member link identifier is out of the scope of this document.

When receives a Test packet, the micro STAMP Session-Reflector MUST use the receiving member link to correlate the Test packet to a micro STAMP session. If there is no such a micro STAMP session, the Test packet MUST be discarded. If the Reflector Member Link ID is not zero, the micro STAMP Session-Reflector MUST use the Reflector member

link identifier to check whether it associates with the micro STAMP session. If it does not, the Test packet MUST be discarded and no reflected Test packet will be sent back the Session-Sender. If all validation passed, the Session-Reflector sends a reflected Test packet to the Session-Sender. The micro STAMP Session-Reflector MUST put the Sender and Reflector member link identifiers that are associated with the micro STAMP session in the Sender Member Link ID and Reflector Member Link ID fields (see Figure 8 and Figure 9) respectively. The Sender member link identifier is copied from the received Test packet.

When receives a reflected Test packet, the micro STAMP Session-Sender MUST use the receiving member link to correlate the reflected Test Packet to a micro STAMP session. If there is no such a session, the reflected Test packet MUST be discarded. If a matched micro STAMP session exists, the Session-Sender MUST use the identifier carried in the Sender Member Link ID field to check whether it associates with the session. If the checking failed, the Test packet MUST be discarded.

6. IANA Considerations

6.1. Mico OWAMP-Control Command

This document requires the IANA to allocate the following command type from OWAMP-Control Command Number Registry.

Value	Description	Semantics Definition
TBD1	Request-OW-Micro-Session	This document, Section 3.1

6.2. Mico TWAMP-Control Command

This document requires the IANA to allocate the following command type from TWAMP-Control Command Number Registry.

Value	Description	Semantics Definition
TBD1	Request-TW-Micro-Session	This document, Section 4.1

7. Security Considerations

The security considerations in [RFC4656], [RFC5357], [RFC8762] apply to this document.

8. Acknowledgements

The authors would like to thank Min Xiao, Fang Xin for the valuable comments to this work.

9. References

9.1. Normative References

[I-D.ietf-ippm-stamp-option-tlv]
Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-way Active Measurement Protocol Optional Extensions", draft-ietf-ippm-stamp-option-tlv-09 (work in progress), August 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[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>.

9.2. Informative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[IEEE802.1AX]
IEEE Std. 802.1AX, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", November 2008.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

Authors' Addresses

Zhenqiang Li
China Mobile

Email: li_zhenqiang@hotmail.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

G. Mirsky
X. Min
ZTE Corp.
February 22, 2021

Error Performance Measurement in Packet-switched Networks
draft-mirsky-ippm-epm-02

Abstract

This document describes the use of the error performance metric to characterize a packet-switched network's conformance to the pre-defined set of performance objectives. In this document, metrics that characterize error performance in a packet-switched network (PSN) are defined, as well as methods to measure and calculate them. Also, the requirements for an active Operation, Administration, and Maintenance protocol to support the error performance measurement in PSN are discussed, and potential candidate protocols are analyzed. All metrics and measurement methods are equally applicable to underlay and overlay networks.

Table of Contents

1. Introduction

Operations, Administration, and Maintenance (OAM) is a collection of methods to detect, characterize, localize failures in a network, and monitor the network's performance using various measurement methods. Traditionally, the former set of OAM tools identified as Fault Management (FM) OAM. The latter - Performance Monitoring (PM) OAM. Some OAM protocols can be used for both groups of tasks, while some serve one particular group. But regardless of how many OAM protocols are in use, network operators and network users are faced with multiple metrics that characterize the network conditions. This document describes a new component of packet-switched network (PSN) OAM.

Error performance measurement (EPM) is a part of an OAM toolset that provides an operator with information related to network measurements for a uni-directional or a bidirectional connection between two systems. In current technology, EPM has been defined only for data communication methods that have a constant bit-rate transmission [ITU.G.826] and not for PSN, where transmissions are statistically random. As a statistically multiplexed network in a PSN, a receiver node does not expect a packet to arrive from a sender node at a specific moment, less from a particular sender. That is what

differentiates PSN from networks built on a constant bit-rate transmission, where a stream of bits between two nodes is always present, whether it represents data or not. That provides the receiver with a predictable number of measurements in a series of measurement intervals. In PSN, on-path OAM methods, i.e., measurement methods that use data flow, cannot provide such predictability and thus be used for EPM. In PSN, EPM needs to use active OAM methods, per definition in [RFC7799]. This document identifies metrics that characterize PSN error performance and methods to measure and calculate them. Also, the requirements for an active OAM protocol to support EPM in PSN are discussed, and potential candidate protocols are analyzed.

2. Conventions used in this document

2.1. Terminology and Acronyms

OAM Operations, Administration, and Maintenance

EP Error Performance

EPM Error Performance Measurement

ES Errored Second

ESR Errored Second Ratio

SES Severely Errored Second

SESR Severely Errored Second Ratio

EFS Error-Free Second

PSN Packet-switched Network

FM Fault Management

PM Performance Monitoring

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. Error Performance Metrics

When analyzing the error performance of a path between two nodes, we need to select a time interval as the unit of EPM. In [ITU.G.826], a time interval of one second is used. It is reasonable to use the same time interval for EPM for PSNs. Further, for the purpose of EPM, each time interval, i.e., second, is classified either as Errored Second (ES), Severely Errored Second (SES), or Error-Free Second (EFS). These are defined as follows:

- An ES is a time interval during which at least one of the performance parameters degraded below its optimal level threshold or a defect was detected.

- An SES is a time interval during which at least one the performance parameters degraded below its critical threshold or a defect was detected.

- Consequently, an EFS is a time interval during which all performance objectives are at or above their respective optimal levels, and no defect has been detected.

The definition of a state of a defect in the network is also necessary for understanding the EPM. In this document, the defect is interpreted as the state of inability to communicate between a particular set of nodes. It is important to note that it is being defined as a state, and thus, it has conditions that define entry into it and exit out of it. Also, the state of defect exists only in connection to the particular group of nodes in the network, not the network as a domain.

3.1. Measure Error Performance Metrics

The definitions of ES, SES, and EFS allow for characterization of the communication between two nodes relative to the level of required and acceptable performance and when performance degrades below the acceptable level. The former condition in this document referred to as network availability. The latter - network unavailability. Based on the definitions, SES is the one-second of network unavailability while ES and EFS present an interval of network availability. But since the conditions of network are everchanging periods of network availability and unavailability need to be defined with duration larger than one-second interval to reduce the number of state changes while correctly reflecting the network condition. The method to determine the state of the network in terms of EPM OAM is described below:

- If ten consecutive SES intervals been detected, then the EPM state of the network determined as unavailability and the beginning of that period of unavailability state is at the start of the first SES in the sequence of the consecutive SES intervals.

- Similarly, ten consecutive non-SES intervals, i.e., either ES or EFS, indicate that the network is in the availability period, i.e., available. The start of that period is at the beginning of the first non-SES interval.

- Resulting from these two definitions, a sequence of less than ten consecutive SES or non-SES intervals does not change the EPM state of the network. For example, if the EPM state is determined as unavailability, a sequence of seven EFS intervals is not viewed as an availability period.

3.2. Calculate Error Performance Metrics

Determining the period in which the path is currently EP-wise is helpful. But because switching between periods requires ten consecutive one-second intervals, conditions that last shorter intervals may not be adequately reflected. Two additional EP OAM metrics can be used, and they are defined as follows:

- errored second ratio (ESR) is the ratio of ES to the total number of seconds in a time of the availability periods during a fixed measurement interval.

- severely errored second ratio (SESR) - is the ratio of SES to the total number of seconds in a time of the availability periods during a fixed measurement interval.

4. Requirements to EPM

TBA

5. Active OAM Protocol for EPM

Digital communication methods characterized as the constant-bit rate digital paths and connections allow measurement of the error performance without using an active OAM. That is possible because a predictable flow of digital signals is expected at an egress system. That is not the case for packet-switched networks that are based on the principle of statistical multiplexing flows. The latter usually improves the utilization of the communication network's resources, but it also makes the flow unpredictable for the egress system. For that reason, an active OAM has to be used in measuring the error performance in a network. A combination of OAM protocols can provide

the necessary for EPM functionality. For example, Bidirectional Forwarding Detection (BFD) [RFC5880] can be used to monitor the continuity of a path between the ingress and egress systems. And STAMP [RFC8762] can be used to measure and calculate performance metrics that are used as Service Level Objectives. But using two protocols and correlating the state of the network from them adds to the complexity in network operation.

The Integrated OAM, described in [I-D.mmm-rtgwg-integrated-oam], combines lightweight FM OAM with the comprehensive set of performance measurement methods. PM component of the Integrated OAM is based on [RFC6374] that supports, among other measurement methods, one-way and two-way packet loss and packet delay measurements.

6. IANA Considerations

TBA

7. Security Considerations

TBA

8. Acknowledgments

TBA

9. References

9.1. Normative References

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[I-D.mmm-rtgwg-integrated-oam] Mirsky, G., Min, X., and G. Mishra, "Integrated Operation, Administration, and Maintenance", draft-mmm-rtgwg-integrated-oam-00 (work in progress), February 2021.

[ITU.G.826]
ITU-T, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T G.826, December 2002.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <https://www.rfc-editor.org/info/rfc6374>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[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>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

IPPM
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2021

H. Song
Futurewei
Z. Li
S. Peng
Huawei Technologies
J. Guichard
Futurewei

January 4, 2021

Approaches on Supporting IOAM in IPv6
draft-song-ippm-ioam-ipv6-support-02

Abstract

It has been proposed to encapsulate IOAM tracing option data fields in IPv6 HbH options header. However, due to size of the trace data and the extension header location in the IPv6 packets, the proposal creates practical challenges for implementation, especially when other extension headers, such as a routing header, also exist and require in-network processing. We propose several alternative approaches to address this challenge, including separating the IOAM trace data to a different extension header, using the postcard-based telemetry (e.g., IOAM DEX) instead, and applying the segment IOAM trace export scheme, based on the network scenario and application requirements. We discuss the pros and cons of each approach and hope to foster standard convergence and industry adoption.

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 8, 2021.

Table of Contents

1. Introduction

In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] defines two tracing options, pre-allocated tracing option and incremental tracing option, which record hop-by-hop data along a packet's forwarding path. The draft [I-D.ietf-ippm-ioam-ipv6-options] proposes the method to encapsulate IOAM trace option data fields in IPv6. Because the tracing options requires per hop processing, such options can only be encapsulated in IPv6 Hop-by-Hop (HbH) options header. The draft [I-D.ioametal-ippm-6man-ioam-ipv6-deployment] further describes some deployment approaches.

[RFC8200] mandates that the HbH options header, if exists, must be the first extension header following the IPv6 header. However, the IOAM trace data can be large, which easily amount to tens to hundreds of bytes, making accessing other headers after it difficult or even impossible. There are practical limitations on how far the hardware

can reach into a packet in forwarding hardware. The IOAM tracing option cannot be applied if it makes other extension headers inaccessible. Even if the other headers can be reached, the deeper they are, the higher the cost to access and process them, and the lower the forwarding performance. A potentially more detrimental issue is that the incremental tracing option will expand the HbH header at each hop and push back all other headers, which keeps shifting the locations of the other extension headers, further complicating the hardware implementation and impeding the forwarding.

The issue becomes more severe when SRv6 and IOAM coexist. The Segment Routing Extension Header (SRH) [I-D.ietf-6man-segment-routing-header] is encapsulated in a routing header which is after the HbH options header. SRH itself can be large. It requires read and write operations at each SRv6 node. If it is deeply embedded in a packet and its location keeps shifting, either it is beyond the reach of hardware or the forwarding performance degrades.

We can avoid the problem by simply not using both at the same time, but apparently this is not ideal, because IOAM is an important OAM tool and it is even more wanted when SRv6 brings more operational complexity into IPv6 networks.

Our second recourse is to limit the IOAM to SRv6 nodes only. That is, consider SRv6 as an overlay tunnel over IPv6 and apply the IOAM pipe mode as discussed in [I-D.song-ippm-ioam-tunnel-mode], which only collects data at each SRv6 nodes. To realize this, [I-D.ali-spring-ioam-srv6] describes an approach that encapsulates the IOAM option data fields in an SRH TLV. [I-D.song-6man-srv6-pbt] and [I-D.ietf-6man-spring-srv6-oam] describe another approach to enable postcard-based telemetry for SRv6 without needing IOAM option encapsulation. In either case, the SRH is close to the packet front and its location is fixed. [I-D.song-spring-siam] proposes to support IOAM in the payload of the dedicated SRv6 probing packets only. While these approaches are useful for use cases that only need to monitor the segment end points, it fails to cover all the IPv6 nodes in a network.

So the proposition of this draft is, suppose we need to apply IOAM on all nodes in an SRv6 network, how we can amend the approach in [I-D.ietf-ippm-ioam-ipv6-options] or use alternative approaches to circumvent the aforementioned issues. In this draft, we propose three such approaches: (1) separating the IOAM trace data to a different extension header, (2) using the postcard-based telemetry (e.g., IOAM DEX) instead, and (3) applying the segment IOAM trace export scheme. We discuss the pros and cons of each approach and hope to foster standard convergence and industry adoption.

2. IOAM Data Separate and Postpose

An IOAM trace type data fields contain two parts: instruction and trace data. Although by convention the trace data part immediately follow the instruction part, there is not fundamental reason why these two parts must stick together. This observation provides us an optimization opportunity to amend the original proposal in [I-D.ietf-ippm-ioam-ipv6-options].

We separate the IOAM trace type data fields into the instruction part and the trace data part. We encapsulate only the instruction part in the HbH options header, and encapsulate the trace data part in another metadata extension header after all the IPv6 extension headers and before upper layer protocol headers. This arrangement allows high performance hardware implementation. When using the incremental data trace, even if the data trace size increases at each node, all IPv6 extension headers remain intact and new data is inserted at a fixed location.

Figure 1 shows the HbH option format for IOAM trace type instruction. The field specification is identical to that in [RFC8200] and [I-D.ietf-ippm-ioam-data].

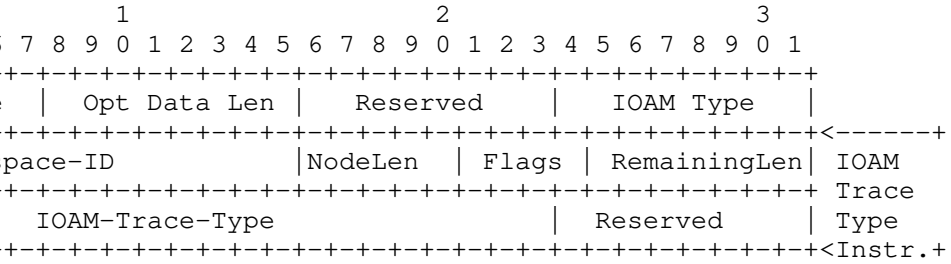


Figure 1: HbH Option Format for IOAM Trace Type Instruction

Figure 2 shows the TLV option format for IOAM trace type data. The IOAM trace type data format is compliant with [I-D.ietf-ippm-ioam-data].

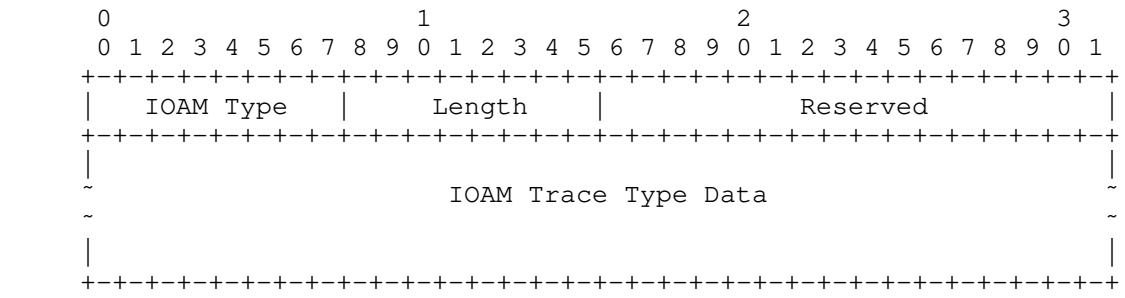


Figure 2: Option Format for IOAM Trace Type Data

2.1. IOAM Trace Data Encapsulation

We have basically two methods to encapsulate the IOAM trace data. First, we can define a new IPv6 extension header which is dedicated to metadata. Once standardized, this extension header can also be used to host potential metadata from other applications such as NSH for SFC [RFC8300]. Second, this option can be carried as a TLV option in another existing extension header such as the destination header. The only requirement is that this extension header should be the last one in the extension header chain. The first method is cleaner but it requires extra standard effort; the second method is simpler but it needs to overcome the access constraints exerted by [RFC8300].

3. Segment IOAM Data Export

If the overhead of the IOAM trace type data fields is under control, we may still manage to encapsulate both instruction and data in HbH options header as in [I-D.ietf-ippm-ioam-ipv6-options]. To this end, we introduce two sub-approaches.

3.1. Independent of SRv6

[I-D.song-ippm-segment-ioam] proposes an enhancement to IOAM trace type which can configure the allowable overhead of the IOAM trace type data fields. Once the trace data size is up to the limit at a network node (i.e., a segment or a fixed number of network nodes are traversed), the trace data will be stripped and exported so room is made to accommodate new trace data from nodes in the next segment of the forwarding path.

This approach requires some moderate updates to the IOAM trace type data fields, as described in [I-D.song-ippm-segment-ioam]. Figure 3 shows the format of the HbH Option Header containing Segment IOAM trace type data fields. A flag bit (#23) in the Flags field is used

to indicate the current header is a segment IOAM header. In this context, the last octet in the IOAM header is partitioned into two 4-bit nibbles. The first nibble (SSize) is used to save the segment size and the second nibble (RHop) is used to save the remaining hops. This limits the maximum segment size to 15.

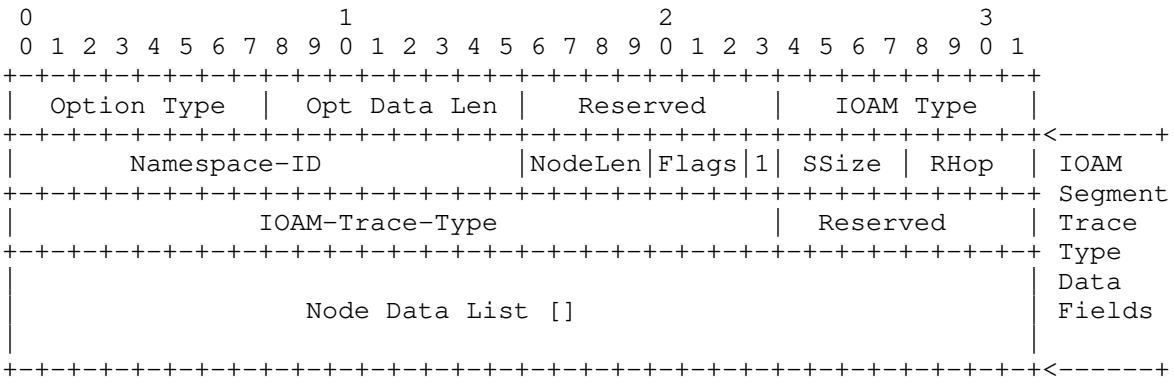


Figure 3: HbH Option Format for Segment IOAM Trace Type Data Fields

At the beginning of each segment, the segment size (SSize) and the remaining hops (RHop) are initialized: RHop is set to equal to SSize. At each hop, if RHop is not zero, the node data is added to the node data list and then RHop is decremented by 1. If RHop is equal to 0 when receiving the packet, the node needs to remove (in incremental trace option) or clear (in pre-allocated trace option) the IOAM node data list and reset RHop to SSize.

In this case, if we use the IOAM pre-allocated trace type, the size and location of each IPv6 extension header is fixed and predictable, and the hardware capability and performance can be guaranteed.

3.2. Export at SRv6 node

Whenever a packet with the IOAM option reaches a SRv6 node which needs to access the SRH, we can configure the node to export immediately the IOAM trace data accumulated so far. In this case, basically at each SRv6 node, the HbH header size is fixed and the header contains an IOAM option with only the instruction part. After the SRH processing, this node can add local IOAM trace data in the HbH option header before forwarding the packet.

The incremental trace type can be used in this approach. In an extreme case when every node is also an SRv6 node, this approach regresses to a per-hop postcard-based telemetry approach as described in [I-D.song-ippm-postcard-based-telemetry]. In this case, the HbH

option for IOAM can even be avoided altogether if we can find a way to simply mark the packet for postcard export.

4. Direct Export Option

As an embodiment of the PBT-I approach introduced in [I-D.song-ippm-postcard-based-telemetry], IOAM Direct Export (DEX) Option Type discussed in [I-D.ioamteam-ippm-ioam-direct-export] can be used to replace the IOAM trace type. IOAM DEX only needs to encapsulate a fix-size instruction header in the HbH option header.

Figure 4 shows the HbH option format for IOAM DEX type fields. The field specification is identical to that in [RFC8200] and [I-D.ioamteam-ippm-ioam-direct-export].

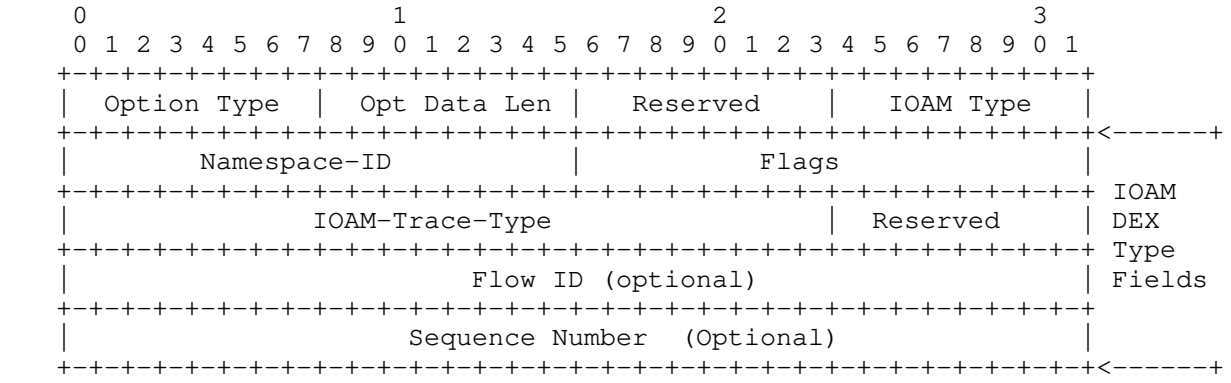


Figure 4: HbH Option Format for IOAM DEX Type Fields

5. Comparison

The following table compares the existing approach and the four other alternative approaches proposed in this draft.

Approach	Pros	Cons
IOAM Trace in HbH	Comply w/ IOAM Data Spec	Variable and long HbH header impeding access of later extension headers
IOAM Trace Data Separate and Postpose (Sec. 2)	Fix-size and short HbH header, good for later extension header access	Need extra extension header to hold trace data
Segment IOAM Data Export (Sec. 3.1)	Fix-size and controllable HbH header size	Need to enhance IOAM trace type data field spec.
Trace Export at SRv6 nodes (Sec. 3.2)	Can be done through configuration	Specific to SRv6; No better than PB & IOAM DEX in the worst case
IOAM Direct Export in HbH (Sec. 4)	Comply w/ IOAM DEX Spec; Fix-size and short HbH	Need export data correlation

Figure 5: Comparison of Different Approaches

6. Security Considerations

TBD.

7. Normative References

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Nainar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-03 (work in progress), November 2020.

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.

[I-D.ioametal-ippm-6man-ioam-ipv6-deployment]
Bhandari, S., Brockners, F., Mizrahi, T., Kfir, A., Gafni, B., Spiegel, M., Krishnan, S., and M. Smith, "Deployment Considerations for In-situ OAM with IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-deployment-03 (work in progress), March 2020.

[I-D.ioamteam-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.

[I-D.song-6man-srv6-pbt]
Song, H., "Support Postcard-Based Telemetry for SRv6 OAM", draft-song-6man-srv6-pbt-01 (work in progress), October 2019.

[I-D.song-ippm-ioam-tunnel-mode]
Song, H., Li, Z., Zhou, T., and Z. Wang, "In-situ OAM Processing in Tunnels", draft-song-ippm-ioam-tunnel-mode-00 (work in progress), June 2018.

[I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

[I-D.song-ippm-segment-ioam]
Song, H. and T. Zhou, "Control In-situ OAM Overhead with Segment IOAM", draft-song-ippm-segment-ioam-01 (work in progress), April 2018.

[I-D.song-spring-siam]
Song, H. and T. Pan, "SRv6 In-situ Active Measurement", draft-song-spring-siam-00 (work in progress), December 2020.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

Authors' Addresses

Haoyu Song
Futurewei
USA

Email: haoyu.song@futurewei.com

Zhenbin Li
Huawei Technologies
China

Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
China

Email: pengshuping@huawei.com

James Guichard
Futurewei
USA

Email: james.n.guichard@futurewei.com

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 26, 2021.

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:

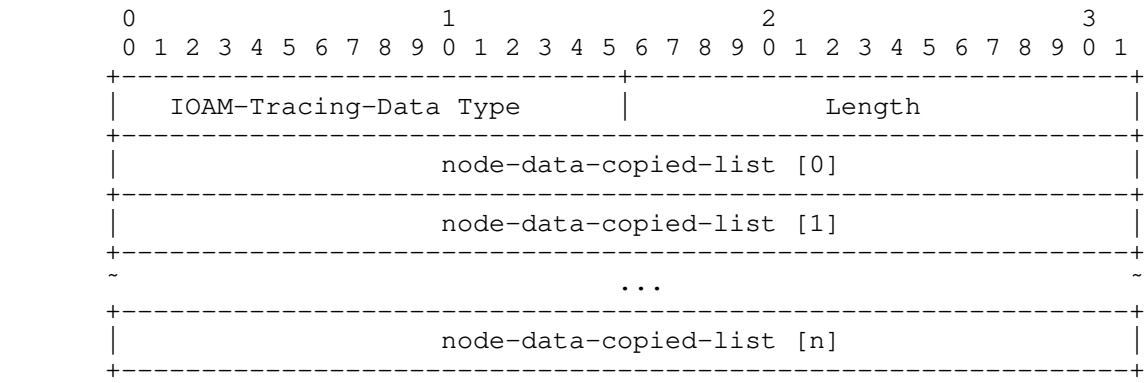


Fig. 1 IOAM-Tracing-Data TLV Format

where fields are defined as the following:

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

- 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.

- 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:

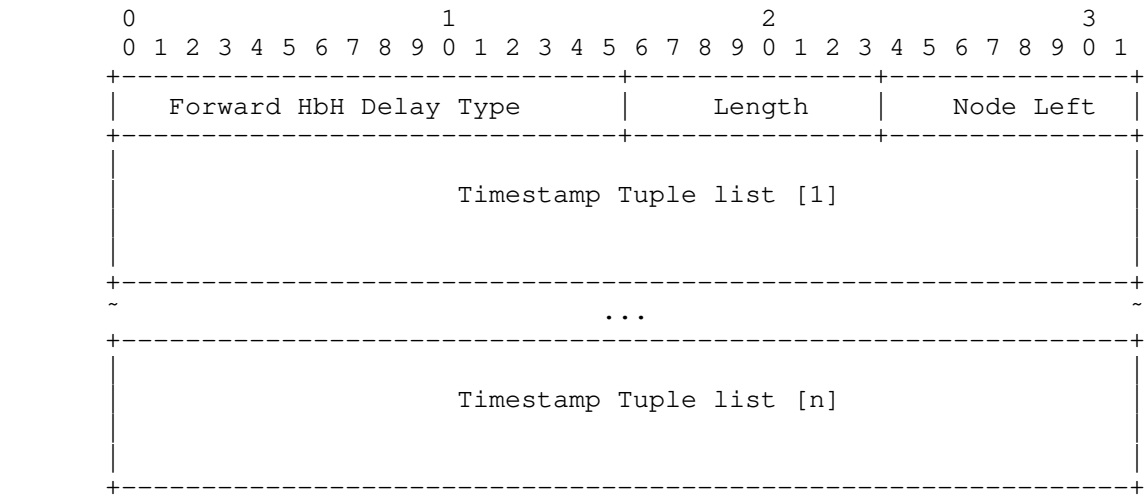


Fig. 2 Forward HbH Delay TLV Format

where fields are defined as the following:

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

- 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.

- 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.

- 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.

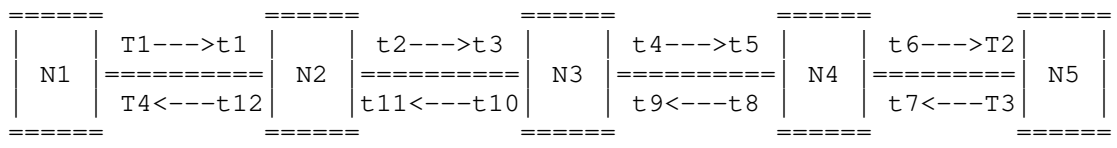


Fig. 3 Reference Topology

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:

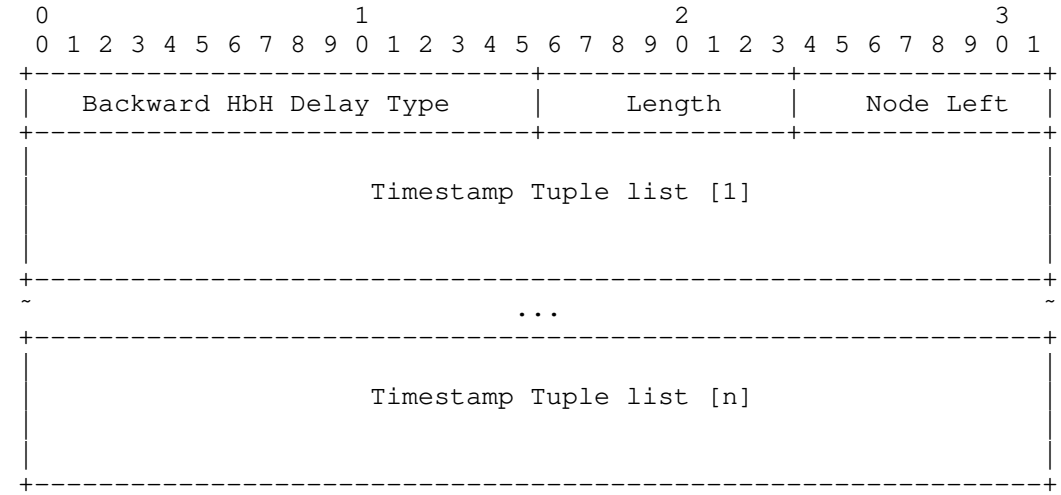


Fig. 4 Backward HbH Delay TLV Format

where fields are defined as the following:

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

- 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.

- 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.

- 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:

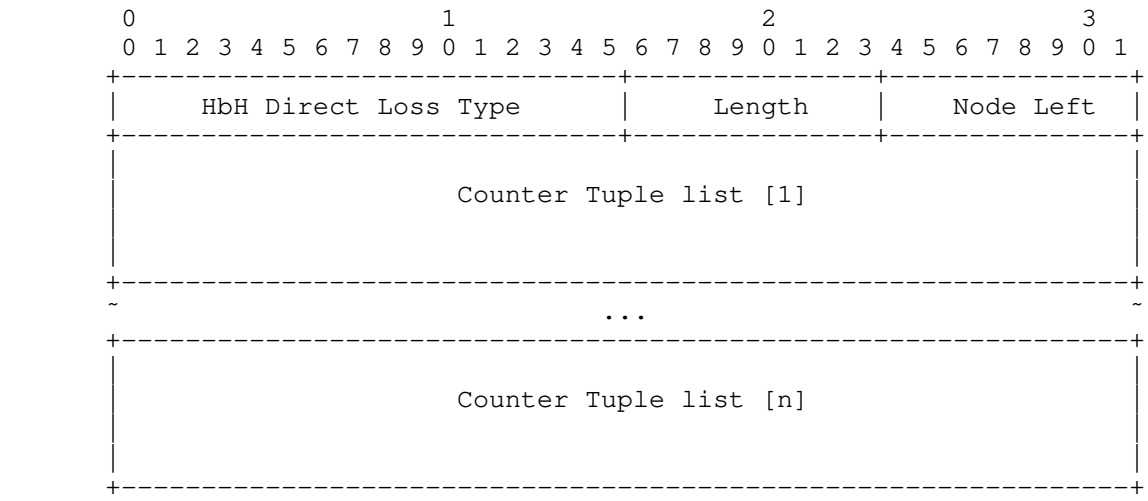


Fig. 5 HbH Direct Loss TLV Format

where fields are defined as the following:

- HbH Direct Loss Type: To be assigned by IANA.

- 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.

- 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.

- 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:

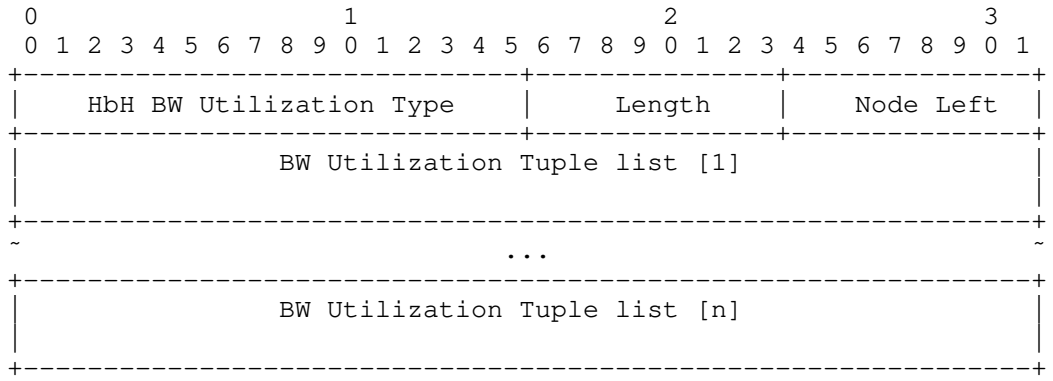


Fig. 6 HbH Bandwidth Utilization TLV Format

where fields are defined as the following:

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

- 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.

- 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.

- 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:

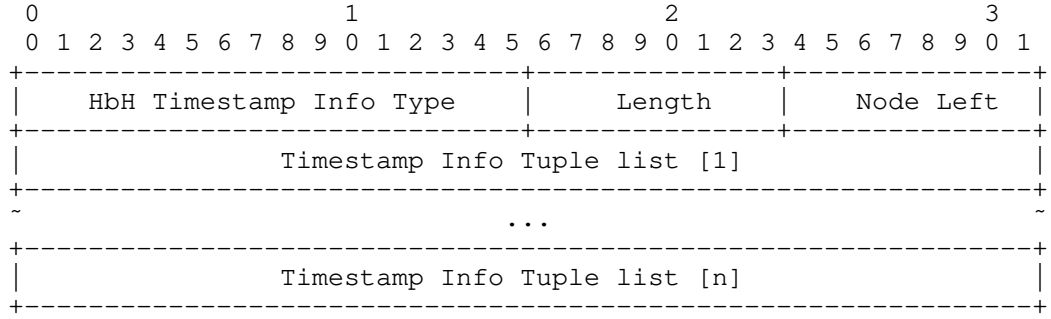


Fig. 6 HbH Timestamp Information TLV Format

where fields are defined as the following:

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

- 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.

- 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.

- 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:

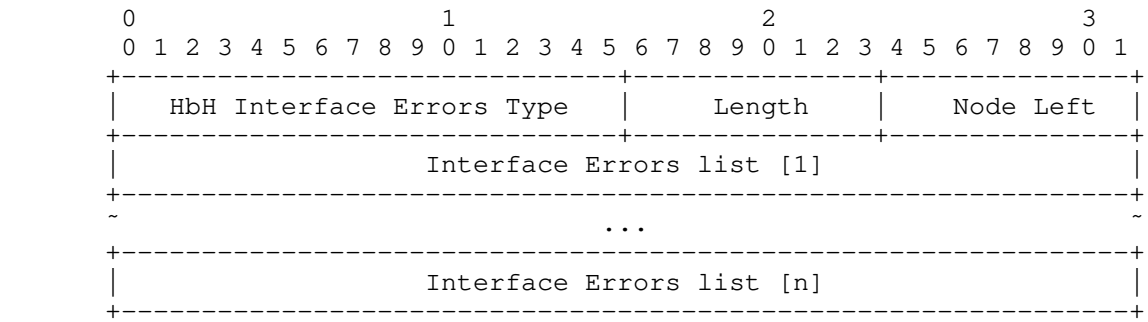


Fig. 6 HbH Timestamp Information TLV Format

where fields are defined as the following:

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

- 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.

- 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.

- 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

IP Performance Measurement
Internet-Draft
Intended status: Informational
Expires: January 13, 2021

H. Yang
T. Sun
K. Yao
China Mobile

July 12, 2020

Application Oriented High Precision Delay and Jitter Measurement
draft-yang-ippm-ptp-measurement-00

Abstract

As 5G has arisen and it is still evolving, there become many more time sensitive services which require high precision of measurements. In addition, in order to better simulate the transmission of packets of actual services, the length and priorities of the measurement packets SHOULD be customized, especially for some network that is inclined to get congested. This document introduces a new way to measure the time delay and jitter between two devices by making adjustments based on PTP Sync message and Delay_Req message, which could, to a great extent, get close to the messages of actual services as well as achieve high precison of measured metrics, so as to meet all requirements mentioned above.

Table of Contents

1. Introduction

The precision of some conventional ways used to measure the one-way or round-trip delay and jitter, including ICMP (ping command) and Iperf, a measurement tool, is highly dependent on NTP[RFC5905]precision which is between millisecond and second. As 5G has arisen and it is still continuously evolving, many industrial scenarios, such as internet of vehicles, and other time sensitive sevices have new requirements for time precision which is in microsecond and even in nanosecond. With the growing support of Precision Time Protocol (PTP) [IEEE.1588.2008], in many industrial scenarios, such as industrial control network and video transmission network, devices can be synchronized in very high precision that is in sub-microsecond.

Although TWAMP has already supported PTP timestamp, as is stated in[RFC8186] , the current protocol doesn't allow customizing the length and priorities of packets. Since packets of actual services have different length and priorities, which MAY lead to different time delay, the measurement packets need to be designed to meet such requirements. This document introduces a new way to measure the time delay and jitter between two devices by making adjustments based on PTP Sync message and Delay_Req message, which could, to a great

extent, get close to the messages of actual services as well as achieve high precison of measured metrics, so as to meet all requirements mentioned above.

2. Conventions Used in This Document

2.1. Terminology

NTP Network Time Protocol

PTP Precision Time Protocol

TWAMP Two-Way Active Measurement Protocol

DSCP Differentiated Services Code Point

ICMP Internet Control Message Protocol

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. Adjustments on PTP Sync Message and Delay_Req Message

Since PTP Sync message and Delay_Req message are used to synchronize two devices, namely Source and Destination. In order to use these messages to measure time delay and jitter, this document introduces a new way by making some adjustments on original messages. Utilizing the modified PTP Sync message and Delay_Req message, people can measure the forward and backward time delay and jitter between two nodes respectively within the same synchonized network.

3.1. Adjustments on PTP Header

The adjustments can be done through setting the field, FlagField, in the PTP header. The format of the PTP header is shown in figure 1. There are two consecutive flag bits in the field, PTP profile Specific 1 and PTP profile Specific 2, whose default values are false. PTP profile Specific 1 takes up the sixth bit while PTP profile Specific 2 takes up the seventh bit. Both of values inside FlagField are changed to be true, as is shown in figure 2, indicating the message is not used for synchronization, but for measurement.

* PTP profile Specific 1: False -> True

* PTP profile Specific 2: False -> True

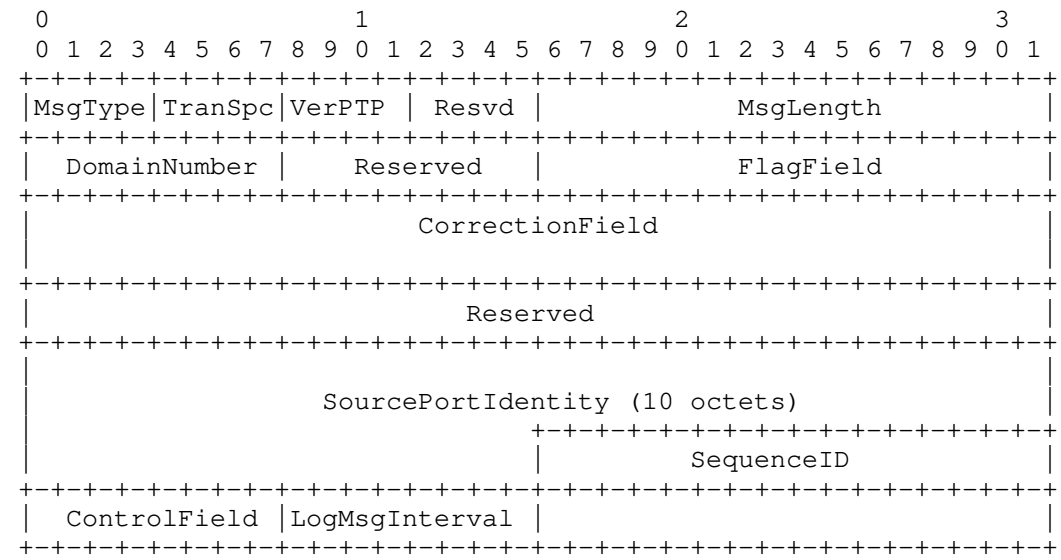


Figure 1: PTP Header Format

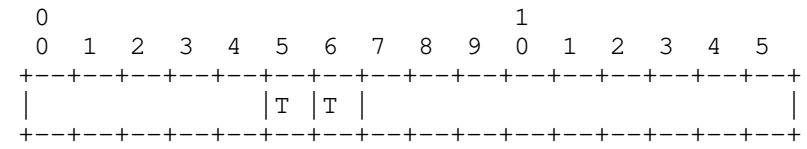


Figure 2: Modified FlagField

3.2. Customization of Length and Priority

Another feature of the modified PTP Sync message or Delay_Req message is that their length and priorities can be set manually in order to get close to the messages of actual services, and this is crucial for some time sensitive services. Customization of message length and priority can be done in adjustments below.

3.2.1. Length

The complete PTP Sync message or Delay_Req message is composed by three major parts, header, body, and suffix, as described in PTPv2 [IEEE.1588.2008] . The specification allows the suffix to be zero length if the message does not carry any information other than its timestamp. To simulate the transmission of messages of actual services, the suffix can be filled with extra bytes, and in this way,

the total length of this PTP Sync message or Delay_Req message can be the same as the actual one. Thereby, in the figure below, the suffix is labeled as 'customized'.

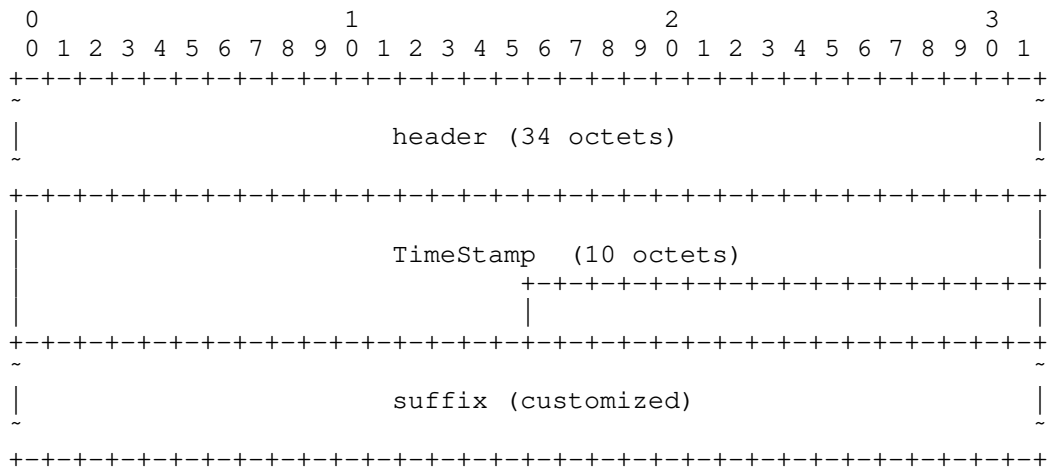


Figure 3: PTP Sync or Delay_Req Message Format

3.2.2. Priority

Priorities of packets are set in the DS field of IP header which is defined in [RFC2474] . The format of IP header is shown in the figure below where the DS field occupies the second octet. The first 6 bits of the DS field is named as DSCP, differentiated services codepoint, which are used to represent maximum 64 priorities.

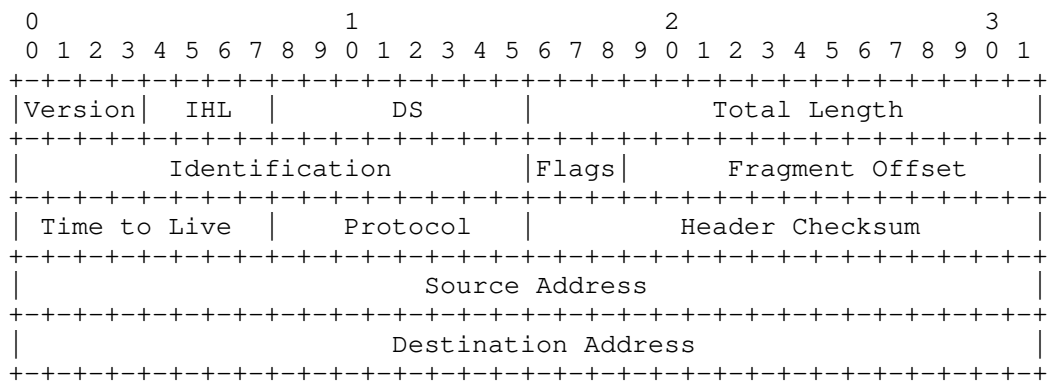


Figure 4: IP Header Format

The complete encapsulation of modified PTP Sync message or Delay_Req message by the UDP header, IP header, and Mac header is shown in the figure below, with their length and prioritities customized.

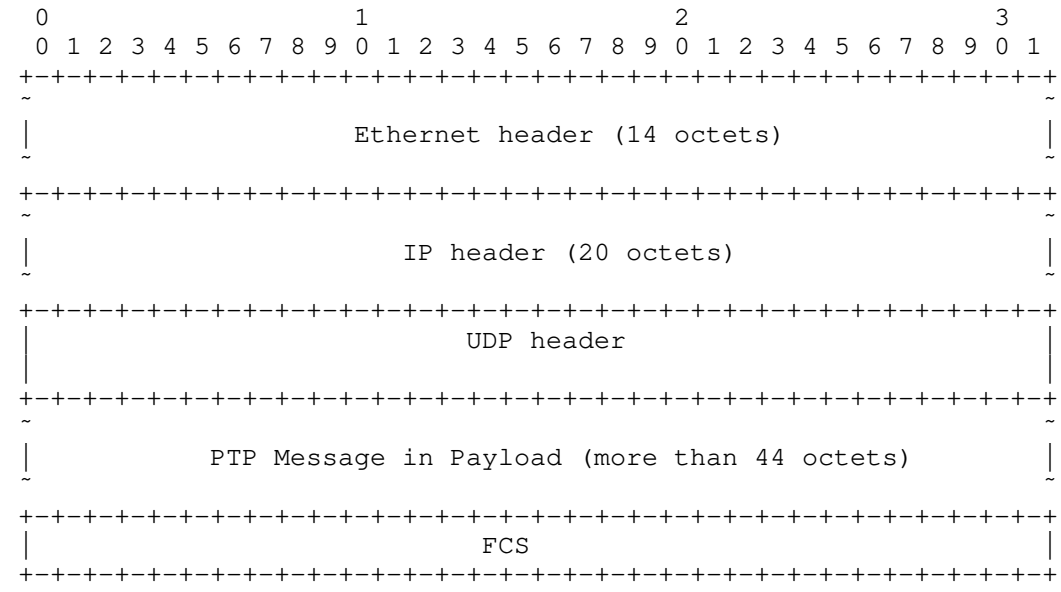


Figure 5: Format of PTP Message over UDP

4. Measurement Procedures

* First of all, the network to which both source and destination are connected needs to be synchronized globally.

* Before measuring the time delay and jitter between source and destination, the measurement mode of devices needs to be enabled and adjustments need to be done by following section 3. For the source device, the destination address needs to be configured and the PTP Sync message carrying the source timestamp MUST be converted into measurement mode by modifying the FlagField inside the message header. For the destination device, the address of receiver is configured as the source address and the PTP Delay_Req message carrying the destination timestamp MUST be turned into measurement mode too by following the same way.

* To measure the time delay and jitter of the forward path, the source device sends the modified PTP Sync message to the destination device. When packets are transmitted inside the middle network from source to destination, the nodes between them will check the IP address of destination. If it’s not the same as the local address,

then pass it to the next node.When packets are receieved by destination device, the measurement mode of the PTP Sync message can be decided by recognizing the FlagField inside its message header . Then the source timestamp and the arrival timestamp generated by destination device will be uploaded to CPUs in upper layers to count the difference of these two timestamps, which is just the time delay of the forward path. To measusre the forward jitter, the source needs to send the modified PTP Sync message to the destination for one more time, and the jitter is the difference of two consecutive time delays.

* To measure the time delay and jitter of the backward path, the destination device sends the modified PTP Delay_Req message to the source device. The message carries the local timestamp of the destination and it can be recognized by the source device with its adjusted message header. Upon receipt of the modified PTP Delay_Req message, the source device will generate a local timestamp and upload it together with the timestamp inside the message to CPUs in upper layers. Then the time delay will be achieved by calculating the difference of two timestamps and the backward jitter is the difference of two consecutive backward time delays.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Normative References

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <https://www.rfc-editor.org/info/rfc8186>.

Authors' Addresses

Hongwei Yang
China Mobile
Beijing 100053
China

Email: yanghongwei@chinamobile.com

Tao Sun
China Mobile
Beijing 100053
China

Email: suntao@chinamobile.com

Kehan Yao
China Mobile
Beijing 100053
China

Email: yaokehan@chinamobile.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: July 22, 2021

T. Zhou, Ed.
G. Fioccola
Huawei
S. Lee
LG U+
M. Cociglio
Telecom Italia
W. Li
Huawei

January 18, 2021

Enhanced Alternate Marking Method
draft-zhou-ippm-enhanced-alternate-marking-06

Abstract

This document extends the IPv6 alternate marking option to provide the enhanced capabilities.

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 22, 2021.

Table of Contents

1. Introduction

The Alternate Marking [RFC8321] and Multipoint Alternate Marking [I-D.ietf-ippm-multipoint-alt-mark] define the Alternate Marking technique that is an hybrid performance measurement method, per [RFC7799] classification of measurement methods. This method is based on marking consecutive batches of packets and it can be used to measure packet loss, latency, and jitter on live traffic.

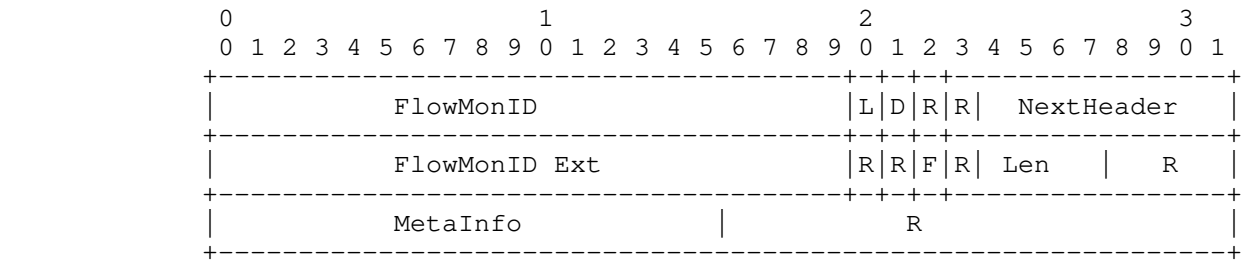
AltMark Option [I-D.ietf-6man-ipv6-alt-mark] applies the Alternate Marking Method for IPv6 protocol, and defines Extension Header Option to encode Alternate Marking Method for both Hop-by-Hop Options Header and Destination Options Header.

While the AltMark Option implement the basic alternate marking method, this document defines the extended data fields for the AltMark Option and provides the enhanced capabilities.

It is worth mentioning that the enhanced capabilities are intended for further use and are optional.

2. Data Fields Format

The following figure shows the data fields format for enhanced alternate marking. This data is expected to be encapsulated to specific transports.



where:

- o FlowMonID - Flow Monitoring Identification is the same as defined in AltMark Option [I-D.ietf-6man-ipv6-alt-mark].
- o L and D - Loss Flag and Delay Flag are the same as defined in AltMark Option [I-D.ietf-6man-ipv6-alt-mark].
- o NextHeader - Identify whether to carry the extended data fields.
- o FlowMonID Ext - 20 bits unsigned integer. This used to extend the FlowMonID to reduce the conflict when random allocation is applied
- o R - Reserved for further use. This bit MUST be set to zero.
- o F - Flow direction identification. F = 1, indicate the flow direction is forward.
- o Len - Length. It indicates the length of extension headers.
- o MetaInfo - A 16 bits Bitmap to indicate more meta data attached for the enhanced function.

3. Enhanced Alternate Marking capabilities

The extended data fields presented in the previous section can be used for several uses. Some possible applications can be:

1. shortest marking periods of single marking method for thicker packet loss measurements.
2. more dense delay measurements than double marking method (down to each packet).
3. increase the entropy of flow monitoring identifier by extending the size of FlowMonID.

4. and so on.

4. Security Considerations

TBD

5. IANA Considerations

This document has no request to IANA.

6. References

6.1. Normative References

[I-D.ietf-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate Marking method for passive and hybrid performance monitoring", draft-ietf-ippm-multipoint-alt-mark-09 (work in progress), March 2020.

[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>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

6.2. Informative References

[I-D.ietf-6man-ipv6-alt-mark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-02 (work in progress), October 2020.

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Shinyoung Lee
LG U+
71, Magokjungang 8-ro, Gangseo-gu
Seoul
Republic of Korea

Email: leesy@lguplus.co.kr

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

Weidong Li
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: poly.li@huawei.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: January 31, 2021

T. Zhou, Ed.
Huawei
J. Guichard
Futurewei
F. Brockners
S. Raghavan
Cisco Systems
July 30, 2020

A YANG Data Model for In-Situ OAM
draft-zhou-ippm-ioam-yang-08

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in user packets while the packets traverse a path between two points in the network. This document defines a YANG module for the IOAM function.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 31, 2021.

Table of Contents

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records OAM information within user packets while the packets traverse a network. The data types and data formats for IOAM data records have been defined in [I-D.ietf-ippm-ioam-data]. The IOAM data can be embedded in many protocol encapsulations such as Network Services Header (NSH) and IPv6.

This document defines a data model for IOAM capabilities using the YANG data modeling language [RFC7950]. This YANG model supports all the five IOAM options, which are Incremental Tracing Option, Pre-allocated Tracing Option, Direct Export Option[I-D.ietf-ippm-ioam-direct-export], Proof of Transit(PoT) Option, and Edge-to-Edge Option.

2. Conventions used in this document

The keywords "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

BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are used in this specification:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC7950].

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Design of the IOAM YANG Data Model

3.1. Profiles

The IOAM model is organized as list of profiles as shown in the following figure. Each profile associates with one flow and the corresponding IOAM information.

```
module: ietf-ioam
   +--rw ioam
      +--rw ioam-profiles
         +--rw admin-config
         |  +--rw enabled?   boolean
         +--rw ioam-profile* [profile-name]
            +--rw profile-name                    string
            +--rw filter
            |  +--rw filter-type?   ioam-filter-type
            |  +--rw acl-name?      -> /acl:acls/acl/name
            +--rw protocol-type?                  ioam-protocol-type
            +--rw incremental-tracing-profile {incremental-trace}?
            |  ...
            +--rw preallocated-tracing-profile {preallocated-trace}?
            |  ...
            +--rw direct-export-profile {direct-export}?
            |  ...
            +--rw pot-profile {proof-of-transit}?
            |  ...
            +--rw e2e-profile {edge-to-edge}?
               ...
```

The "enabled" is an administrative configuration. When it is set to true, IOAM configuration is enabled for the system. Meanwhile, the IOAM data-plane functionality is enabled.

The "filter" is used to identify a flow, where the IOAM profile can apply. There may be multiple filter types. ACL [RFC8519] is the default one.

The IOAM data can be encapsulated into multiple protocols, e.g., IPv6 [I-D.ietf-ippm-ioam-ipv6-options] and NSH [I-D.ietf-sfc-ioam-nsh]. The "protocol-type" is used to indicate where the IOAM is applied. For example, if the "protocol-type" is IPv6, the IOAM ingress node will encapsulate the associated flow with the IPv6-IOAM [I-D.ietf-ippm-ioam-ipv6-options] format.

IOAM data includes five encapsulation types, i.e., incremental tracing data, preallocated tracing data, direct export data, prove of transit data and end to end data. In practice, multiple IOAM data types can be encapsulated into the same IOAM header. The "ioam-profile" contains a set of sub-profiles, each of which relates to one encapsulation type. The configured object may not support all the sub-profiles. The supported sub-profiles are indicated by 5 defined features, i.e., "incremental-trace", "preallocated-trace", "direct export", "proof-of-transit", "edge-to-edge".

3.2. Preallocated Tracing Profile

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information . The "preallocated-tracing-profile" contains the detailed information for the preallocated tracing data. The information includes:

- enabled: indicates whether the preallocated tracing profile is enabled.
- node-action: indicates the operation (e.g., encapsulate IOAM header, transit the IOAM data, or decapsulate IOAM header) applied to the dedicated flow.
- use-namespace: indicate the namespace used for the trace types.
- trace-type: indicates the per-hop data to be captured by the IOAM enabled nodes and included in the node data list.
- Loopback mode is used to send a copy of a packet back towards the source.
- Active mode indicates that a packet is used for active measurement.

```
+--rw preallocated-tracing-profile {preallocated-trace}?
   +--rw enabled?                boolean
   +--rw node-action?            ioam-node-action
   +--rw trace-types
   |  +--rw use-namespace?   ioam-namespace
   |  +--rw trace-type*      ioam-trace-type
   +--rw enable-loopback-mode?   boolean
   +--rw enable-active-mode?   boolean
```

3.3. Incremental Tracing Profile

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header. The "incremental-tracing-profile" contains the detailed information for the incremental tracing data. The detailed information is the same as the Preallocated Tracing Profile, but with one more variable, "max-length", which restricts the length of the IOAM header.

```
   +--rw incremental-tracing-profile {incremental-trace}?
      +--rw enabled?                boolean
      +--rw node-action?            ioam-node-action
      +--rw trace-types
      |  +--rw use-namespace?   ioam-namespace
      |  +--rw trace-type*   ioam-trace-type
      +--rw enable-loopback-mode?   boolean
      +--rw enable-active-mode?   boolean
      +--rw max-length?             uint32
```

3.4. Direct Export Profile

The direct export option is used as a trigger for IOAM nodes to export IOAM data to a receiving entity (or entities). The "direct-export-profile" contains the detailed information for the direct export data. The detailed information is the same as the Preallocated Tracing Profile, but with one more optional variable, "flow-id", which is used to correlate the exported data of the same flow from multiple nodes and from multiple packets.

```
   +--rw direct-export-profile {direct-export}?
      +--rw enabled?                boolean
      +--rw node-action?            ioam-node-action
      +--rw trace-types
      |  +--rw use-namespace?   ioam-namespace
      |  +--rw trace-type*      ioam-trace-type
      +--rw enable-loopback-mode?   boolean
      +--rw enable-active-mode?   boolean
      +--rw flow-id?             uint32
```

3.5. Proof of Transit Profile

The IOAM Proof of Transit data is to support the path or service function chain verification use cases. The "pot-profile" contains the detailed information for the prove of transit data. The detailed information are described in [I-D.ietf-sfc-proof-of-transit].

```
   +--rw pot-profile {proof-of-transit}?
      +--rw enabled?                boolean
      +--rw active-profile-index?   pot:profile-index-range
      +--rw pot-profile-list* [pot-profile-index]
         +--rw pot-profile-index    profile-index-range
         +--rw prime-number         uint64
         +--rw secret-share         uint64
         +--rw public-polynomial    uint64
         +--rw lpc                  uint64
         +--rw validator?           boolean
         +--rw validator-key?       uint64
         +--rw bitmask?             uint64
             +--rw opot-masks
            +--rw downstream-mask*   uint64
            +--rw upstream-mask*     uint64
```

3.6. Edge to Edge Profile

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node. The "e2e-profile" contains the detailed information for the edge to edge data. The detailed information includes:

- enabled: indicates whether the edge to edge profile is enabled.
- node-action is the same semantic as in Section 2.2.
- use-namespace: indicate the namespace used for the edge to edge types.
- e2e-type indicates data to be carried from the ingress IOAM node to the egress IOAM node.

```
   +--rw e2e-profile {edge-to-edge}?
      +--rw enabled?       boolean
      +--rw node-action?   ioam-node-action
      +--rw e2e-types
         +--rw use-namespace?   ioam-namespace
         +--rw e2e-type*        ioam-e2e-type
```

4. IOAM YANG Module

```
 <CODE BEGINS> file "ietf-ioam@2020-07-13.yang"
 module ietf-ioam {
   yang-version 1.1;
   namespace "urn:ietf:params:xml:ns:yang:ietf-ioam";
   prefix "ioam";
```

```
   import ietf-pot-profile {
     prefix "pot";
     reference "draft-ietf-sfc-proof-of-transit";
   }

   import ietf-access-control-list {
     prefix "acl";
     reference
       "RFC 8519: YANG Data Model for Network Access Control
        Lists (ACLs)";
   }

   organization
     "IETF IPPM (IP Performance Metrics) Working Group";

   contact
     "WG Web: <http://tools.ietf.org/wg/ippm>
      WG List: <ippm@ietf.org>
      Editor: zhoutianran@huawei.com
      Editor: james.n.guichard@futurewei.com
      Editor: fbrockne@cisco.com
      Editor: srihari@cisco.com";

   description
     "This YANG module specifies a vendor-independent data
      model for the In Situ OAM (IOAM).

     Copyright (c) 2020 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

   revision 2020-07-13 {
     description "Initial revision.";
     reference "draft-zhou-ippm-ioam-yang";
   }

  /*
   * FEATURES
   */
```

```
   feature incremental-trace
   {
     description
       "This feature indicated that the incremental tracing option is
        supported";
     reference "draft-ietf-ippm-ioam-data";
   }

   feature preallocated-trace
   {
     description
       "This feature indicated that the preallocated tracing option is
        supported";
     reference "draft-ietf-ippm-ioam-data";
   }

   feature direct-export
   {
     description
       "This feature indicated that the direct export option is
        supported";
     reference "ietf-ippm-ioam-direct-export";
   }

   feature proof-of-transit
   {
     description
       "This feature indicated that the proof of transit option is
        supported";
     reference "draft-ietf-ippm-ioam-data";
   }

   feature edge-to-edge
   {
     description
       "This feature indicated that the edge to edge option is
        supported";
     reference "draft-ietf-ippm-ioam-data";
   }

  /*
   * IDENTITIES
   */
   identity base-filter {
     description
       "Base identity to represent a filter. A filter is used to
       specify the flow to apply the IOAM profile. ";
   }
```

```
   identity acl-filter {
     base base-filter;
     description
       "Apply ACL rules to specify the flow.";
   }

   identity base-protocol {
     description
       "Base identity to represent the carrier protocol. It's used to
        indicate what layer and protocol the IOAM data is embedded.";
   }

   identity ipv6-protocol {
     base base-protocol;
     description
       "The described IOAM data is embedded in IPv6 protocol.";
     reference "ietf-ippm-ioam-ipv6-options";
   }

   identity nsh-protocol  {
     base base-protocol;
     description
       "The described IOAM data is embedded in NSH.";
     reference "ietf-sfc-ioam-nsh";
   }

   identity base-node-action {
     description
       "Base identity to represent the node actions. It's used to
        indicate what action the node will take.";
   }

   identity action-encapsulate {
     base base-node-action;
     description
       "indicate the node is to encapsulate the IOAM packet";
   }

   identity action-transit {
     base base-node-action;
     description
       "indicate the node is to transit the IOAM packet";
   }

   identity action-decapsulate {
     base base-node-action;
     description
       "indicate the node is to decapsulate the IOAM packet";
```

```
   }

   identity base-trace-type {
     description
       "Base identity to represent trace types";
   }

   identity trace-hop-lim-node-id {
     base base-trace-type;
     description
       "indicates presence of Hop_Lim and node_id in the
        node data.";
   }

   identity trace-if-id {
     base base-trace-type;
     description
       "indicates presence of ingress_if_id and egress_if_id in the
        node data.";
   }

   identity trace-timestamp-seconds {
     base base-trace-type;
     description
       "indicates presence of time stamp seconds in the node data.";
   }

   identity trace-timestamp-nanoseconds {
     base base-trace-type;
     description
       "indicates presence of time stamp nanoseconds in the node data.";
   }

   identity trace-transit-delay {
     base base-trace-type;
     description
       "indicates presence of transit delay in the node data.";
   }

   identity trace-namespace-data {
     base base-trace-type;
     description
       "indicates presence of namespace specific data (short format)
        in the node data.";
   }

   identity trace-queue-depth {
     base base-trace-type;
```

```
    description
      "indicates presence of queue depth in the node data.";
  }

  identity trace-opaque-state-snapshot {
    base base-trace-type;
    description
      "indicates presence of variable length Opaque State Snapshot
       field.";
  }

  identity trace-hop-lim-node-id-wide {
    base base-trace-type;
    description
      "indicates presence of Hop_Lim and node_id wide in the
       node data.";
  }

  identity trace-if-id-wide {
    base base-trace-type;
    description
      "indicates presence of ingress_if_id and egress_if_id wide in
       the node data.";
  }

  identity trace-namespace-data-wide {
    base base-trace-type;
    description
      "indicates presence of namespace specific data in wide format
       in the node data.";
  }

  identity trace-buffer-occupancy {
    base base-trace-type;
    description
      "indicates presence of buffer occupancy in the node data.";
  }

  identity trace-checksum-complement {
    base base-trace-type;
    description
      "indicates presence of the Checksum Complement node data.";
  }

  identity base-pot-type {
    description
      "Base identity to represent Proof of Transit(PoT) types";
  }
```

```
   identity pot-bytes-16 {
     base base-pot-type;
     description
       "POT data is a 16 Octet field.";
   }

   identity base-e2e-type {
     description
       "Base identity to represent e2e types";
   }

   identity e2e-seq-num-64 {
     base base-e2e-type;
     description
       "indicates presence of a 64-bit sequence number";
   }

   identity e2e-seq-num-32 {
     base base-e2e-type;
     description
       "indicates presence of a 32-bit sequence number";
   }

   identity e2e-timestamp-seconds {
     base base-e2e-type;
     description
       "indicates presence of timestamp seconds for the
        transmission of the frame";
   }

   identity e2e-timestamp-subseconds {
     base base-e2e-type;
     description
       "indicates presence of timestamp subseconds for the
        transmission of the frame";
   }

   identity base-namespace {
     description
       "Base identity to represent the namespace";
   }

   identity namespace-ietf {
     base base-namespace;
     description
       "namespace that specified in IETF.";
   }
```

```
  /*
   * TYPE DEFINITIONS
   */

   typedef ioam-filter-type {
     type identityref {
       base base-filter;
     }
     description
       "Specifies a known type of filter.";
   }

   typedef ioam-protocol-type {
     type identityref {
       base base-protocol;
     }
     description
       "Specifies a known type of carrier protocol for the IOAM data.";
   }

   typedef ioam-node-action {
     type identityref {
       base base-node-action;
     }
     description
       "Specifies a known type of node action.";
   }

   typedef ioam-trace-type {
     type identityref {
       base base-trace-type;
     }
     description
       "Specifies a known trace type.";
   }

   typedef ioam-pot-type {
     type identityref {
       base base-pot-type;
     }
     description
       "Specifies a known pot type.";
   }

   typedef ioam-e2e-type {
     type identityref {
       base base-e2e-type;
     }
```

```
    description
      "Specifies a known e2e type.";
  }

  typedef ioam-namespace {
    type identityref {
      base base-namespace;
    }
    description
      "Specifies the supported namespace.";
  }

 /*
  * GROUP DEFINITIONS
  */

  grouping ioam-filter {
    description "A grouping for IOAM filter definition";

    leaf filter-type {
      type ioam-filter-type;
      description "filter type";
    }

    leaf acl-name {
      when "../filter-type = 'ioam:acl-filter'";
      type leafref {
        path "/acl:acls/acl:acl/acl:name";
      }
      description "Access Control List name.";
    }
  }

  grouping encap-tracing {
    description
      "A grouping for the generic configuration for
       tracing profile.";

    container trace-types {
      description
        "the list of trace types for encapsulate";

      leaf use-namespace {
        type ioam-namespace;
        description
          "the namespace used for the encapsulation";
      }
```

```
      leaf-list trace-type {
        type ioam-trace-type;
        description
          "The trace type is only defined at the encapsulation node.";
      }
    }

    leaf enable-loopback-mode {
      type boolean;
      default false;
      description
        "Loopback mode is used to send a copy of a packet back towards
        the source. The loopback mode is only defined at the
        encapsulation node.";
    }

    leaf enable-active-mode {
      type boolean;
      default false;
      description
        "Active mode indicates that a packet is used for active
         measurement. An IOAM decapsulating node that receives a
         packet with the Active flag set in one of its Trace options
         must terminate the packet.";
    }
  }

  grouping ioam-incremental-tracing-profile {
    description
      "A grouping for incremental tracing profile.";

    leaf node-action {
      type ioam-node-action;
      description "node action";
    }

    uses encap-tracing {
      when "node-action = 'ioam:action-encapsulate'";
    }

    leaf max-length {
      when "../node-action = 'ioam:action-encapsulate'";
      type uint32;
      description
        "This field specifies the maximum length of the node data list
        in octets. The max-length is only defined at the
        encapsulation node. And it's only used for the incremental
        tracing mode.";
```

```
    }
  }

  grouping ioam-preallocated-tracing-profile {
    description
      "A grouping for incremental tracing profile.";


    leaf node-action {
      type ioam-node-action;
      description "node action";
    }

    uses encap-tracing {
      when "node-action = 'ioam:action-encapsulate'";
    }
  }

  grouping ioam-direct-export-profile {
    description
      "A grouping for direct export profile.";

    leaf node-action {
      type ioam-node-action;
      description "node action";
    }

    uses encap-tracing {
      when "node-action = 'ioam:action-encapsulate'";
    }

    leaf flow-id {
      when "../node-action = 'ioam:action-encapsulate'";
      type uint32;
      description
        "flow-id is used to correlate the exported data of the same
         flow from multiple nodes and from multiple packets.";
    }
  }

  grouping ioam-e2e-profile {
    description
      "A grouping for end to end profile.";

    leaf node-action {
      type ioam-node-action;
      description
        "indicate how the node act for this profile";
```

```
    }

    container e2e-types {
      when "../node-action = 'ioam:action-encapsulate'";
      description
        "the list of e2e types for encapsulate";

      leaf use-namespace {
        type ioam-namespace;
        description
          "the namespace used for the encapsulation";
      }

      leaf-list e2e-type {
        type ioam-e2e-type;
        description
          "The e2e type is only defined at the encapsulation node.";
      }
    }
  }

  grouping ioam-admin-config {
    description
      "IOAM top-level administrative configuration.";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, IOAM configuration is enabled for the system.
         Meanwhile, the IOAM data-plane functionality is enabled.";
    }
  }

 /*
  * DATA NODES
  */

  container ioam {
    description "IOAM top level container";

    container ioam-profiles {
      description
        "Contains a list of IOAM profiles.";

      container admin-config {
        description
          "Contains all the administrative configurations related to
```

```
           the IOAM functionalities and all the IOAM profiles.";

        uses ioam-admin-config;
      }

      list ioam-profile {
        key "profile-name";
        ordered-by user;
        description
          "A list of IOAM profiles that configured on the node.";

        leaf profile-name {
          type string;
          mandatory true;
          description
            "Unique identifier for each IOAM profile";
        }

        container filter {
          uses ioam-filter;
          description
            "The filter which is used to indicate the flow to apply
            IOAM.";
        }

        leaf protocol-type {
          type ioam-protocol-type;
          description
            "This item is used to indicate the carrier protocol where
            the IOAM is applied.";
        }

        container incremental-tracing-profile {
          if-feature incremental-trace;
          description
            "describe the profile for incremental tracing option";

          leaf enabled {
            type boolean;
            default false;
            description
              "When true, apply incremental tracing option to the
               specified flow identified by the filter.";
          }

          uses ioam-incremental-tracing-profile;
        }
```

```
        container preallocated-tracing-profile {
          if-feature preallocated-trace;
          description
            "describe the profile for preallocated tracing option";

          leaf enabled {
            type boolean;
            default false;
            description
              "When true, apply preallocated tracing option to the
               specified flow identified by the following filter.";
          }

          uses ioam-preallocated-tracing-profile;
        }

        container direct-export-profile {
          if-feature direct-export;
          description
            "describe the profile for direct-export option";

          leaf enabled {
            type boolean;
            default false;
            description
              "When true, apply direct-export option to the
               specified flow identified by the following filter.";
          }

          uses ioam-direct-export-profile;
        }

        container pot-profile {
          if-feature proof-of-transit;
          description
            "describe the profile for PoT option";

          leaf enabled {
            type boolean;
            default false;
            description
              "When true, apply Proof of Transit option to the
               specified flow identified by the following filter.";
          }

          leaf active-profile-index {
            type pot:profile-index-range;
            description
```

```
                "Proof of transit profile index that is currently
                 active. Will be set in the first hop of the path
                 or chain. Other nodes will not use this field.";
            }

            uses pot:pot-profile;
          }

          container e2e-profile {
            if-feature edge-to-edge;
            description
              "describe the profile for e2e option";

            leaf enabled {
              type boolean;
              default false;
              description
                "When true, apply End to end option to the
                 specified flow identified by the following filter.";
            }

            uses ioam-e2e-profile;
          }
        }
      }
    }
  }
 <CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config)

to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

o /ioam/ioam-profiles/admin-config

The items in the container above include the top level administrative configurations related to the IOAM functionalities and all the IOAM profiles. Unexpected changes to these items could lead to the IOAM function disruption and/ or misbehavior of all the IOAM profiles.

o /ioam/ioam-profiles/ioam-profile

The entries in the list above include the whole IOAM profile configurations which indirectly create or modify the device configurations. Unexpected changes to these entries could lead to the mistake of the IOAM behavior for the corresponding flows.

6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

IANA is requested to assign a new URI from the IETF XML Registry [RFC3688]. The following URI is suggested:

```
URI: urn:ietf:params:xml:ns:yang:ietf-ioam
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document also requests a new YANG module name in the YANG Module Names registry [RFC7950] with the following suggestion:

```
name: ietf-ioam
namespace: urn:ietf:params:xml:ns:yang:ietf-ioam
prefix: ioam
reference: RFC XXXX
```

7. Acknowledgements

For their valuable comments, discussions, and feedback, we wish to acknowledge Greg Mirsky, Reshad Rahman and Tom Petch.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-00 (work in progress), February 2020.

[I-D.ietf-sfc-proof-of-transit]
Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S. Youell, "Proof of Transit", draft-ietf-sfc-proof-of-transit-06 (work in progress), June 2020.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <https://www.rfc-editor.org/info/rfc8519>.

8.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., and R. Asati, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-02 (work in progress), July 2020.

[I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-04 (work in progress), June 2020.

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Jim Guichard
Futurewei
United States of America

Email: james.n.guichard@futurewei.com

Frank Brockners
Cisco Systems
Hansaallee 249, 3rd Floor
Duesseldorf, Nordrhein-Westfalen 40549
Germany

Email: fbrockne@cisco.com

Srihari Raghavan
Cisco Systems
Tril Infopark Sez, Ramanujan IT City
Neville Block, 2nd floor, Old Mahabalipuram Road
Chennai, Tamil Nadu 600113
India

Email: srihari@cisco.com