

IPPM Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 2, 2020

M. Cociglio  
Telecom Italia  
G. Fioccola  
Huawei Technologies  
F. Bulgarella  
R. Sisto  
Politecnico di Torino  
July 1, 2019

New Spin bit enabled measurements with one or two more bits  
draft-cfb-ippm-spinbit-new-measurements-01

## Abstract

This document introduces additional measurements by using the same spin bit signal as defined in [I-D.trammell-ippm-spin]. The spin bit signal alone is not enough to evaluate correctly in every network condition the RTT of a flow. In order to solve this problem, it is theorized the possibility of introducing an additional validation signal called delay bit, similar to what is done by the Valid Edge Counter (VEC), but using just one bit instead of two. An alternative with two bits is also introduced with a so called loss bit.

## 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 2, 2020.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Spin bit and Delay bit mechanism . . . . .	3
2.1. Delay Sample generation . . . . .	5
2.1.1. The recovery process . . . . .	5
2.2. Delay Sample reflection . . . . .	6
3. Using the Spin bit and Delay bit for Hybrid RTT Measurement . . . . .	7
3.1. End-to-end RTT measurement . . . . .	7
3.2. Half-RTT measurement . . . . .	7
3.3. Intra-domain RTT measurement . . . . .	7
4. Observer's algorithm and Waiting Interval . . . . .	8
5. Adding a Loss bit to Delay bit and Spin bit . . . . .	9
6. Round Trip Packet Loss measurement . . . . .	9
6.1. RTT dependent Packet Loss using one bit . . . . .	10
6.2. RTT independent Packet Loss using two bits . . . . .	10
7. Protocols . . . . .	11
7.1. QUIC . . . . .	11
7.2. TCP . . . . .	11
8. Security Considerations . . . . .	11
9. Acknowledgements . . . . .	11
10. IANA Considerations . . . . .	11
11. References . . . . .	11
11.1. Normative References . . . . .	11
11.2. Informative References . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

[I-D.trammell-ippm-spin] defines 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.

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

The document introduces a mechanism applicable to any transport-layer protocol, then explains how to bind the signal 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 only (without the VEC) to QUIC for experimentation purposes.

Note that both the spin bit and the delay bit are inspired by RFC 8321 [RFC8321]. This is also mentioned in [I-D.trammell-quic-spin].

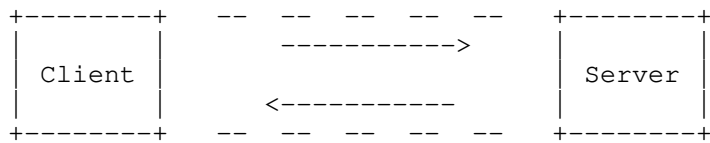
## 2. Spin bit and Delay bit mechanism

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.

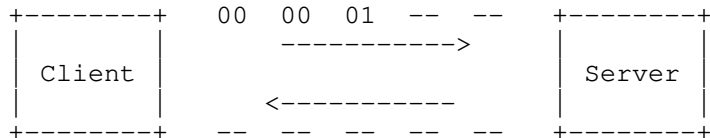
A simple observer placed in an intermediate point, tracking the delay sample and the relative timestamp in every spin bit period, can measure the end-to-end round trip delay of the connection. In the same way as seen with the spin bit and the VEC, it is possible to carry out other types of measurements.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

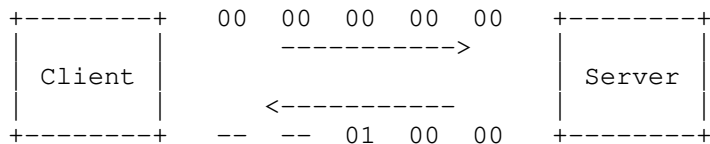
The next figure describes the Delay bit mechanism: the first bit is the spin bit and the second one is the delay bit.



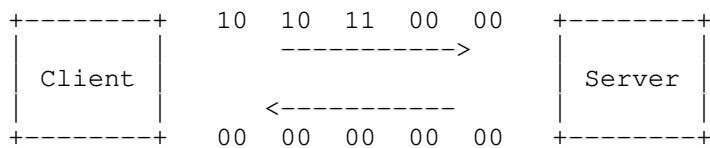
(a) No traffic at beginning.



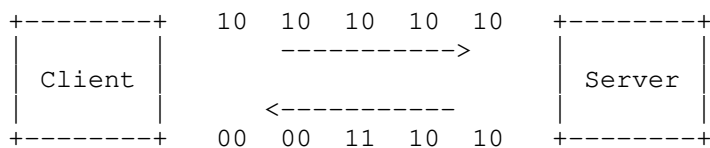
(b) The Client starts sending data and sets the first packet as Delay Sample.



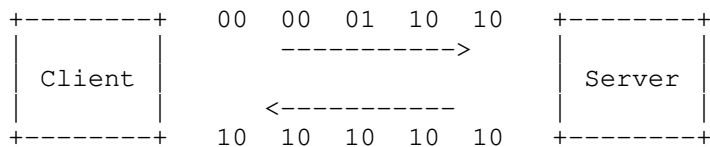
(c) The Server starts sending data and reflects the Delay Sample.



(d) The Client inverts the spin bit and reflects the Delay Sample.



(e) The Server reflects the Delay Sample.



- (f) The client reverts the spin bit and reflects the Delay Sample.

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:

- o 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;
- o 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.

It is noteworthy that differently from what happens with the VEC for which the reflection always concerns the edge of the period, in this case 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 (with or without the VEC).

#### 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.

#### 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.

#### 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.

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:

- o 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 an 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 bit to Delay bit and Spin bit

It is possible to introduce a mechanism to evaluate also the packet loss together with the delay measurement. In particular, the Client can select and mark a train of packets for this purpose, by using a loss bit, additionally to the spin bit and delay bit.

These packets bounce between Client and Server to complete two rounds and an Observer counts the marked packets during the two rounds and compares the counters to find Round Trip (RT) losses.

The problem to be solved is to choose the right number of packets to mark to avoid marked packets congestion on the slowest traffic direction. But the solution is simple, because it is enough to choose the number of packets that transit on the slowest direction during an RTT.

#### 6. Round Trip Packet Loss measurement

The Client generates a train of marked packets (Packet Loss Samples) by using the additional bit called Loss bit. The marked packets are generated at the slowest direction rate (only when a packet arrives the Client marks an outgoing packet). The Server reflects these packets accordingly and, as a consequence, it could insert some not-marked packets. Then the client reflects the marked packets and the server reflects the marked packets again. The Client generates a new train of marked packets and so on.

The Packet Loss calculation can be made after the comparison of counters taken by the on-path passive observer. Indeed the Observer in the middle (upstream or downstream) sees the packet train twice

and so it calculates the Observer Round Trip Packet Loss that, statistically, will be equal to the end-to-end Round Trip Packet Loss. So this measurement can be simply referred as Round Trip Packet Loss (RTPL).

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

The method allows the packet loss calculation for a portion of the traffic but it is useful to perform RT Packet Loss measurement that gives useful information coupled with RTT.

#### 6.1. RTT dependent Packet Loss using one bit

Using a single bit in addition to the spin bit and delay bit enables passive measurability of the end-to-end round-trip loss rate.

The algorithm requires a mechanism to individually identify each train of packets in order to enable the observer to distinguish between trains belonging to different rounds. This is achieved by introducing a temporal pause of  $2 \times \text{RTT}$  duration during which no marked packets are forwarded. Marked packets are generated by the client for the duration of an RTT in order to be synchronized with the spin bit algorithm and to have a sufficient numbers of marked packets.

However, this single bit methodology replies and exposes the RTT of the connection in any case, when the spin bit and the delay bit are used and when these are disabled.

#### 6.2. RTT independent Packet Loss using two bits

An RTT independent version of this algorithm requires two bits and can be used when both spin bit and delay bit are disabled. This implies that an observer must be able to determine whether the spin bit is active and correctly spinning or not (choosing, accordingly, the right version of packet loss measurement to be used).

Without using the spin bit, it is difficult to find the right pause duration but, with a two bits packet loss field, the temporal pause necessary to distinguish the different train of packets is no longer needed. That's because packets generated and reflected by the client are marked using two different marking values. Furthermore, instead of generating marked packets for the duration of an RTT, a fixed duration for the generation phase can be used (e.g. 100ms).

In this way, no information related to the RTT of the connection is transmitted on the wire.

## 7. Protocols

### 7.1. QUIC

The binding of this signal to QUIC is partially described in [I-D.ietf-quic-spin-exp], which adds the spin bit only to QUIC.

From an implementation point of view, the delay bit is placed in the partially unencrypted (but authenticated) QUIC header, alongside the spin bit, occupying one of the two bits left reserved for future experiments. As things stand, according to [I-D.ietf-quic-transport], the proposed scheme of the first header's byte would be 01SDRKPP.

### 7.2. TCP

The signal 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 loss bit.

## 8. Security Considerations

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

## 9. Acknowledgements

tbc

## 10. IANA Considerations

tbc

## 11. References

### 11.1. 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-20 (work in progress), April 2019.

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

## 11.2. Informative References

- [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.

## Authors' Addresses

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

Email: [mauro.cociglio@telecomitalia.it](mailto:mauro.cociglio@telecomitalia.it)

Giuseppe Fioccola  
Huawei Technologies  
Riesstrasse, 25  
Munich 80992  
Germany

Email: [giuseppe.fioccola@huawei.com](mailto:giuseppe.fioccola@huawei.com)

Fabio Bulgarella  
Politecnico di Torino

Email: [fabio.bulgarella@guest.telecomitalia.it](mailto:fabio.bulgarella@guest.telecomitalia.it)

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

Email: [riccardo.sisto@polito.it](mailto:riccardo.sisto@polito.it)

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.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions Used in This Document . . . . .	3
2.1. Requirements Language . . . . .	3
2.2. Abbreviations . . . . .	3
2.3. Reference Topology . . . . .	4
3. Overview . . . . .	5
3.1. Example Provisioning Model . . . . .	6
4. Probe Messages . . . . .	7
4.1. Probe Query Message . . . . .	7
4.1.1. Delay Measurement Query Message . . . . .	7
4.1.2. Loss Measurement Query Message . . . . .	8
4.1.3. Probe Query for Links . . . . .	9
4.1.4. Probe Query for SR Policy . . . . .	9
4.2. Probe Response Message . . . . .	11
4.2.1. One-way Measurement Mode . . . . .	11
4.2.2. Two-way Measurement Mode . . . . .	11
4.2.3. Loopback Measurement Mode . . . . .	13
4.3. Additional Probe Message Processing Rules . . . . .	14
4.3.1. TTL and Hop Limit . . . . .	14
4.3.2. Router Alert Option . . . . .	14
4.3.3. UDP Checksum . . . . .	14
5. Performance Measurement for P2MP SR Policies . . . . .	14
6. ECMP Support for SR Policies . . . . .	16
7. Performance Delay and Liveness Monitoring . . . . .	16
8. Security Considerations . . . . .	16
9. IANA Considerations . . . . .	17
10. References . . . . .	17
10.1. Normative References . . . . .	17
10.2. Informative References . . . . .	17
Acknowledgments . . . . .	20
Authors' Addresses . . . . .	21

## 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.

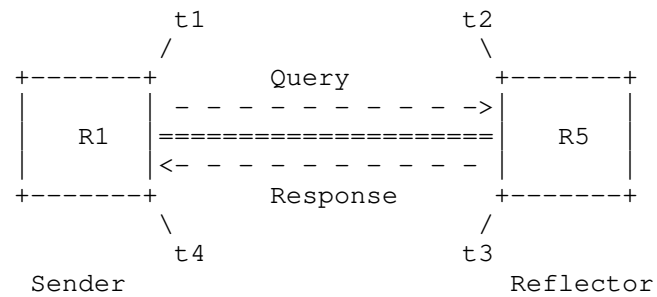
TC: Traffic Class.

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:

- o 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.
- o 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:

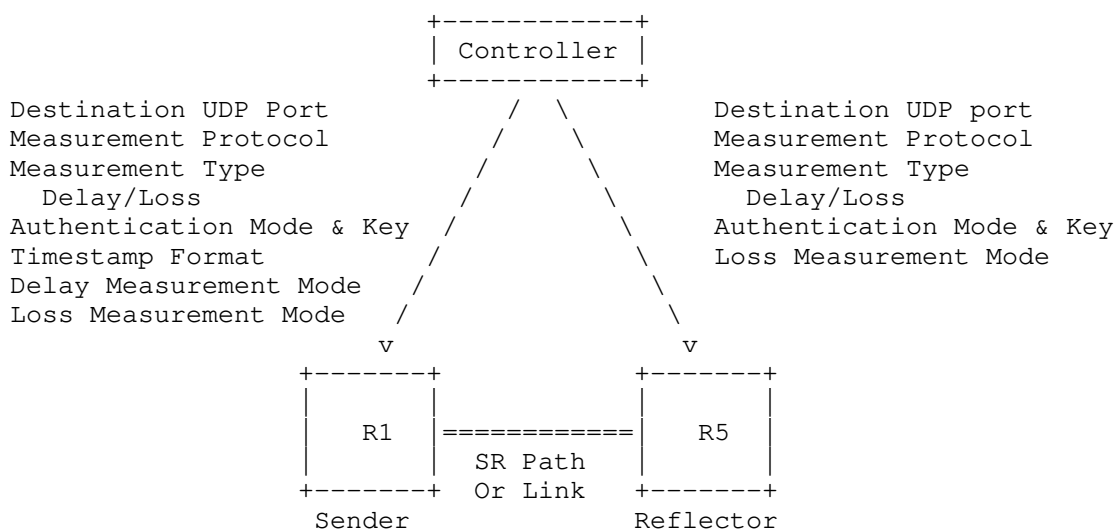


Figure 1: Example Provisioning Model

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].

```

+-----+
| IP Header                                     |
. Source IP Address = Sender IPv4 or IPv6 Address .
. Destination IP Address = Reflector IPv4 or IPv6 Address .
. Protocol = UDP .
. .
+-----+
| UDP Header                                   |
. Source Port = As chosen by Sender .
. Destination Port = User-configured Port for Delay Measurement.
. .
+-----+
| Payload = DM Message as specified in Section 4.2.1 of RFC 5357 |
. Payload = DM Message as specified in Section 4.1.2 of RFC 5357.
. .
+-----+

```

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].

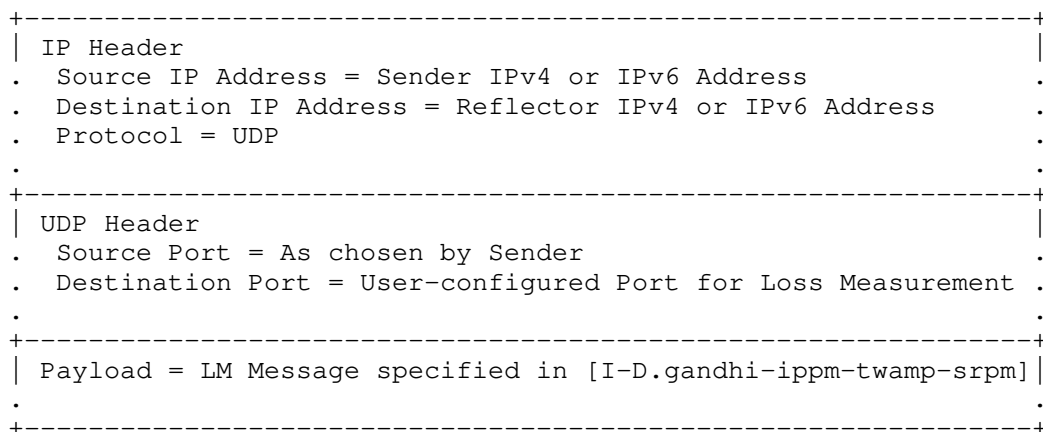


Figure 3: LM Probe Query Message

#### 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.

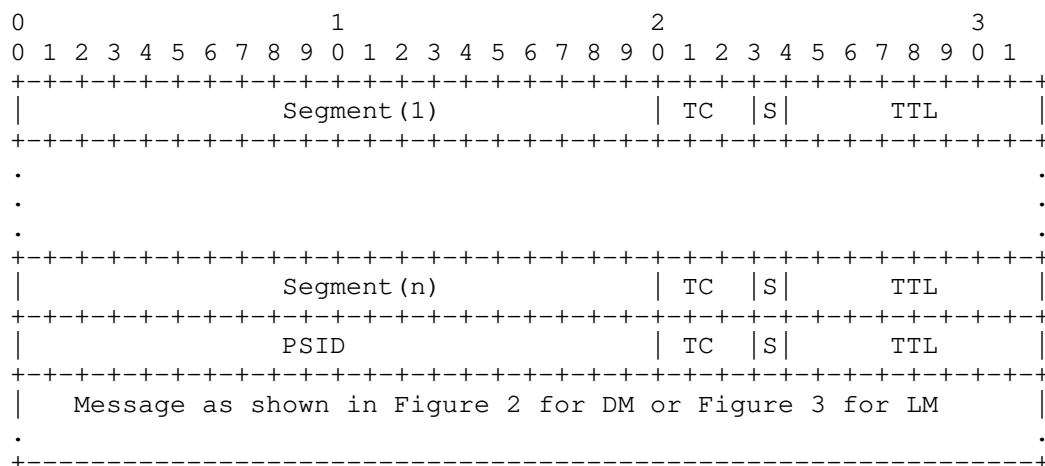


Figure 4: Example Probe Query Message for SR-MPLS Policy

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.

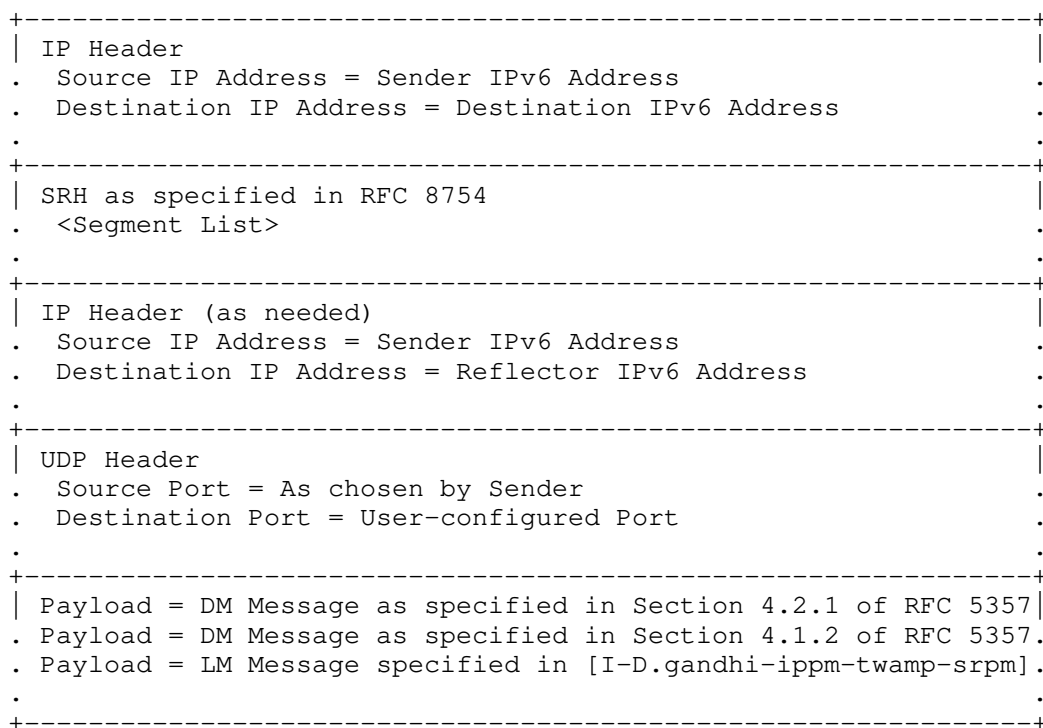


Figure 5: Example Probe Query Message for SRv6 Policy

## 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.

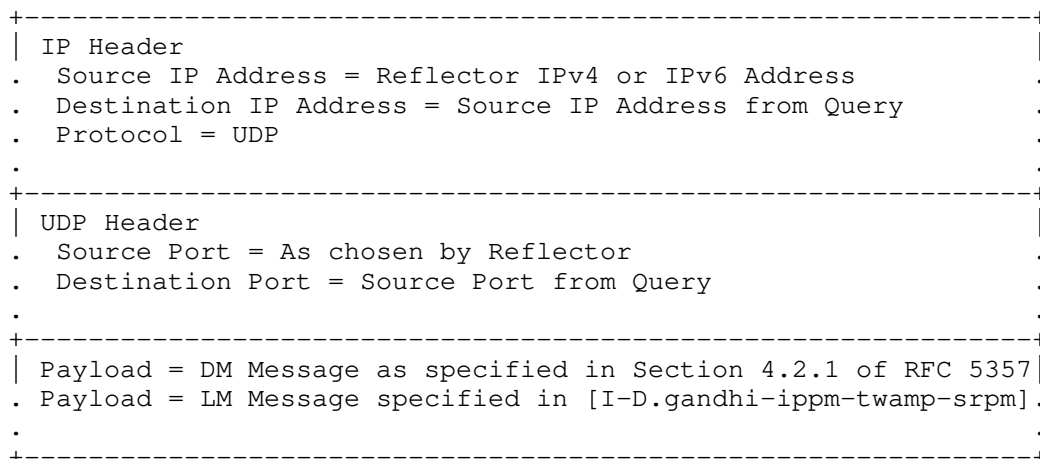


Figure 6: Probe Response Message

### 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  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  are collected by the probes. However, only timestamps  $t_1$  and  $t_2$  are used to measure one-way delay as  $(t_2 - t_1)$ .

### 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  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  are collected by the probes. All four timestamps are used to measure two-way delay as  $((t_4 - t_1) - (t_3 - t_2))$ .





```

+-----+
| IP Header                                     |
. Source IP Address = Reflector IPv6 Address   .
. Destination IP Address = Destination IPv6 Address .
.                                             .
+-----+
| SRH as specified in RFC 8754                 |
. <Segment List>                             .
.                                             .
+-----+
| IP Header (as needed)                       |
. Source IP Address = Reflector IPv6 Address   .
. Destination IP Address = Source IPv6 Address from Query .
.                                             .
+-----+
| UDP Header                                   |
. Source Port = As chosen by Sender           .
. Destination Port = User-configured Port     .
.                                             .
+-----+
| Payload = DM Message as specified in Section 4.2.1 of RFC 5357 |
. Payload = LM Message specified in [I-D.gandhi-ippm-twamp-srpm].
.                                             .
+-----+

```

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:

- o 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.
- o The probe query messages can contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].
- o The Destination Address is set to the loopback address from range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 address.
- o 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.
- o The P2MP root node measures the delay and loss performance for each P2MP leaf node of the end-to-end P2MP SR Policy.

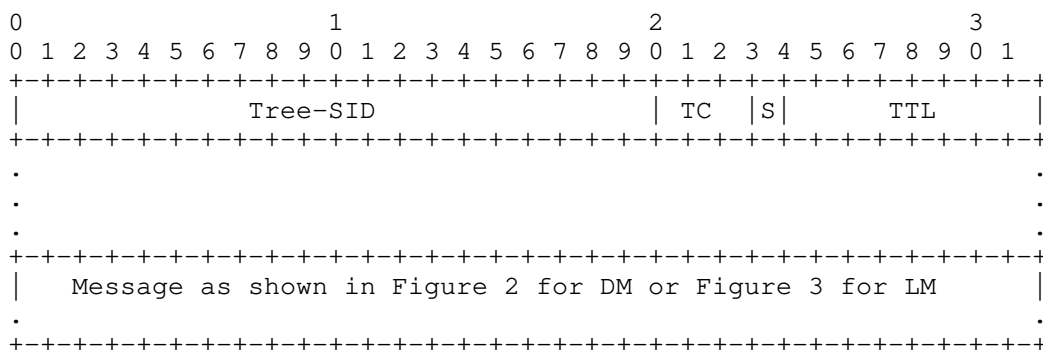


Figure 9: Example Probe Query with Tree-SID for SR-MPLS Policy

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: September 10, 2020

A. Morton  
AT&T Labs  
M. Bagnulo  
UC3M  
P. Eardley  
BT  
K. D'Souza  
AT&T Labs  
March 9, 2020

Initial Performance Metrics Registry Entries  
draft-ietf-ippm-initial-registry-16

Abstract

This memo defines the set of Initial Entries for the IANA Performance Metrics Registry. The set includes: UDP Round-trip Latency and Loss, Packet Delay Variation, DNS Response Latency and Loss, UDP Poisson One-way Delay and Loss, UDP Periodic One-way Delay and Loss, ICMP Round-trip Latency and Loss, and TCP round-trip Latency and Loss.

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.

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 September 10, 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	6
2. Scope . . . . .	7
3. Registry Categories and Columns . . . . .	7
4. UDP Round-trip Latency and Loss Registry Entries . . . . .	8
4.1. Summary . . . . .	9
4.1.1. ID (Identifier) . . . . .	9
4.1.2. Name . . . . .	9
4.1.3. URI . . . . .	9
4.1.4. Description . . . . .	9
4.1.5. Change Controller . . . . .	9
4.1.6. Version (of Registry Format) . . . . .	9
4.2. Metric Definition . . . . .	10
4.2.1. Reference Definition . . . . .	10
4.2.2. Fixed Parameters . . . . .	10
4.3. Method of Measurement . . . . .	11
4.3.1. Reference Method . . . . .	11
4.3.2. Packet Stream Generation . . . . .	12
4.3.3. Traffic Filtering (observation) Details . . . . .	13
4.3.4. Sampling Distribution . . . . .	13
4.3.5. Run-time Parameters and Data Format . . . . .	13
4.3.6. Roles . . . . .	14
4.4. Output . . . . .	14
4.4.1. Type . . . . .	14
4.4.2. Reference Definition . . . . .	14
4.4.3. Metric Units . . . . .	15
4.4.4. Calibration . . . . .	15
4.5. Administrative items . . . . .	16
4.5.1. Status . . . . .	16
4.5.2. Requester . . . . .	16
4.5.3. Revision . . . . .	16
4.5.4. Revision Date . . . . .	16

4.6.	Comments and Remarks . . . . .	16
5.	Packet Delay Variation Registry Entry . . . . .	16
5.1.	Summary . . . . .	16
5.1.1.	ID (Identifier) . . . . .	16
5.1.2.	Name . . . . .	16
5.1.3.	URI . . . . .	17
5.1.4.	Description . . . . .	17
5.1.5.	Change Controller . . . . .	17
5.1.6.	Version (of Registry Format) . . . . .	17
5.2.	Metric Definition . . . . .	17
5.2.1.	Reference Definition . . . . .	17
5.2.2.	Fixed Parameters . . . . .	18
5.3.	Method of Measurement . . . . .	19
5.3.1.	Reference Method . . . . .	19
5.3.2.	Packet Stream Generation . . . . .	19
5.3.3.	Traffic Filtering (observation) Details . . . . .	20
5.3.4.	Sampling Distribution . . . . .	20
5.3.5.	Run-time Parameters and Data Format . . . . .	20
5.3.6.	Roles . . . . .	21
5.4.	Output . . . . .	21
5.4.1.	Type . . . . .	21
5.4.2.	Reference Definition . . . . .	21
5.4.3.	Metric Units . . . . .	22
5.4.4.	Calibration . . . . .	22
5.5.	Administrative items . . . . .	23
5.5.1.	Status . . . . .	23
5.5.2.	Requester . . . . .	23
5.5.3.	Revision . . . . .	23
5.5.4.	Revision Date . . . . .	23
5.6.	Comments and Remarks . . . . .	23
6.	DNS Response Latency and Loss Registry Entries . . . . .	23
6.1.	Summary . . . . .	23
6.1.1.	ID (Identifier) . . . . .	24
6.1.2.	Name . . . . .	24
6.1.3.	URI . . . . .	24
6.1.4.	Description . . . . .	24
6.1.5.	Change Controller . . . . .	24
6.1.6.	Version (of Registry Format) . . . . .	24
6.2.	Metric Definition . . . . .	24
6.2.1.	Reference Definition . . . . .	24
6.2.2.	Fixed Parameters . . . . .	25
6.3.	Method of Measurement . . . . .	27
6.3.1.	Reference Method . . . . .	27
6.3.2.	Packet Stream Generation . . . . .	28
6.3.3.	Traffic Filtering (observation) Details . . . . .	29
6.3.4.	Sampling Distribution . . . . .	29
6.3.5.	Run-time Parameters and Data Format . . . . .	29
6.3.6.	Roles . . . . .	30

6.4.	Output . . . . .	30
6.4.1.	Type . . . . .	30
6.4.2.	Reference Definition . . . . .	31
6.4.3.	Metric Units . . . . .	31
6.4.4.	Calibration . . . . .	31
6.5.	Administrative items . . . . .	32
6.5.1.	Status . . . . .	32
6.5.2.	Requester . . . . .	32
6.5.3.	Revision . . . . .	32
6.5.4.	Revision Date . . . . .	32
6.6.	Comments and Remarks . . . . .	32
7.	UDP Poisson One-way Delay and Loss Registry Entries . . . . .	32
7.1.	Summary . . . . .	32
7.1.1.	ID (Identifier) . . . . .	33
7.1.2.	Name . . . . .	33
7.1.3.	URI . . . . .	33
7.1.4.	Description . . . . .	33
7.2.	Metric Definition . . . . .	34
7.2.1.	Reference Definition . . . . .	34
7.2.2.	Fixed Parameters . . . . .	35
7.3.	Method of Measurement . . . . .	36
7.3.1.	Reference Method . . . . .	36
7.3.2.	Packet Stream Generation . . . . .	36
7.3.3.	Traffic Filtering (observation) Details . . . . .	37
7.3.4.	Sampling Distribution . . . . .	37
7.3.5.	Run-time Parameters and Data Format . . . . .	37
7.3.6.	Roles . . . . .	38
7.4.	Output . . . . .	38
7.4.1.	Type . . . . .	38
7.4.2.	Reference Definition . . . . .	38
7.4.3.	Metric Units . . . . .	41
7.4.4.	Calibration . . . . .	41
7.5.	Administrative items . . . . .	42
7.5.1.	Status . . . . .	42
7.5.2.	Requester . . . . .	42
7.5.3.	Revision . . . . .	42
7.5.4.	Revision Date . . . . .	43
7.6.	Comments and Remarks . . . . .	43
8.	UDP Periodic One-way Delay and Loss Registry Entries . . . . .	43
8.1.	Summary . . . . .	43
8.1.1.	ID (Identifier) . . . . .	43
8.1.2.	Name . . . . .	43
8.1.3.	URI . . . . .	44
8.1.4.	Description . . . . .	44
8.2.	Metric Definition . . . . .	44
8.2.1.	Reference Definition . . . . .	44
8.2.2.	Fixed Parameters . . . . .	45
8.3.	Method of Measurement . . . . .	46

8.3.1.	Reference Method . . . . .	46
8.3.2.	Packet Stream Generation . . . . .	47
8.3.3.	Traffic Filtering (observation) Details . . . . .	48
8.3.4.	Sampling Distribution . . . . .	48
8.3.5.	Run-time Parameters and Data Format . . . . .	48
8.3.6.	Roles . . . . .	48
8.4.	Output . . . . .	49
8.4.1.	Type . . . . .	49
8.4.2.	Reference Definition . . . . .	49
8.4.3.	Metric Units . . . . .	52
8.4.4.	Calibration . . . . .	52
8.5.	Administrative items . . . . .	53
8.5.1.	Status . . . . .	53
8.5.2.	Requester . . . . .	53
8.5.3.	Revision . . . . .	53
8.5.4.	Revision Date . . . . .	53
8.6.	Comments and Remarks . . . . .	54
9.	ICMP Round-trip Latency and Loss Registry Entries . . . . .	54
9.1.	Summary . . . . .	54
9.1.1.	ID (Identifier) . . . . .	54
9.1.2.	Name . . . . .	54
9.1.3.	URI . . . . .	54
9.1.4.	Description . . . . .	55
9.1.5.	Change Controller . . . . .	55
9.1.6.	Version (of Registry Format) . . . . .	55
9.2.	Metric Definition . . . . .	55
9.2.1.	Reference Definition . . . . .	55
9.2.2.	Fixed Parameters . . . . .	56
9.3.	Method of Measurement . . . . .	57
9.3.1.	Reference Method . . . . .	57
9.3.2.	Packet Stream Generation . . . . .	58
9.3.3.	Traffic Filtering (observation) Details . . . . .	59
9.3.4.	Sampling Distribution . . . . .	59
9.3.5.	Run-time Parameters and Data Format . . . . .	59
9.3.6.	Roles . . . . .	59
9.4.	Output . . . . .	60
9.4.1.	Type . . . . .	60
9.4.2.	Reference Definition . . . . .	60
9.4.3.	Metric Units . . . . .	62
9.4.4.	Calibration . . . . .	62
9.5.	Administrative items . . . . .	62
9.5.1.	Status . . . . .	62
9.5.2.	Requester . . . . .	63
9.5.3.	Revision . . . . .	63
9.5.4.	Revision Date . . . . .	63
9.6.	Comments and Remarks . . . . .	63
10.	TCP Round-Trip Delay and Loss Registry Entries . . . . .	63
10.1.	Summary . . . . .	63



10.1.1.	ID (Identifier)	63
10.1.2.	Name	63
10.1.3.	URI	64
10.1.4.	Description	64
10.1.5.	Change Controller	64
10.1.6.	Version (of Registry Format)	64
10.2.	Metric Definition	65
10.2.1.	Reference Definitions	65
10.2.2.	Fixed Parameters	67
10.3.	Method of Measurement	68
10.3.1.	Reference Methods	68
10.3.2.	Packet Stream Generation	70
10.3.3.	Traffic Filtering (observation) Details	70
10.3.4.	Sampling Distribution	70
10.3.5.	Run-time Parameters and Data Format	70
10.3.6.	Roles	71
10.4.	Output	71
10.4.1.	Type	71
10.4.2.	Reference Definition	71
10.4.3.	Metric Units	73
10.4.4.	Calibration	73
10.5.	Administrative items	73
10.5.1.	Status	73
10.5.2.	Requester	73
10.5.3.	Revision	74
10.5.4.	Revision Date	74
10.6.	Comments and Remarks	74
11.	Security Considerations	74
12.	IANA Considerations	74
13.	Acknowledgements	74
14.	References	75
14.1.	Normative References	75
14.2.	Informative References	77
	Authors' Addresses	78

## 1. Introduction

This memo proposes an initial set of entries for the Performance Metrics Registry. It uses terms and definitions from the IPPM literature, primarily [RFC2330].

Although there are several standard templates for organizing specifications of performance metrics (see [RFC7679] for an example of the traditional IPPM template, based to large extent on the Benchmarking Methodology Working Group's traditional template in [RFC1242], and see [RFC6390] for a similar template), none of these templates were intended to become the basis for the columns of an IETF-wide registry of metrics. While examining aspects of metric

specifications which need to be registered, it became clear that none of the existing metric templates fully satisfies the particular needs of a registry.

Therefore, [I-D.ietf-ippm-metric-registry] defines the overall format for a Performance Metrics Registry. Section 5 of [I-D.ietf-ippm-metric-registry] also gives guidelines for those requesting registration of a Metric, that is the creation of entry(s) in the Performance Metrics Registry: "In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose." The process in [I-D.ietf-ippm-metric-registry] also requires that new entries are administered by IANA through Specification Required policy, which will ensure that the metrics are tightly defined.

## 2. Scope

This document defines a set of initial Performance Metrics Registry entries. Most are Active Performance Metrics, which are based on RFCs prepared in the IPPM working group of the IETF, according to their framework [RFC2330] and its updates.

## 3. Registry Categories and Columns

This memo uses the terminology defined in [I-D.ietf-ippm-metric-registry].

This section provides the categories and columns of the registry, for easy reference. An entry (row) therefore gives a complete description of a Registered Metric.

## Legend:

Registry Categories and Columns, shown as

Category	
Column	Column

## Summary

Identifier	Name	URI	Desc.	Reference	Change Controller	Ver
------------	------	-----	-------	-----------	-------------------	-----

## Metric Definition

Reference Definition	Fixed Parameters
----------------------	------------------

## Method of Measurement

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

## Output

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

## Administrative Information

Status	Requester	Rev	Rev.Date
--------	-----------	-----	----------

## Comments and Remarks

## 4. UDP Round-trip Latency and Loss Registry Entries

This section specifies an initial registry entry for the UDP Round-trip Latency, and another entry for UDP Round-trip Loss Ratio.

Note: Each Registry entry only produces a "raw" output or a statistical summary. To describe both "raw" and one or more statistics efficiently, the Identifier, Name, and Output Categories can be split and a single section can specify two or more closely-related metrics. For example, this section specifies two Registry entries with many common columns. See Section 7 for an example specifying multiple Registry entries with many common columns.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes

two closely-related registry entries. As a result, IANA is also asked to assign a corresponding URL to each Named Metric.

#### 4.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

##### 4.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

##### 4.1.2. Name

RTDelay\_Active\_IP-UDP-Periodic\_RFCXXXXsec4\_Seconds\_95Percentile

RTLoss\_Active\_IP-UDP-Periodic\_RFCXXXXsec4\_Percent\_LossRatio

##### 4.1.3. URI

URL: <https://www.iana.org/> ... <name>

##### 4.1.4. Description

RTDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the 95th percentile of their conditional delay distribution.

RTLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

##### 4.1.5. Change Controller

IETF

##### 4.1.6. Version (of Registry Format)

1.0

#### 4.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

##### 4.2.1. Reference Definition

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

##### 4.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- \* DSCP: set to 0

- \* TTL: set to 255
  - \* Protocol: set to 17 (UDP)
  - o IPv6 header values:
    - \* DSCP: set to 0
    - \* Hop Count: set to 255
    - \* Next Header: set to 17 (UDP)
    - \* Flow Label: set to zero
    - \* Extension Headers: none
  - o UDP header values:
    - \* Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
  - o UDP Payload
    - \* total of 100 bytes
- Other measurement parameters:
- o Tmax: a loss threshold waiting time
    - \* 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

#### 4.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

##### 4.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Tmax defined under Fixed Parameters. However, the Periodic stream will be generated according to [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTT value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which MUST be included in the method of measurement for this metric.

#### 4.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see

section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

#### 4.3.3. Traffic Filtering (observation) Details

NA

#### 4.3.4. Sampling Distribution

NA

#### 4.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of



[RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

#### 4.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

#### 4.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 4.4.1. Type

Percentile -- for the conditional distribution of all packets with a valid value of Round-trip delay (undefined delays are excluded), a single value corresponding to the 95th percentile, as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of Round-trip delay for which the Empirical Distribution Function (EDF),  $F(95\text{Percentile}) \geq 95\%$  of the singleton Round-trip delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

##### 4.4.2. Reference Definition

For all outputs ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of

[RFC6991])). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalPkts the count of packets sent by the Src to Dst during the measurement interval.

For

RTDelay\_Active\_IP-UDP-Periodic\_RFCXXXXsec4\_Seconds\_95Percentile:

95Percentile The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.0000000001 seconds (1.0 ns).

For

RTLoss\_Active\_IP-UDP-Periodic\_RFCXXXXsec4\_Percent\_LossRatio:

Percentile The numeric value of the result is expressed in units of lost packets to total packets times 100%, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.0000000001.

#### 4.4.3. Metric Units

The 95th Percentile of Round-trip Delay is expressed in seconds.

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

#### 4.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

#### 4.5. Administrative items

##### 4.5.1. Status

Current

##### 4.5.2. Requester

This RFC number

##### 4.5.3. Revision

1.0

##### 4.5.4. Revision Date

YYYY-MM-DD

#### 4.6. Comments and Remarks

None.

### 5. Packet Delay Variation Registry Entry

This section gives an initial registry entry for a Packet Delay Variation metric.

#### 5.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

##### 5.1.1. ID (Identifier)

<insert numeric identifier, an integer>

##### 5.1.2. Name

OWPDV\_Active\_IP-UDP-Periodic\_RFCXXXXsec5\_Seconds\_95Percentile

### 5.1.3. URI

URL: <https://www.iana.org/> ... <name>

### 5.1.4. Description

An assessment of packet delay variation with respect to the minimum delay observed on the periodic stream, and the Output is expressed as the 95th percentile of the packet delay variation distribution.

### 5.1.5. Change Controller

IETF

### 5.1.6. Version (of Registry Format)

1.0

## 5.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

### 5.2.1. Reference Definition

Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998. [RFC2330]

Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002. [RFC3393]

Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009. [RFC5481]

Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010. [RFC5905]

See sections 2.4 and 3.4 of [RFC3393]. Singleton delay differences measured are referred to by the variable name "ddT" (applicable to all forms of delay variation). However, this metric entry specifies the PDV form defined in section 4.2 of [RFC5481], where the singleton PDV for packet *i* is referred to by the variable name "PDV(*i*)".

### 5.2.2. Fixed Parameters

- o IPv4 header values:
  - \* DSCP: set to 0
  - \* TTL: set to 255
  - \* Protocol: set to 17 (UDP)
- o IPv6 header values:
  - \* DSCP: set to 0
  - \* Hop Count: set to 255
  - \* Next Header: set to 17 (UDP)
  - \* Flow Label: set to zero
  - \* Extension Headers: none
- o UDP header values:
  - \* Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- o UDP Payload
  - \* total of 200 bytes

Other measurement parameters:

- Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].
- F a selection function unambiguously defining the packets from the stream selected for the metric. See section 4.2 of [RFC5481] for the PDV form.

See the Packet Stream generation category for two additional Fixed Parameters.

### 5.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 5.3.1. Reference Method

See section 2.6 and 3.6 of [RFC3393] for general singleton element calculations. This metric entry requires implementation of the PDV form defined in section 4.2 of [RFC5481]. Also see measurement considerations in section 8 of [RFC5481].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

If a standard measurement protocol is employed, then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The chosen measurement protocol will dictate the format of sequence numbers and time-stamps, if they are conveyed in the packet payload.

#### 5.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

dT the duration of the interval for allowed sample start times, with value 1.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

The T0 parameter will be reported as a measured parameter. Parameters incT and dT are Fixed Parameters.

#### 5.3.3. Traffic Filtering (observation) Details

NA

#### 5.3.4. Sampling Distribution

NA

#### 5.3.5. Run-time Parameters and Data Format

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

#### 5.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

#### 5.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 5.4.1. Type

Percentile -- for the conditional distribution of all packets with a valid value of one-way delay (undefined delays are excluded), a single value corresponding to the 95th percentile, as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way PDV for which the Empirical Distribution Function (EDF),  $F(95\text{Percentile}) \geq 95\%$  of the singleton one-way PDV values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

##### 5.4.2. Reference Definition

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].



**95Percentile** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 5.4.3. Metric Units

The 95th Percentile of one-way PDV is expressed in seconds.

#### 5.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

**time\_offset** The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

## 5.5. Administrative items

### 5.5.1. Status

Current

### 5.5.2. Requester

This RFC number

### 5.5.3. Revision

1.0

### 5.5.4. Revision Date

YYYY-MM-DD

## 5.6. Comments and Remarks

Lost packets represent a challenge for delay variation metrics. See section 4.1 of [RFC3393] and the delay variation applicability statement [RFC5481] for extensive analysis and comparison of PDV and an alternate metric, IPDV.

## 6. DNS Response Latency and Loss Registry Entries

This section gives initial registry entries for DNS Response Latency and Loss from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different names. RFC 2681 [RFC2681] defines a Round-trip delay metric. We build on that metric by specifying several of the input parameters to precisely define two metrics for measuring DNS latency and loss.

Note to IANA: Each Registry "Name" below specifies a single registry entry, whose output format varies in accordance with the name.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

### 6.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

#### 6.1.1. ID (Identifier)

<insert numeric identifier, an integer>

IANA is asked to assign different numeric identifiers to each of the two Named Metrics.

#### 6.1.2. Name

RTDNS\_Active\_IP-UDP-Poisson\_RFCXXXXsec6\_Seconds\_Raw

RLDNS\_Active\_IP-UDP-Poisson\_RFCXXXXsec6\_Logical\_Raw

#### 6.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 6.1.4. Description

This is a metric for DNS Response performance from a network user's perspective, for a specific named resource. The metric can be measured repeatedly using different resource names.

RTDNS: This metric assesses the response time, the interval from the query transmission to the response.

RLDNS: This metric indicates that the response was deemed lost. In other words, the response time exceeded the maximum waiting time.

#### 6.1.5. Change Controller

IETF

#### 6.1.6. Version (of Registry Format)

1.0

### 6.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

#### 6.2.1. Reference Definition

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987. (and updates)

[RFC1035]

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

For DNS Response Latency, the entities in [RFC1035] must be mapped to [RFC2681]. The Local Host with its User Program and Resolver take the role of "Src", and the Foreign Name Server takes the role of "Dst".

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst at T" is directionally ambiguous in the text, this metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both response time and loss metrics employ a maximum waiting time for received responses, so the count of lost packets to total packets sent is the basis for the loss determination as per Section 4.3 of [RFC6673].

#### 6.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- \* DSCP: set to 0
- \* TTL set to 255
- \* Protocol: set to 17 (UDP)

- o IPv6 header values:

- \* DSCP: set to 0
- \* Hop Count: set to 255
- \* Next Header: set to 17 (UDP)
- \* Flow Label: set to zero
- \* Extension Headers: none
- o UDP header values:
  - \* Source port: 53
  - \* Destination port: 53
  - \* Checksum: the checksum must be calculated and the non-zero checksum included in the header
- o Payload: The payload contains a DNS message as defined in RFC 1035 [RFC1035] with the following values:
  - \* The DNS header section contains:
    - + Identification (see the Run-time column)
    - + QR: set to 0 (Query)
    - + OPCODE: set to 0 (standard query)
    - + AA: not set
    - + TC: not set
    - + RD: set to one (recursion desired)
    - + RA: not set
    - + RCODE: not set
    - + QDCOUNT: set to one (only one entry)
    - + ANCOUNT: not set
    - + NSCOUNT: not set
    - + ARCOUNT: not set

- \* The Question section contains:
  - + QNAME: the Fully Qualified Domain Name (FQDN) provided as input for the test, see the Run-time column
  - + QTYPE: the query type provided as input for the test, see the Run-time column
  - + QCLASS: set to 1 for IN
- \* The other sections do not contain any Resource Records.

Other measurement parameters:

- o Tmax: a loss threshold waiting time (and to help disambiguate queries)
  - \* 5.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

Observation: reply packets will contain a DNS response and may contain RRs.

### 6.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 6.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Timeout defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a response packet lost. Lost packets SHALL be designated as having undefined delay and counted for the RLDNS metric.

The calculations on the delay (RTT) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process

which calculates the RTT value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving reply.

DNS Messages bearing Queries provide for random ID Numbers in the Identification header field, so more than one query may be launched while a previous request is outstanding when the ID Number is used. Therefore, the ID Number MUST be retained at the Src and included with each response packet to disambiguate packet reordering if it occurs.

IF a DNS response does not arrive within Tmax, the response time RTDNS is undefined, and RLDNS = 1. The Message ID SHALL be used to disambiguate the successive queries that are otherwise identical.

Since the ID Number field is only 16 bits in length, it places a limit on the number of simultaneous outstanding DNS queries during a stress test from a single Src address.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. However, the DNS Server is expected to perform all required functions to prepare and send a response, so the response time will include processing time and network delay. Section 8 of [RFC6673] presents additional requirements which SHALL be included in the method of measurement for this metric.

In addition to operations described in [RFC2681], the Src MUST parse the DNS headers of the reply and prepare the query response information for subsequent reporting as a measured result, along with the Round-Trip Delay.

#### 6.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. The reciprocal of lambda is the average packet spacing, thus the Run-time Parameter is  $\text{Reciprocal\_lambda} = 1/\text{lambda}$ , in seconds.

Method 3 is used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Run-time Parameters), and the Src sends each packet at the computed times.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

#### 6.3.3. Traffic Filtering (observation) Details

NA

#### 6.3.4. Sampling Distribution

NA

#### 6.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of



[RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

Reciprocal\_lambda average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

Trunc Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value).

ID The 16-bit identifier assigned by the program that generates the query, and which must vary in successive queries (a list of IDs is needed), see Section 4.1.1 of [RFC1035]. This identifier is copied into the corresponding reply and can be used by the requester (Src) to match-up replies to outstanding queries.

QNAME The domain name of the Query, formatted as specified in section 4 of [RFC6991].

QTYPE The Query Type, which will correspond to the IP address family of the query (decimal 1 for IPv4 or 28 for IPv6, formatted as a uint16, as per section 9.2 of [RFC6020]).

#### 6.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

#### 6.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 6.4.1. Type

Raw -- for each DNS Query packet sent, sets of values as defined in the next column, including the status of the response, only assigning delay values to successful query-response pairs.

#### 6.4.2. Reference Definition

For all outputs:

T the time the DNS Query was sent during the measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

dT The time value of the round-trip delay to receive the DNS response, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]. This value is undefined when the response packet is not received at Src within waiting time Tmax seconds.

Rcode The value of the Rcode field in the DNS response header, expressed as a uint64 as specified in section 9.2 of [RFC6020]. Non-zero values convey errors in the response, and such replies must be analyzed separately from successful requests.

#### 6.4.3. Metric Units

RTDNS: Round-trip Delay, dT, is expressed in seconds.

RTLDNS: the Logical value, where 1 = Lost and 0 = Received.

#### 6.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address and payload manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the

portion of the Output result resolution which is the result of system noise, and thus inaccurate.

#### 6.5. Administrative items

##### 6.5.1. Status

Current

##### 6.5.2. Requester

This RFC number

##### 6.5.3. Revision

1.0

##### 6.5.4. Revision Date

YYYY-MM-DD

#### 6.6. Comments and Remarks

None

### 7. UDP Poisson One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Poisson One-way Delay, and one for UDP Poisson One-way Loss.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

#### 7.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

#### 7.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the six Metrics.

#### 7.1.2. Name

OWDelay\_Active\_IP-UDP-Poisson-  
Payload250B\_RFCXXXXsec7\_Seconds\_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss\_Active\_IP-UDP-Poisson-  
Payload250B\_RFCXXXXsec7\_Percent\_LossRatio

#### 7.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 7.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

## 7.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

### 7.2.1. Reference Definition

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

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

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

### 7.2.2. Fixed Parameters

#### Type-P:

- o IPv4 header values:

- \* DSCP: set to 0
- \* TTL: set to 255
- \* Protocol: Set to 17 (UDP)

- o IPv6 header values:

- \* DSCP: set to 0
- \* Hop Count: set to 255
- \* Next Header: set to 17 (UDP)
- \* Flow Label: set to zero
- \* Extension Headers: none

- o UDP header values:

- \* Checksum: the checksum MUST be calculated and the non-zero checksum included in the header

- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]

- \* Security features in use influence the number of Padding octets.
- \* 250 octets total, including the TWAMP format type, which MUST be reported.

#### Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

### 7.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 7.3.1. Reference Method

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

#### 7.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 11.1.3 of RFC 2681 [RFC2330] provides three methods to generate Poisson sampling intervals. The reciprocal of lambda is the

average packet spacing, thus the Run-time Parameter is  $\text{Reciprocal\_lambda} = 1/\text{lambda}$ , in seconds.

Method 3 SHALL be used, where given a start time (Run-time Parameter), the subsequent send times are all computed prior to measurement by computing the pseudo-random distribution of inter-packet send times, (truncating the distribution as specified in the Parameter Trunc), and the Src sends each packet at the computed times.

Note that Trunc is the upper limit on inter-packet times in the Poisson distribution. A random value greater than Trunc is set equal to Trunc instead.

**Reciprocal\_lambda** average packet interval for Poisson Streams expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].  $\text{Reciprocal\_lambda} = 1$  second.

**Trunc** Upper limit on Poisson distribution expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) with resolution of 0.0001 seconds (0.1 ms), and with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905] (values above this limit will be clipped and set to the limit value).  $\text{Trunc} = 30.0000$  seconds.

#### 7.3.3. Traffic Filtering (observation) Details

NA

#### 7.3.4. Sampling Distribution

NA

#### 7.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

**Src** the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])



Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

#### 7.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src. This is the TWAMP Session-Reflector.

#### 7.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 7.4.1. Type

See subsection titles below for Types.

##### 7.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

#### 7.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF),  $F(95\text{Percentile}) \geq 95\%$  of the singleton one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

**95Percentile** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 7.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

**Mean** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001

seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 7.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 7.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index,  $j$ , where  $1 \leq j \leq N$   
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$  for all  $n$

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

## 7.4.2.5. Std\_Dev

The Std\_Dev SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 6.1.4 of [RFC6049] for a closely related method for calculating this statistic. The formula is the classic calculation for standard deviation of a population.

Define Population Std\_Dev\_Delay as follows:

(where all packets  $n = 1$  through  $N$  have a value for Delay[n], and MeanDelay calculated as in 7.4.2.2), and SQRT[] is the Square Root function:

$$\text{Std\_Dev} = \text{SQRT} \left[ \frac{1}{N} \sum_{n=1}^N (\text{Delay}[n] - \text{MeanDelay})^2 \right]$$

Std\_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

## 7.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds.

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

## 7.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g.,

deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets [RFC5905] are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

## 7.5. Administrative items

### 7.5.1. Status

Current

### 7.5.2. Requester

This RFC number

### 7.5.3. Revision

1.0

#### 7.5.4. Revision Date

YYYY-MM-DD

#### 7.6. Comments and Remarks

None

### 8. UDP Periodic One-way Delay and Loss Registry Entries

This section specifies five initial registry entries for the UDP Periodic One-way Delay, and one for UDP Periodic One-way Loss.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes six closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

#### 8.1. Summary

This category includes multiple indexes to the registry entries, the element ID and metric name.

##### 8.1.1. ID (Identifier)

IANA is asked to assign a different numeric identifiers to each of the six Metrics.

##### 8.1.2. Name

OWDelay\_Active\_IP-UDP-Periodic20m-  
Payload142B\_RFCXXXXsec8\_Seconds\_<statistic>

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max

- o StdDev

OWLoss\_Active\_IP-UDP-Periodic-  
Payload142B\_RFCXXXXsec8\_Percent\_LossRatio

#### 8.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 8.1.4. Description

OWDelay: This metric assesses the delay of a stream of packets exchanged between two hosts (or measurement points), and reports the <statistic> One-way delay for all successfully exchanged packets based on their conditional delay distribution.

where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

OWLoss: This metric assesses the loss ratio of a stream of packets exchanged between two hosts (which are the two measurement points), and the Output is the One-way loss ratio for all successfully received packets expressed as a percentage.

### 8.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

#### 8.2.1. Reference Definition

For Delay:

Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<http://www.rfc-editor.org/info/rfc7679>>.

[RFC7679]

Morton, A., and Stephan, E., "Spatial Composition of Metrics", RFC 6049, January 2011.

[RFC6049]

Section 3.4 of [RFC7679] provides the reference definition of the singleton (single value) One-way delay metric. Section 4.4 of [RFC7679] provides the reference definition expanded to cover a multi-value sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Only successful packet transfers with finite delay are included in the sample, as prescribed in section 4.1.2 of [RFC6049].

For loss:

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

Section 2.4 of [RFC7680] provides the reference definition of the singleton (single value) one-way loss metric. Section 3.4 of [RFC7680] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

#### 8.2.2. Fixed Parameters

Type-P:

- o IPv4 header values:

- \* DSCP: set to 0
- \* TTL: set to 255
- \* Protocol: Set to 17 (UDP)

- o IPv6 header values:

- \* DSCP: set to 0
- \* Hop Count: set to 255
- \* Next Header: set to 17 (UDP)



- \* Flow Label: set to zero
- \* Extension Headers: none
- o UDP header values:
  - \* Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- o UDP Payload: TWAMP Test Packet Formats, Section 4.1.2 of [RFC5357]
  - \* Security features in use influence the number of Padding octets.
  - \* 142 octets total, including the TWAMP format (and format type MUST be reported, if used)

Other measurement parameters:

Tmax: a loss threshold waiting time with value 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

See the Packet Stream generation category for two additional Fixed Parameters.

### 8.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 8.3.1. Reference Method

The methodology for this metric is defined as Type-P-One-way-Delay-Poisson-Stream in section 3.6 of [RFC7679] and section 4.6 of [RFC7679] using the Type-P and Tmax defined under Fixed Parameters. However, a Periodic stream is used, as defined in [RFC3432].

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the OWLoss metric.

The calculations on the one-way delay SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the one-way delay value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet.

Since a standard measurement protocol is employed [RFC5357], then the measurement process will determine the sequence numbers or timestamps applied to test packets after the Fixed and Runtime parameters are passed to that process. The measurement protocol dictates the format of sequence numbers and time-stamps conveyed in the TWAMP-Test packet payload.

#### 8.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

Section 3 of [RFC3432] prescribes the method for generating Periodic streams using associated parameters.

incT the nominal duration of inter-packet interval, first bit to first bit, with value 0.0200 expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

dT the duration of the interval for allowed sample start times, with value 1.0000, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

T0 the actual start time of the periodic stream, determined from T0 and dT.

NOTE: an initiation process with a number of control exchanges resulting in unpredictable start times (within a time interval) may be sufficient to avoid synchronization of periodic streams, and therefore a valid replacement for selecting a start time at random from a fixed interval.

These stream parameters will be specified as Run-time parameters.

#### 8.3.3. Traffic Filtering (observation) Details

NA

#### 8.3.4. Sampling Distribution

NA

#### 8.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Tf a time, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a end time date is ignored and Tf is interpreted as the Duration of the measurement interval.

#### 8.3.6. Roles

Src launches each packet and waits for return transmissions from Dst. This is the TWAMP Session-Sender.

Dst waits for each packet from Src and sends a return packet to Src.  
This is the TWAMP Session-Reflector.

#### 8.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 8.4.1. Type

See subsection titles in Reference Definition for Latency Types.

##### 8.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

###### 8.4.2.1. Percentile95

The 95th percentile SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3 of [RFC3393] for details on the percentile statistic (where Round-trip delay should be substituted for "ipdv").

The percentile = 95, meaning that the reported delay, "95Percentile", is the smallest value of one-way delay for which the Empirical Distribution Function (EDF),  $F(95\text{Percentile}) \geq 95\%$  of the singleton

one-way delay values in the conditional distribution. See section 11.3 of [RFC2330] for the definition of the percentile statistic using the EDF.

**95Percentile** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 8.4.2.2. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

**Mean** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 8.4.2.3. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

**Min** The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 8.4.2.4. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index,  $j$ , where  $1 \leq j \leq N$   
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$  for all  $n$

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 8.4.2.5. Std\_Dev

The Std\_Dev SHALL be calculated using the conditional distribution of all packets with a finite value of One-way delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is the classic calculation for standard deviation of a population.

Define Population Std\_Dev\_Delay as follows:  
 (where all packets  $n = 1$  through  $N$  have a value for Delay[n],  
 and MeanDelay calculated as in 7.4.2.2), and SQRT[] is the  
 Square Root function:

$$\text{Std\_Dev} = \text{SQRT} \left[ \frac{1}{(N)} \sum_{n=1}^N (\text{Delay}[n] - \text{MeanDelay})^2 \right]$$

Std\_Dev The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 8.4.3. Metric Units

The <statistic> of One-way Delay is expressed in seconds, where <statistic> is one of:

- o 95Percentile
- o Mean
- o Min
- o Max
- o StdDev

The One-way Loss Ratio is expressed as a percentage of lost packets to total packets sent.

#### 8.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets [RFC5905] of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets [RFC5905] are smoothed, thus the random variation is not usually represented in the results).

`time_offset` The time value of the result is expressed in units of seconds, as a signed value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result. In any measurement, the measurement function SHOULD report its current estimate of time offset [RFC5905] as an indicator of the degree of synchronization.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

## 8.5. Administrative items

### 8.5.1. Status

Current

### 8.5.2. Requester

This RFC number

### 8.5.3. Revision

1.0

### 8.5.4. Revision Date

YYYY-MM-DD



## 8.6. Comments and Remarks

None.

## 9. ICMP Round-trip Latency and Loss Registry Entries

This section specifies three initial registry entries for the ICMP Round-trip Latency, and another entry for ICMP Round-trip Loss Ratio.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There is an additional metric name for the Loss metric.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes two closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

### 9.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

#### 9.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

#### 9.1.2. Name

RTDelay\_Active\_IP-ICMP-SendOnRcv\_RFCXXXXsec9\_Seconds\_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss\_Active\_IP-ICMP-SendOnRcv\_RFCXXXXsec9\_Percent\_LossRatio

#### 9.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 9.1.4. Description

RTDelay: This metric assesses the delay of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss: This metric assesses the loss ratio of a stream of ICMP packets exchanged between two hosts (which are the two measurement points), and the Output is the Round-trip loss ratio for all successfully exchanged packets expressed as a percentage.

#### 9.1.5. Change Controller

IETF

#### 9.1.6. Version (of Registry Format)

1.0

### 9.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

#### 9.2.1. Reference Definition

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample. Note that terms such as singleton and sample are defined in Section 11 of [RFC2330].

Note that although the [RFC2681] definition of "Round-trip-Delay between Src and Dst" is directionally ambiguous in the text, this

metric tightens the definition further to recognize that the host in the "Src" role will send the first packet to "Dst", and ultimately receive the corresponding return packet from "Dst" (when neither are lost).

Finally, note that the variable "dT" is used in [RFC2681] to refer to the value of Round-trip delay in metric definitions and methods. The variable "dT" has been re-used in other IPPM literature to refer to different quantities, and cannot be used as a global variable name.

Morton, A., "Round-trip Packet Loss Metrics", RFC 6673, August 2012.

[RFC6673]

Both delay and loss metrics employ a maximum waiting time for received packets, so the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

#### 9.2.2. Fixed Parameters

Type-P as defined in Section 13 of [RFC2330]:

- o IPv4 header values:

- \* DSCP: set to 0
- \* TTL: set to 255
- \* Protocol: Set to 01 (ICMP)

- o IPv6 header values:

- \* DSCP: set to 0
- \* Hop Count: set to 255
- \* Next Header: set to 128 decimal (ICMP)
- \* Flow Label: set to zero
- \* Extension Headers: none

- o ICMP header values:

- \* Type: 8 (Echo Request)
- \* Code: 0

- \* Checksum: the checksum MUST be calculated and the non-zero checksum included in the header
- \* (Identifier and Sequence Number set at Run-Time)
- o ICMP Payload
  - \* total of 32 bytes of random info, constant per test.

Other measurement parameters:

- o Tmax: a loss threshold waiting time
  - \* 3.0, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms), with lossless conversion to/from the 32-bit NTP timestamp as per section 6 of [RFC5905].

### 9.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 9.3.1. Reference Method

The methodology for this metric is defined as Type-P-Round-trip-Delay-Poisson-Stream in section 2.6 of RFC 2681 [RFC2681] and section 3.6 of RFC 2681 [RFC2681] using the Type-P and Tmax defined under Fixed Parameters.

The reference method distinguishes between long-delayed packets and lost packets by implementing a maximum waiting time for packet arrival. Tmax is the waiting time used as the threshold to declare a packet lost. Lost packets SHALL be designated as having undefined delay, and counted for the RTLoss metric.

The calculations on the delay (RTD) SHALL be performed on the conditional distribution, conditioned on successful packet arrival within Tmax. Also, when all packet delays are stored, the process which calculates the RTD value MUST enforce the Tmax threshold on stored values before calculations. See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

The reference method requires some way to distinguish between different packets in a stream to establish correspondence between sending times and receiving times for each successfully-arriving packet. Sequence numbers or other send-order identification MUST be retained at the Src or included with each packet to disambiguate packet reordering if it occurs.

The measurement process will determine the sequence numbers applied to test packets after the Fixed and Runtime parameters are passed to that process. The ICMP measurement process and protocol will dictate the format of sequence numbers and other identifiers.

Refer to Section 4.4 of [RFC6673] for expanded discussion of the instruction to "send a Type-P packet back to the Src as quickly as possible" in Section 2.6 of RFC 2681 [RFC2681]. Section 8 of [RFC6673] presents additional requirements which MUST be included in the method of measurement for this metric.

#### 9.3.2. Packet Stream Generation

This section gives the details of the packet traffic which is the basis for measurement. In IPPM metrics, this is called the Stream, and can easily be described by providing the list of stream parameters.

The ICMP metrics use a sending discipline called "SendOnRcv" or Send On Receive. This is a modification of Section 3 of [RFC3432], which prescribes the method for generating Periodic streams using associated parameters as defined below for this description:

incT the nominal duration of inter-packet interval, first bit to first bit

dT the duration of the interval for allowed sample start times

The incT stream parameter will be specified as a Run-time parameter, and dT is not used in SendOnRcv.

A SendOnRcv sender behaves exactly like a Periodic stream generator while all reply packets arrive with  $RTD < incT$ , and the inter-packet interval will be constant.

If a reply packet arrives with  $RTD \geq incT$ , then the inter-packet interval for the next sending time is nominally RTD.

If a reply packet fails to arrive within Tmax, then the inter-packet interval for the next sending time is nominally Tmax.

If an immediate send on reply arrival is desired, then set incT=0.

#### 9.3.3. Traffic Filtering (observation) Details

NA

#### 9.3.4. Sampling Distribution

NA

#### 9.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the Src Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the Dst Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

incT the nominal duration of inter-packet interval, first bit to first bit, expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 4 (see section 9.3 of [RFC6020]) and with resolution of 0.0001 seconds (0.1 ms).

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Tf is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Count The total count of ICMP Echo Requests to send, formatted as a uint16, as per section 9.2 of [RFC6020].

(see the Packet Stream Generation section for additional Run-time parameters)

#### 9.3.6. Roles

Src launches each packet and waits for return transmissions from Dst.

Dst waits for each packet from Src and sends a return packet to Src.

#### 9.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 9.4.1. Type

See subsection titles in Reference Definition for Latency Types.

LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 6.1 of [RFC6673].

##### 9.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

TotalCount the count of packets actually sent by the Src to Dst during the measurement interval.

For LossRatio -- the count of lost packets to total packets sent is the basis for the loss ratio calculation as per Section 4.1 of [RFC7680].

For each <statistic>, one of the following sub-sections apply:

###### 9.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 9.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 9.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay } [j])$$

such that for some index,  $j$ , where  $1 \leq j \leq N$   
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$  for all  $n$

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001



seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 9.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Loss Ratio is expressed as a percentage of lost packets to total packets sent.

#### 9.4.4. Calibration

Section 3.7.3 of [RFC7679] provides a means to quantify the systematic and random errors of a time measurement. In-situ calibration could be enabled with an internal loopback at the Source host that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

When a measurement controller requests a calibration measurement, the loopback is applied and the result is output in the same format as a normal measurement with additional indication that it is a calibration result.

Both internal loopback calibration and clock synchronization can be used to estimate the available accuracy of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.

#### 9.5. Administrative items

##### 9.5.1. Status

Current

#### 9.5.2. Requester

This RFC number

#### 9.5.3. Revision

1.0

#### 9.5.4. Revision Date

YYYY-MM-DD

#### 9.6. Comments and Remarks

None

### 10. TCP Round-Trip Delay and Loss Registry Entries

This section specifies three initial registry entries for the Passive assessment of TCP Round-Trip Delay (RTD) and another entry for TCP Round-trip Loss Count.

IANA Note: Registry "Name" below specifies multiple registry entries, whose output format varies according to the <statistic> element of the name that specifies one form of statistical summary. There are two additional metric names for Singleton RT Delay and Packet Count metrics.

All column entries beside the ID, Name, Description, and Output Reference Method categories are the same, thus this section proposes four closely-related registry entries. As a result, IANA is also asked to assign corresponding URLs to each Named Metric.

#### 10.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

##### 10.1.1. ID (Identifier)

IANA is asked to assign different numeric identifiers to each of the four Named Metrics.

##### 10.1.2. Name

RTDelay\_Passive\_IP-TCP\_RFCXXXXsec10\_Seconds\_<statistic>

where <statistic> is one of:

- o Mean
- o Min
- o Max

RTDelay\_Passive\_IP-TCP-HS\_RFCXXXXsec10\_Seconds\_Singleton

Note that a mid-point observer only has the opportunity to compose a single RTDelay on the TCP Hand Shake.

RTLoss\_Passive\_IP-TCP\_RFCXXXXsec10\_Packet\_Count

#### 10.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 10.1.4. Description

RTDelay: This metric assesses the round-trip delay of TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the Round-trip delay for all successfully exchanged packets expressed as the <statistic> of their conditional delay distribution, where <statistic> is one of:

- o Mean
- o Min
- o Max

RTLoss: This metric assesses the estimated loss count for TCP packets constituting a single connection, exchanged between two hosts. We consider the measurement of round-trip delay based on a single Observation Point [RFC7011] somewhere in the network. The Output is the estimated Loss Count for the measurement interval.

#### 10.1.5. Change Controller

IETF

#### 10.1.6. Version (of Registry Format)

1.0

## 10.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the RFC reference and values of input factors, called fixed parameters.

### 10.2.1. Reference Definitions

Although there is no RFC that describes passive measurement of Round-Trip Delay, the parallel definition for Active measurement is:

Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2681]

This metric definition uses the terms singleton and sample as defined in Section 11 of [RFC2330]. (Section 2.4 of [RFC2681] provides the reference definition of the singleton (single value) Round-trip delay metric. Section 3.4 of [RFC2681] provides the reference definition expanded to cover a multi-singleton sample.)

With the Observation Point [RFC7011] (OP) typically located between the hosts participating in the TCP connection, the Round-trip Delay metric requires two individual measurements between the OP and each host, such that the Spatial Composition [RFC6049] of the measurements yields a Round-trip Delay singleton (we are extending the composition of one-way subpath delays to subpath round-trip delay).

Using the direction of TCP SYN transmission to anchor the nomenclature, host A sends the SYN and host B replies with SYN-ACK during connection establishment. The direction of SYN transfer is considered the Forward direction of transmission, from A through OP to B (Reverse is B through OP to A).

Traffic filters reduce the packet stream at the OP to a Qualified bidirectional flow of packets.

In the definitions below, Corresponding Packets are transferred in different directions and convey a common value in a TCP header field that establishes correspondence (to the extent possible). Examples may be found in the TCP timestamp fields.

For a real number,  $RTD_{fwd}$ ,  $\gg$  the Round-trip Delay in the Forward direction from OP to host B at time  $T'$  is  $RTD_{fwd}$   $\ll$  it is REQUIRED that OP observed a Qualified Packet to host B at wire-time  $T'$ , that host B received that packet and sent a Corresponding Packet back to

host A, and OP observed the Corresponding Packet at wire-time  $T' + \text{RTD\_fwd}$ .

For a real number,  $\text{RTD\_rev}$ ,  $\gg$  the Round-trip Delay in the Reverse direction from OP to host A at time  $T''$  is  $\text{RTD\_rev} \ll$  it is REQUIRED that OP observed a Qualified Packet to host A at wire-time  $T''$ , that host A received that packet and sent a Corresponding Packet back to host B, and that OP observed the Corresponding Packet at wire-time  $T'' + \text{RTD\_rev}$ .

Ideally, the packet sent from host B to host A in both definitions above SHOULD be the same packet (or, when measuring  $\text{RTD\_rev}$  first, the packet from host A to host B in both definitions should be the same).

The REQUIRED Composition Function for a singleton of Round-trip Delay at time T (where T is the earliest of  $T'$  and  $T''$  above) is:

$$\text{RTDelay} = \text{RTD\_fwd} + \text{RTD\_rev}$$

Note that when OP is located at host A or host B, one of the terms composing  $\text{RTDelay}$  will be zero or negligible.

When the Qualified and Corresponding Packets are a TCP-SYN and a TCP-SYN-ACK, then  $\text{RTD\_fwd} == \text{RTD\_HS\_fwd}$ .

When the Qualified and Corresponding Packets are a TCP-SYN-ACK and a TCP-ACK, then  $\text{RTD\_rev} == \text{RTD\_HS\_rev}$ .

The REQUIRED Composition Function for a singleton of Round-trip Delay for the connection Hand Shake:

$$\text{RTDelay\_HS} = \text{RTD\_HS\_fwd} + \text{RTD\_HS\_rev}$$

The definition of Round-trip Loss Count uses the nomenclature developed above, based on observation of the TCP header sequence numbers and storing the sequence number gaps observed. Packet Losses can be inferred from:

- o Out-of-order segments: TCP segments are transmitted with monotonically increasing sequence numbers, but these segments may be received out of order. Section 3 of [RFC4737] describes the notion of "next expected" sequence numbers which can be adapted to TCP segments (for the purpose of detecting reordered packets). Observation of out-of-order segments indicates loss on the path prior to the OP, and creates a gap.

- o Duplicate segments: Section 2 of [RFC5560] defines identical packets and is suitable for evaluation of TCP packets to detect duplication. Observation of duplicate segments \*without a corresponding gap\* indicates loss on the path following the OP (because they overlap part of the delivered sequence numbers already observed at OP).

Each observation of an out-of-order or duplicate infers a singleton of loss, but composition of Round-trip Loss Counts will be conducted over a measurement interval which is synonymous with a single TCP connection.

With the above observations in the Forward direction over a measurement interval, the count of out-of-order and duplicate segments is defined as RTL\_fwd. Comparable observations in the Reverse direction are defined as RTL\_rev.

For a measurement interval (corresponding to a single TCP connection), T0 to Tf, the REQUIRED Composition Function for a the two single-direction counts of inferred loss is:

$RTL_{Loss} = RTL_{fwd} + RTL_{rev}$

#### 10.2.2. Fixed Parameters

##### Traffic Filters:

- o IPv4 header values:
  - \* DSCP: set to 0
  - \* Protocol: Set to 06 (TCP)
- o IPv6 header values:
  - \* DSCP: set to 0
  - \* Hop Count: set to 255
  - \* Next Header: set to 6 (TCP)
  - \* Flow Label: set to zero
  - \* Extension Headers: none
- o TCP header values:
  - \* Flags: ACK, SYN, FIN, set as required

- \* Timestamp Option (TSopt): Set

- + Section 3.2 of [RFC7323]

### 10.3. Method of Measurement

This category includes columns for references to relevant sections of the RFC(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 10.3.1. Reference Methods

The foundation methodology for this metric is defined in Section 4 of [RFC7323] using the Timestamp Option with modifications that allow application at a mid-path Observation Point (OP) [RFC7011]. Further details and applicable heuristics were derived from [Strowes] and [Trammell-14].

The Traffic Filter at the OP is configured to observe a single TCP connection. When the SYN, SYN-ACK, ACK handshake occurs, it offers the first opportunity to measure both RTD\_fwd (on the SYN to SYN-ACK pair) and RTD\_rev (on the SYN-ACK to ACK pair). Label this singleton of RTDelay as RTDelay\_HS (composed using the forward and reverse measurement pair). RTDelay\_HS SHALL be treated separately from other RTDelays on data-bearing packets and their ACKs. The RTDelay\_HS value MAY be used as a sanity check on other Composed values of RTDelay.

For payload bearing packets, the OP measures the time interval between observation of a packet with Sequence Number *s*, and the corresponding ACK with same Sequence number. When the payload is transferred from host A to host B, the observed interval is RTD\_fwd.

Because many data transfers are unidirectional (say, in the Forward direction from host A to host B), it is necessary to use pure ACK packets with Timestamp (TSval) and their Timestamp value echo to perform a RTD\_rev measurement. The time interval between observation of the ACK from B to A, and the corresponding packet with Timestamp echo (TSecr) is the RTD\_rev.

#### Delay Measurement Filtering Heuristics:

If Data payloads were transferred in both Forward and Reverse directions, then the Round-Trip Time Measurement Rule in Section 4.1 of [RFC7323] could be applied. This rule essentially excludes any measurement using a packet unless it makes progress in the transfer (advances the left edge of the send window, consistent with [Strowes]).

A different heuristic from [Trammell-14] is to exclude any RTD\_rev that is larger than previously observed values. This would tend to exclude Reverse measurements taken when the Application has no data ready to send, because considerable time could be added to RTD\_rev from this source of error.

Note that the above Heuristic assumes that host A is sending data. Host A expecting a download would mean that this heuristic should be applied to RTD\_fwd.

The statistic calculations to summarize the delay (RTDelay) SHALL be performed on the conditional distribution, conditioned on successful Forward and Reverse measurements which follow the Heuristics.

#### Method for Inferring Loss:

The OP tracks sequence numbers and stores gaps for each direction of transmission, as well as the next-expected sequence number as in [Trammell-14] and [RFC4737]. Loss is inferred from Out-of-order segments and Duplicate segments.

#### Loss Measurement Filtering Heuristics:

[Trammell-14] adds a window of evaluation based on the RTDelay.

Distinguish Re-ordered from OOO due to loss, because sequence number gap is filled during the same RTDelay window. Segments detected as re-ordered according to [RFC4737] MUST reduce the Loss Count inferred from Out-of-order segments.

Spurious (unneeded) retransmissions (observed as duplicates) can also be reduced this way, as described in [Trammell-14].

#### Sources of Error:

The principal source of RTDelay error is the host processing time to return a packet that defines the termination of a time interval. The heuristics above intend to mitigate these errors by excluding measurements where host processing time is a significant part of RTD\_fwd or RTD\_rev.

A key source of RTLoss error is observation loss, described in section 3 of [Trammell-14].



### 10.3.2. Packet Stream Generation

NA

### 10.3.3. Traffic Filtering (observation) Details

The Fixed Parameters above give a portion of the Traffic Filter. Other aspects will be supplied as Run-time Parameters (below).

### 10.3.4. Sampling Distribution

This metric requires a complete sample of all packets that qualify according to the Traffic Filter criteria.

### 10.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

Src the IP address of the host in the host A Role (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see Section 4 of [RFC6991])

Dst the IP address of the host in the host B (format ipv4-address-no-zone value for IPv4, or ipv6-address-no-zone value for IPv6, see section 4 of [RFC6991])

T0 a time, the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. When T0 is "all-zeros", a start time is unspecified and Td is to be interpreted as the Duration of the measurement interval. The start time is controlled through other means.

Td Optionally, the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]), or the duration (see T0). The UTC Time Zone is required by Section 6.1 of [RFC2330]. Alternatively, the end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK packets exchanged between host A and B effectively ends the interval.

TTL or Hop Limit Set at desired value.

#### 10.3.6. Roles

host A launches the SYN packet to open the connection, and synonymous with an IP address.

host B replies with the SYN-ACK packet to open the connection, and synonymous with an IP address.

#### 10.4. Output

This category specifies all details of the Output of measurements using the metric.

##### 10.4.1. Type

See subsection titles in Reference Definition for RTDelay Types.

For RTLoss -- the count of lost packets.

##### 10.4.2. Reference Definition

For all output types ---

T0 the start of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330].

Tf the end of a measurement interval, (format "date-and-time" as specified in Section 5.6 of [RFC3339], see also Section 3 of [RFC6991]). The UTC Time Zone is required by Section 6.1 of [RFC2330]. The end of the measurement interval MAY be controlled by the measured connection, where the second pair of FIN and ACK packets exchanged between host A and B effectively ends the interval.

... ..

For RTDelay\_HS -- the Round trip delay of the Handshake.

For RTLoss -- the count of lost packets.

For each <statistic>, one of the following sub-sections apply:

## 10.4.2.1. Mean

The mean SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.2.2 of [RFC6049] for details on calculating this statistic, and 4.2.3 of [RFC6049].

Mean The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

## 10.4.2.2. Min

The minimum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for details on calculating this statistic, and 4.3.3 of [RFC6049].

Min The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

## 10.4.2.3. Max

The maximum SHALL be calculated using the conditional distribution of all packets with a finite value of Round-trip delay (undefined delays are excluded), a single value as follows:

See section 4.1 of [RFC3393] for details on the conditional distribution to exclude undefined values of delay, and Section 5 of [RFC6703] for background on this analysis choice.

See section 4.3.2 of [RFC6049] for a closely related method for calculating this statistic, and 4.3.3 of [RFC6049]. The formula is as follows:

$$\text{Max} = (\text{FiniteDelay}[j])$$

such that for some index,  $j$ , where  $1 \leq j \leq N$   
 $\text{FiniteDelay}[j] \geq \text{FiniteDelay}[n]$  for all  $n$

Max The time value of the result is expressed in units of seconds, as a positive value of type decimal64 with fraction digits = 9 (see section 9.3 of [RFC6020]) with resolution of 0.000000001 seconds (1.0 ns), and with lossless conversion to/from the 64-bit NTP timestamp as per section 6 of RFC [RFC5905]

#### 10.4.3. Metric Units

The <statistic> of Round-trip Delay is expressed in seconds, where <statistic> is one of:

- o Mean
- o Min
- o Max

The Round-trip Delay of the Hand Shake is expressed in seconds.

The Round-trip Loss Count is expressed as a number of packets.

#### 10.4.4. Calibration

Passive measurements at an OP could be calibrated against an active measurement (with loss emulation) at host A or B, where the active measurement represents the ground-truth.

### 10.5. Administrative items

#### 10.5.1. Status

Current

#### 10.5.2. Requester

This RFC number

## 10.5.3. Revision

1.0

## 10.5.4. Revision Date

YYYY-MM-DD

## 10.6. Comments and Remarks

None.

## 11. Security Considerations

These registry entries represent no known implications for Internet Security. Each RFC referenced above contains a Security Considerations section. Further, the LMAP Framework [RFC7594] provides both security and privacy considerations for measurements.

There are potential privacy considerations for observed traffic, particularly for passive metrics in section 10. An attacker that knows that its TCP connection is being measured can modify its behavior to skew the measurement results.

## 12. IANA Considerations

IANA is requested to populate The Performance Metrics Registry defined in [I-D.ietf-ippm-metric-registry] with the values defined in sections 4 through 10.

See the IANA Considerations section of [I-D.ietf-ippm-metric-registry] for additional requests and considerations.

## 13. Acknowledgements

The authors thank Brian Trammell for suggesting the term "Run-time Parameters", which led to the distinction between run-time and fixed parameters implemented in this memo, for identifying the IPFIX metric with Flow Key as an example, for suggesting the Passive TCP RTD metric and supporting references, and for many other productive suggestions. Thanks to Peter Koch, who provided several useful suggestions for disambiguating successive DNS Queries in the DNS Response time metric.

The authors also acknowledge the constructive reviews and helpful suggestions from Barbara Stark, Juergen Schoenwaelder, Tim Carey, Yaakov Stein, and participants in the LMAP working group. Thanks to

Michelle Cotton for her early IANA reviews, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL.

## 14. References

### 14.1. Normative References

- [I-D.ietf-ippm-metric-registry]  
Bagnulo, M., Claise, B., Eardley, P., and A. Morton,  
"Registry for Performance Metrics", Internet Draft (work  
in progress) draft-ietf-ippm-metric-registry, 2019.
- [RFC1035] Mockapetris, P., "Domain names - implementation and  
specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,  
November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [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>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:  
Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,  
<<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation  
Metric for IP Performance Metrics (IPPM)", RFC 3393,  
DOI 10.17487/RFC3393, November 2002,  
<<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network  
performance measurement with periodic streams", RFC 3432,  
DOI 10.17487/RFC3432, November 2002,  
<<https://www.rfc-editor.org/info/rfc3432>>.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<https://www.rfc-editor.org/info/rfc4737>>.
- [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>>.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [RFC5560] Uijterwaal, H., "A One-Way Packet Duplication Metric", RFC 5560, DOI 10.17487/RFC5560, May 2009, <<https://www.rfc-editor.org/info/rfc5560>>.
- [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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6049] Morton, A. and E. Stephan, "Spatial Composition of Metrics", RFC 6049, DOI 10.17487/RFC6049, January 2011, <<https://www.rfc-editor.org/info/rfc6049>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

- [RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <<https://www.rfc-editor.org/info/rfc7323>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [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>>.
- [Strowes] Strowes, S., "Passively Measuring TCP Round Trip Times, Communications of the ACM, Vol. 56 No. 10, Pages 57-64", September 2013.
- [Trammell-14]  
Trammell, B., "Inline Data Integrity Signals for Passive Measurement, In: Dainotti A., Mahanti A., Uhlig S. (eds) Traffic Monitoring and Analysis. TMA 2014. Lecture Notes in Computer Science, vol 8406. Springer, Berlin, Heidelberg [https://link.springer.com/chapter/10.1007/978-3-642-54999-1\\_2](https://link.springer.com/chapter/10.1007/978-3-642-54999-1_2)", March 2014.

#### 14.2. Informative References

- [RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <<https://www.rfc-editor.org/info/rfc1242>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [RFC6703] Morton, A., Ramachandran, G., and G. Maguluri, "Reporting IP Network Performance Metrics: Different Points of View", RFC 6703, DOI 10.17487/RFC6703, August 2012, <<https://www.rfc-editor.org/info/rfc6703>>.



[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>>.

## 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: [acmorton@att.com](mailto:acmorton@att.com)

Marcelo Bagnulo  
Universidad Carlos III de  
Madrid  
Av. Universidad 30  
Leganes, Madrid 28911  
SPAIN

Phone: 34 91 6249500  
Email: [marcelo@it.uc3m.es](mailto:marcelo@it.uc3m.es)  
URI: <http://www.it.uc3m.es>

Philip Eardley  
BT  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: [philip.eardley@bt.com](mailto:philip.eardley@bt.com)

Kevin D'Souza  
AT&T Labs  
200 Laurel Avenue South  
Middletown,, NJ 07748  
USA

Phone: +1 732 420 xxxx  
Email: [kld@att.com](mailto:kld@att.com)

ippm  
Internet-Draft  
Intended status: Standards Track  
Expires: June 16, 2022

F. Brockners, Ed.  
Cisco  
S. Bhandari, Ed.  
Thoughtspot  
T. Mizrahi, Ed.  
Huawei  
December 13, 2021

Data Fields for In-situ OAM  
draft-ietf-ippm-ioam-data-17

## Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path in the network. This document discusses the data fields and associated data types for in-situ OAM. In-situ OAM data fields can be encapsulated into a variety of protocols such as NSH, Segment Routing, Geneve, or IPv6. In-situ OAM can be used to complement OAM mechanisms based on, e.g., ICMP or other types of probe 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 June 16, 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Contributors . . . . .	3
3. Conventions . . . . .	4
4. Scope, Applicability, and Assumptions . . . . .	5
5. IOAM Data-Fields, Types, Nodes . . . . .	6
5.1. IOAM Data-Fields and Option-Types . . . . .	7
5.2. IOAM-Domains and types of IOAM Nodes . . . . .	7
5.3. IOAM-Namespaces . . . . .	8
5.4. IOAM Trace Option-Types . . . . .	11
5.4.1. Pre-allocated and Incremental Trace Option-Types . .	13
5.4.2. IOAM node data fields and associated formats . . . .	17
5.4.2.1. Hop_Lim and node_id short format . . . . .	18
5.4.2.2. ingress_if_id and egress_if_id . . . . .	19
5.4.2.3. timestamp seconds . . . . .	19
5.4.2.4. timestamp fraction . . . . .	20
5.4.2.5. transit delay . . . . .	20
5.4.2.6. namespace specific data . . . . .	20
5.4.2.7. queue depth . . . . .	21
5.4.2.8. Checksum Complement . . . . .	21
5.4.2.9. Hop_Lim and node_id wide . . . . .	22
5.4.2.10. ingress_if_id and egress_if_id wide . . . . .	22
5.4.2.11. namespace specific data wide . . . . .	22
5.4.2.12. buffer occupancy . . . . .	23
5.4.2.13. Opaque State Snapshot . . . . .	23
5.4.3. Examples of IOAM node data . . . . .	24
5.5. IOAM Proof of Transit Option-Type . . . . .	26
5.5.1. IOAM Proof of Transit Type 0 . . . . .	28
5.6. IOAM Edge-to-Edge Option-Type . . . . .	28
6. Timestamp Formats . . . . .	31
6.1. PTP Truncated Timestamp Format . . . . .	31
6.2. NTP 64-bit Timestamp Format . . . . .	32
6.3. POSIX-based Timestamp Format . . . . .	33
7. IOAM Data Export . . . . .	34
8. IANA Considerations . . . . .	35
8.1. IOAM Option-Type Registry . . . . .	35
8.2. IOAM Trace-Type Registry . . . . .	36
8.3. IOAM Trace-Flags Registry . . . . .	37
8.4. IOAM POT-Type Registry . . . . .	37
8.5. IOAM POT-Flags Registry . . . . .	38

8.6. IOAM E2E-Type Registry . . . . .	38
8.7. IOAM Namespace-ID Registry . . . . .	39
9. Management and Deployment Considerations . . . . .	40
10. Security Considerations . . . . .	40
11. Acknowledgements . . . . .	43
12. References . . . . .	43
12.1. Normative References . . . . .	43
12.2. Informative References . . . . .	44
Contributors' Addresses . . . . .	45
Authors' Addresses . . . . .	47

## 1. Introduction

This document defines data fields for "in-situ" Operations, Administration, and Maintenance (IOAM). In-situ OAM records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than being sent within packets specifically dedicated to OAM. IOAM is to complement mechanisms such as Ping or Traceroute. In terms of "active" or "passive" OAM, "in-situ" OAM can be considered a hybrid OAM type. "In-situ" mechanisms do not require extra packets to be sent. IOAM adds information to the already available data packets and therefore cannot be considered passive. In terms of the classification given in [RFC7799], IOAM could be portrayed as Hybrid Type I. IOAM mechanisms can be leveraged where mechanisms using, e.g., ICMP do not apply or do not offer the desired results, such as proving that a certain traffic flow takes a pre-defined path, SLA verification for the data traffic, detailed statistics on traffic distribution paths in networks that distribute traffic across multiple paths, or scenarios in which probe traffic is potentially handled differently from regular data traffic by the network devices.

The term "in situ OAM" was originally motivated by the use of OAM related mechanisms that add information into a packet. This document uses IOAM as a term defining the IOAM technology. IOAM includes "in-situ" mechanisms, but also mechanisms that could trigger the creation of additional packets dedicated to OAM.

## 2. Contributors

This document was the collective effort of several authors. The text and content were contributed by the editors and the co-authors listed below. The contact information of the co-authors appears at the end of this document.

- o Carlos Pignataro

- o Mickey Spiegel
- o Barak Gafni
- o Jennifer Lemon
- o Hannes Gredler
- o John Leddy
- o Stephen Youell
- o David Mozes
- o Petr Lapukhov
- o Remy Chang
- o Daniel Bernier

### 3. Conventions

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.

Abbreviations and definitions used in this document:

E2E:            Edge to Edge

Geneve:        Generic Network Virtualization Encapsulation [RFC8926]

IOAM:          In-situ Operations, Administration, and Maintenance

MTU:           Maximum Transmit Unit

NSH:           Network Service Header [RFC8300]

OAM:           Operations, Administration, and Maintenance

PMTU:          Path MTU

POT:           Proof of Transit

Short format: "Short format" refers to an IOAM-Data-Field which comprises 4 octets.

SID: Segment Identifier

SR: Segment Routing

VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension [I-D.ietf-nvo3-vxlan-gpe]

Wide format: "Wide format" refers to an IOAM-Data-Field which comprises 8 octets.

#### 4. Scope, Applicability, and Assumptions

IOAM assumes a set of constraints as well as guiding principles and concepts that go hand in hand with the definition of the IOAM data fields. These constraints, guiding principles, and concepts are described in this section. A discussion of how IOAM data fields and the associated concepts are applied to an IOAM deployment are out of scope for this document. Please refer to [I-D.ietf-ippm-ioam-deployment] for IOAM deployment considerations.

Scope: This document defines the data fields and associated data types for in-situ OAM. The in-situ OAM data fields can be encapsulated in a variety of protocols, including NSH, Segment Routing, Geneve, and IPv6. Specification details for these different protocols are outside the scope of this document. It is expected that each such encapsulation would be specified by an RFC, jointly designed by the working group that develops or maintains the encapsulation protocol and the IETF IPPM working group.

Deployment domain (or scope) of in-situ OAM deployment: IOAM is focused on "limited domains" as defined in [RFC8799]. For IOAM, a limited domain could for example be an enterprise campus using physical connections between devices or an overlay network using virtual connections / tunnels for connectivity between said devices. A limited domain which uses IOAM may constitute one or multiple "IOAM-domains", each disambiguated through separate namespace identifiers. An IOAM-domain is bounded by its perimeter or edge. IOAM-domains may overlap inside the limited domain. Designers of protocol encapsulations for IOAM specify mechanisms to ensure that IOAM data stays within an IOAM-domain. In addition, the operator of such a domain is expected to put provisions in place to ensure that IOAM data does not leak beyond the edge of an IOAM-domain using, for example, packet filtering methods. The operator SHOULD consider the potential operational impact of IOAM to mechanisms such as ECMP processing (e.g., load-balancing schemes based on packet length could be impacted by the increased packet size due to IOAM), path MTU (i.e., ensure that the MTU of all links within a domain is sufficiently large to support the increased packet size due to IOAM)

and ICMP message handling (i.e., in case of IPv6, IOAM support for ICMPv6 Echo Request/Reply is desired which would translate into ICMPv6 extensions to enable IOAM-Data-Fields to be copied from an Echo Request message to an Echo Reply message).

**IOAM control points:** IOAM-Data-Fields are added to or removed from the user traffic by the devices which form the edge of a domain. Devices which form an IOAM-Domain can add, update or remove IOAM-Data-Fields. Edge devices of an IOAM-Domain can be hosts or network devices.

**Traffic-sets that IOAM is applied to:** IOAM can be deployed on all or only on subsets of the user traffic. Using IOAM on a selected set of traffic (e.g., per interface, based on an access control list or flow specification defining a specific set of traffic, etc.) could be useful in deployments where the cost of processing IOAM-Data-Fields by encapsulating, transit, or decapsulating node(s) might be a concern from a performance or operational perspective. Thus limiting the amount of traffic IOAM is applied to could be beneficial in some deployments.

**Encapsulation independence:** The definition of IOAM-Data-Fields is independent from the protocols the IOAM-Data-Fields are encapsulated into. IOAM-Data-Fields can be encapsulated into several encapsulating protocols.

**Layering:** If several encapsulation protocols (e.g., in case of tunneling) are stacked on top of each other, IOAM-Data-Fields could be present at multiple layers. The behavior follows the ships-in-the-night model, i.e., IOAM-Data-Fields in one layer are independent from IOAM-Data-Fields in another layer. Layering allows operators to instrument the protocol layer they want to measure. The different layers could, but do not have to, share the same IOAM encapsulation mechanisms.

**IOAM implementation:** The definition of the IOAM-Data-Fields take the specifics of devices with hardware data planes and software data planes into account.

## 5. IOAM Data-Fields, Types, Nodes

This section details IOAM-related nomenclature and describes data types such as IOAM-Data-Fields, IOAM-Types, IOAM-Namespaces as well as the different types of IOAM nodes.

### 5.1. IOAM Data-Fields and Option-Types

An IOAM-Data-Field is a set of bits with a defined format and meaning, which can be stored at a certain place in a packet for the purpose of IOAM.

To accommodate the different uses of IOAM, IOAM-Data-Fields fall into different categories. In IOAM, these categories are referred to as IOAM-Option-Types. A common registry is maintained for IOAM-Option-Types, see Section 8.1 for details. Corresponding to these IOAM-Option-Types, different IOAM-Data-Fields are defined.

This document defines four IOAM-Option-Types:

- o Pre-allocated Trace Option-Type
- o Incremental Trace Option-Type
- o Proof of Transit (POT) Option-Type
- o Edge-to-Edge (E2E) Option-Type

Future IOAM-Option-Types can be allocated by IANA, as described in Section 8.1.

### 5.2. IOAM-Domains and types of IOAM Nodes

Section 4 already mentioned that IOAM is expected to be deployed in a limited domain [RFC8799]. One or more IOAM-Option-Types are added to a packet upon entering an IOAM-Domain and are removed from the packet when exiting the domain. Within the IOAM-Domain, the IOAM-Data-Fields MAY be updated by network nodes that the packet traverses. An IOAM-Domain consists of "IOAM encapsulating nodes", "IOAM decapsulating nodes" and "IOAM transit nodes". The role of a node (i.e., encapsulating, transit, decapsulating) is defined within an IOAM-Namespace (see below). A node can have different roles in different IOAM-Namespace.

A device which adds at least one IOAM-Option-Type to the packet is called an "IOAM encapsulating node", whereas a device which removes an IOAM-Option-Type is referred to as an "IOAM decapsulating node". Nodes within the domain which are aware of IOAM data and read and/or write and/or process IOAM data are called "IOAM transit nodes". IOAM nodes which add or remove the IOAM-Data-Fields can also update the IOAM-Data-Fields at the same time. Or in other words, IOAM encapsulating or decapsulating nodes can also serve as IOAM transit nodes at the same time. Note that not every node in an IOAM-domain needs to be an IOAM transit node. For example, a deployment might



require that packets traverse a set of firewalls which support IOAM. In that case, only the set of firewall nodes would be IOAM transit nodes rather than all nodes.

An "IOAM encapsulating node" incorporates one or more IOAM-Option-Types (from the list of IOAM-Types, see Section 8.1) into packets that IOAM is enabled for. If IOAM is enabled for a selected subset of the traffic, the IOAM encapsulating node is responsible for applying the IOAM functionality to the selected subset.

An "IOAM transit node" reads and/or writes and/or processes one or more of the IOAM-Data-Fields. If both the Pre-allocated and the Incremental Trace Option-Types are present in the packet, each IOAM transit node based on configuration and available implementation of IOAM might populate IOAM trace data in either Pre-allocated or Incremental Trace Option-Type but not both. Note that not populating any of the Trace Option-Types is also valid behavior for an IOAM transit node. A transit node MUST ignore IOAM-Option-Types that it does not understand. A transit node MUST NOT add new IOAM-Option-Types to a packet, MUST NOT remove IOAM-Option-Types from a packet, and MUST NOT change the IOAM-Data-Fields of an IOAM Edge-to-Edge Option-Type.

An "IOAM decapsulating node" removes IOAM-Option-Type(s) from packets.

The role of an IOAM-encapsulating, IOAM-transit or IOAM-decapsulating node is always performed within a specific IOAM-Namespace. This means that an IOAM node which is, e.g., an IOAM-decapsulating node for IOAM-Namespace "A" but not for IOAM-Namespace "B" will only remove the IOAM-Option-Types for IOAM-Namespace "A" from the packet. Note that this applies even for IOAM-Option-Types that the node does not understand, for example an IOAM-Option-Type other than the four described above, that is added in a future revision.

IOAM-Namespaces allow for a namespace-specific definition and interpretation of IOAM-Data-Fields. An interface-id could for example point to a physical interface (e.g., to understand which physical interface of an aggregated link is used when receiving or transmitting a packet) whereas in another case it could refer to a logical interface (e.g., in case of tunnels). Please refer to Section 5.3 for details on IOAM-Namespaces.

### 5.3. IOAM-Namespaces

IOAM-Namespaces add further context to IOAM-Option-Types and associated IOAM-Data-Fields. The IOAM-Option-Types and associated IOAM-Data-Fields are interpreted as defined in this document,

regardless of the value of the IOAM-Namespace. However, IOAM-Namespaces provide a way to group nodes to support different deployment approaches of IOAM (see a few example use-cases below). IOAM-Namespaces also help to resolve potential issues which can occur due to IOAM-Data-Fields not being globally unique (e.g., IOAM node identifiers do not have to be globally unique). IOAM-Data-Fields significance is always within a particular IOAM-Namespace. Given that IOAM-Data-Fields are always interpreted the context of a specific namespace, the namespace-id field always needs to be carried along with the IOAM data-fields themselves.

An IOAM-Namespace is identified by a 16-bit namespace identifier (Namespace-ID). The IOAM-Namespace field is included in all the IOAM-Option-Types defined in this document, and MUST be included in all future IOAM-Option-Types. The Namespace-ID value is divided into two sub-ranges:

- o An operator-assigned range from 0x0001 to 0x7FFF
- o An IANA-assigned range from 0x8000 to 0xFFFF

The IANA-assigned range is intended to allow future extensions to have new and interoperable IOAM functionality, while the operator-assigned range is intended to be domain-specific, and managed by the network operator. The Namespace-ID value of 0x0000 is the "Default-Namespace-ID". The Default-Namespace-ID indicates that no specific namespace is associated with the IOAM data fields in the packet. The Default-Namespace-ID MUST be supported by all nodes implementing IOAM. A use-case for the Default-Namespace-ID are deployments which do not leverage specific namespaces for some or all of their packets that carry IOAM data fields.

Namespace identifiers allow devices which are IOAM capable to determine:

- o whether IOAM-Option-Type(s) need to be processed by a device: If the Namespace-ID contained in a packet does not match any Namespace-ID the node is configured to operate on, then the node MUST NOT change the contents of the IOAM-Data-Fields.
- o which IOAM-Option-Type needs to be processed/updated in case there are multiple IOAM-Option-Types present in the packet. Multiple IOAM-Option-Types can be present in a packet in case of overlapping IOAM-Domains or in case of a layered IOAM deployment.
- o whether IOAM-Option-Type(s) have to be removed from the packet, e.g., at a domain edge or domain boundary.

IOAM-Namespaces support several different uses:

- o IOAM-Namespaces can be used by an operator to distinguish different IOAM-domains. Devices at edges of an IOAM-domain can filter on Namespace-IDs to provide for proper IOAM-domain isolation.
- o IOAM-Namespaces provide additional context for IOAM-Data-Fields and thus can be used to ensure that IOAM-Data-Fields are unique and are interpreted properly by management stations or network controllers. The node identifier field (`node_id`, see below) does not need to be unique in a deployment. This could be the case if an operator wishes to use different node identifiers for different IOAM layers, even within the same device or node identifiers might not be unique for other organizational reasons, such as after a merger of two formerly separated organizations. The Namespace-ID can be used as a context identifier, such that the combination of `node_id` and Namespace-ID will always be unique.
- o Similarly, IOAM-Namespaces can be used to define how certain IOAM-Data-Fields are interpreted: IOAM offers three different timestamp format options. The Namespace-ID can be used to determine the timestamp format. IOAM-Data-Fields (e.g., buffer occupancy) which do not have a unit associated are to be interpreted within the context of a IOAM-Namespace.
- o IOAM-Namespaces can be used to identify different sets of devices (e.g., different types of devices) in a deployment: If an operator desires to insert different IOAM-Data-Fields based on the device, the devices could be grouped into multiple IOAM-Namespaces. This could be due to the fact that the IOAM feature set differs between different sets of devices, or it could be for reasons of optimized space usage in the packet header. It could also stem from hardware or operational limitations on the size of the trace data that can be added and processed, preventing collection of a full trace for a flow.
- o By assigning different IOAM Namespace-IDs to different sets of nodes or network partitions and using a separate instance of an IOAM-Option-Type for each Namespace-ID, a full trace for a flow could be collected and constructed via partial traces from each IOAM-Option-Type in each of the packets in the flow. Example: An operator could choose to group the devices of a domain into two IOAM-Namespaces, in a way that each IOAM-Namespace is represented by one of two IOAM-Option-Types in the packet. Each node would record data only for the IOAM-Namespace that it belongs to, ignoring the other IOAM-Option-Type with a IOAM-Namespace to which it doesn't belong. To retrieve a full view of the deployment, the

captured IOAM-Data-Fields of the two IOAM-Namespaces need to be correlated.

#### 5.4. IOAM Trace Option-Types

In a typical deployment, all nodes in an IOAM-Domain would participate in IOAM and thus be IOAM transit nodes, IOAM encapsulating or IOAM decapsulating nodes. If not all nodes within a domain support IOAM functionality as defined in this document, IOAM tracing information (i.e., node data, see below) can only be collected on those nodes which support IOAM functionality as defined in this document. Nodes which do not support IOAM functionality as defined in this document will forward the packet without any changes to the IOAM-Data-Fields. The maximum number of hops and the minimum path MTU of the IOAM-domain is assumed to be known. An overflow indicator (O-bit) is defined as one of the ways to deal with situations where the PMTU was underestimated, i.e., where the number of hops which are IOAM capable exceeds the available space in the packet.

To optimize hardware and software implementations, IOAM tracing is defined as two separate options. A deployment can choose to configure and support one or both of the following options.

**Pre-allocated Trace-Option:** This trace option is defined as a container of node data fields (see below) with pre-allocated space for each node to populate its information. This option is useful for implementations where it is efficient to allocate the space once and index into the array to populate the data during transit (e.g., software forwarders often fall into this class). The IOAM encapsulating node allocates space for Pre-allocated Trace Option-Type in the packet and sets corresponding fields in this IOAM-Option-Type. The IOAM encapsulating node allocates an array which is used to store operational data retrieved from every node while the packet traverses the domain. IOAM transit nodes update the content of the array, and possibly update the checksums of outer headers. A pointer which is part of the IOAM trace data, points to the next empty slot in the array. An IOAM transit node that updates the content of the pre-allocated option also updates the value of the pointer, which specifies where the next IOAM transit node fills in its data. The "node data list" array (see below) in the packet is populated iteratively as the packet traverses the network, starting with the last entry of the array, i.e., "node data list [n]" is the first entry to be populated, "node data list [n-1]" is the second one, etc.

**Incremental Trace-Option:** This trace option is defined as a container of node data fields where each node allocates and pushes

its node data immediately following the option header. This type of trace recording is useful for some of the hardware implementations as it eliminates the need for the transit network elements to read the full array in the option and allows for arbitrarily long packets as the MTU allows. The IOAM encapsulating node allocates space for the Incremental Trace Option-Type. Based on operational state and configuration, the IOAM encapsulating node sets the fields in the Option-Type that control what IOAM-Data-Fields have to be collected and how large the node data list can grow. IOAM transit nodes push their node data to the node data list subject to any protocol constraints of the encapsulating layer. They then decrease the remaining length available to subsequent nodes and adjust the lengths and possibly checksums in outer headers.

IOAM encapsulating nodes and IOAM decapsulating nodes which support tracing MUST support both Trace-Option-Types. For IOAM transit nodes it is sufficient to support one of the Trace-Option-Types. In the event that both options are utilized in a deployment at the same time, the Incremental Trace-Option MUST be placed before the Pre-allocated Trace-Option. Deployments which mix devices with either the Incremental Trace-Option or the Pre-allocated Trace-Option could result in both Option-Types being present in a packet. Given that the operator knows which equipment is deployed in a particular IOAM-domain, the operator will decide by means of configuration which type(s) of trace options will be used for a particular domain.

Every node data entry holds information for a particular IOAM transit node that is traversed by a packet. The IOAM decapsulating node removes the IOAM-Option-Type(s) and processes and/or exports the associated data. Like all IOAM-Data-Fields, the IOAM-Data-Fields of the IOAM-Trace-Option-Types are defined in the context of an IOAM-Namespace.

IOAM tracing can collect the following types of information:

- o Identification of the IOAM node. An IOAM node identifier can match to a device identifier or a particular control point or subsystem within a device.
- o Identification of the interface that a packet was received on, i.e., ingress interface.
- o Identification of the interface that a packet was sent out on, i.e., egress interface.
- o Time of day when the packet was processed by the node as well as the transit delay. Different definitions of processing time are

feasible and expected, though it is important that all devices of an IOAM-domain follow the same definition.

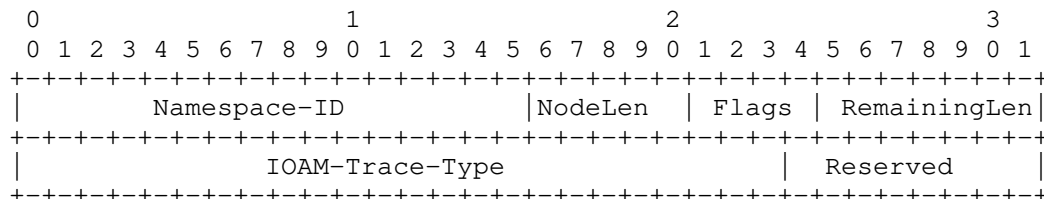
- o Generic data: Format-free information where syntax and semantic of the information is defined by the operator in a specific deployment. For a specific IOAM-Namespace, all IOAM nodes have to interpret the generic data the same way. Examples for generic IOAM data include geo-location information (location of the node at the time the packet was processed), buffer queue fill level or cache fill level at the time the packet was processed, or even a battery charge level.
- o Information to detect whether IOAM trace data was added at every hop or whether certain hops in the domain weren't IOAM transit nodes.

It should be noted that the semantics of some of the node data fields that are defined below, such as the queue depth and buffer occupancy, are implementation specific. This approach is intended to allow IOAM nodes with various different architectures.

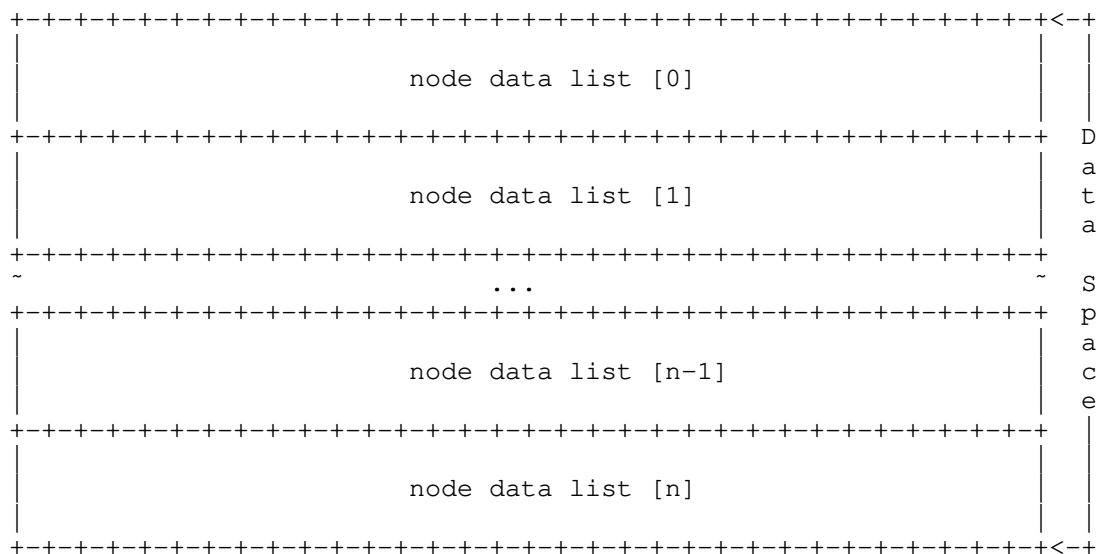
#### 5.4.1. Pre-allocated and Incremental Trace Option-Types

The IOAM Pre-allocated Trace-Option and the IOAM Incremental Trace-Option have similar formats. Except where noted below, the internal formats and fields of the two trace options are identical. Both Trace-Options consist of a fixed size "trace option header" and a variable data space to store gathered data, the "node data list". An IOAM transit node (that is not an IOAM encapsulating node or IOAM decapsulating node) MUST NOT modify any of the fields in the fixed size "trace option header", other than "flags" and "RemainingLen", i.e., an IOAM transit node MUST NOT modify the Namespace-ID, NodeLen, IOAM-Trace-Type, or Reserved fields.

Pre-allocated and incremental trace option headers:



The trace option data MUST be 4-octet aligned:



**Namespace-ID:** 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, the node MUST NOT change the contents of the IOAM-Data-Fields.

**NodeLen:** 5-bit unsigned integer. This field specifies the length of data added by each node in multiples of 4-octets, excluding the length of the "Opaque State Snapshot" field.

If IOAM-Trace-Type bit 22 is not set, then NodeLen specifies the actual length added by each node. If IOAM-Trace-Type bit 22 is

set, then the actual length added by a node would be (NodeLen + length of the "Opaque State Snapshot" field) in 4 octet units.

For example, if 3 IOAM-Trace-Type bits are set and none of them are in wide format, then NodeLen would be 3. If 3 IOAM-Trace-Type bits are set and 2 of them are wide, then NodeLen would be 5.

An IOAM encapsulating node MUST set NodeLen.

A node receiving an IOAM Pre-allocated or Incremental Trace-Option relies on the NodeLen value.

Flags 4-bit field. Flags are allocated by IANA, as specified in Section 8.3. This document allocates a single flag as follows:

Bit 0 "Overflow" (O-bit) (most significant bit). In case a network element is supposed to add node data to a packet, but detects that there are not enough octets left to record the node data, the network element MUST NOT add any fields and MUST set the overflow "O-bit" to "1" in the IOAM-Trace-Option header. This is useful for transit nodes to ignore further processing of the option.

RemainingLen: 7-bit unsigned integer. This field specifies the data space in multiples of 4-octets remaining for recording the node data, before the node data list is considered to have overflowed. The sender MUST assign the initial value of the RemainingLen field. The sender MAY calculate the value of the RemainingLen field by computing the number of node data bytes allowed before exceeding the path MTU (PMTU), given that the PMTU is known to the sender. Subsequent nodes can carry out a simple comparison between RemainingLen and NodeLen, along with the length of the "Opaque State Snapshot" if applicable, to determine whether or not data can be added by this node. When node data is added, the node MUST decrease RemainingLen by the amount of data added. In the pre-allocated trace option, RemainingLen is used to derive the offset in data space to record the node data element. Specifically, the recording of the node data element would start from RemainingLen - NodeLen - sizeof(opaque snapshot) in 4 octet units. If RemainingLen in a pre-allocated trace option exceeds the length of the option, as specified in the lower layer header (which is not within the scope of this document), then the node MUST NOT add any fields.

IOAM-Trace-Type: A 24-bit identifier which specifies which data types are used in this node data list.



The IOAM-Trace-Type value is a bit field. The following bits are defined in this document, with details on each bit described in the Section 5.4.2. The order of packing the data fields in each node data element follows the bit order of the IOAM-Trace-Type field, as follows:

- Bit 0      (Most significant bit) When set, indicates presence of Hop\_Lim and node\_id (short format) in the node data.
- Bit 1      When set, indicates presence of ingress\_if\_id and egress\_if\_id (short format) in the node data.
- Bit 2      When set, indicates presence of timestamp seconds in the node data.
- Bit 3      When set, indicates presence of timestamp fraction in the node data.
- Bit 4      When set, indicates presence of transit delay in the node data.
- Bit 5      When set, indicates presence of IOAM-Namespace specific data (short format) in the node data.
- Bit 6      When set, indicates presence of queue depth in the node data.
- Bit 7      When set, indicates presence of the Checksum Complement node data.
- Bit 8      When set, indicates presence of Hop\_Lim and node\_id in wide format in the node data.
- Bit 9      When set, indicates presence of ingress\_if\_id and egress\_if\_id in wide format in the node data.
- Bit 10     When set, indicates presence of IOAM-Namespace specific data in wide format in the node data.
- Bit 11     When set, indicates presence of buffer occupancy in the node data.
- Bit 12-21 Undefined. These values are available for future assignment in the IOAM Trace-Type Registry (Section 8.2). Every future node data field corresponding to one of these bits MUST be 4-octets long. An IOAM encapsulating node MUST set the value of each undefined bit to 0. If

an IOAM transit node receives a packet with one or more of these bits set to 1, it MUST either:

1. Add corresponding node data filled with the reserved value 0xFFFFFFFF, after the node data fields for the IOAM-Trace-Type bits defined above, such that the total node data added by this node in units of 4-octets is equal to NodeLen, or
2. Not add any node data fields to the packet, even for the IOAM-Trace-Type bits defined above.

Bit 22    When set, indicates presence of variable length Opaque State Snapshot field.

Bit 23    Reserved: MUST be set to zero upon transmission and ignored upon receipt. This bit is reserved to allow for future extensions of the IOAM-Trace-Type bit field.

Section 5.4.2 describes the IOAM-Data-Types and their formats. Within an IOAM-Domain possible combinations of these bits making the IOAM-Trace-Type can be restricted by configuration knobs.

Reserved: 8-bits. An IOAM encapsulating node MUST set the value to zero upon transmission. IOAM transit nodes MUST ignore the received value.

Node data List [n]: Variable-length field. This is a list of node data elements where the content of each node data element is 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. Each node MUST prepend its node data element in front of the node data elements that it received, such that the transmitted node data list begins with this node's data element as the first populated element in the list. The last node data element in this list is the node data of the first IOAM capable node in the path. Populating the node data list in this way ensures that the order of node data list is the same for incremental and pre-allocated trace options. In the pre-allocated trace option, the index contained in RemainingLen identifies the offset for current active node data to be populated.

#### 5.4.2. IOAM node data fields and associated formats

All the IOAM-Data-Fields MUST be 4-octet aligned. If a node which is supposed to update an IOAM-Data-Field is not capable of populating the value of a field set in the IOAM-Trace-Type, the field value MUST be set to 0xFFFFFFFF for 4-octet fields or 0xFFFFFFFFFFFFFFFF for

8-octet fields, indicating that the value is not populated, except when explicitly specified in the field description below.

Some IOAM-Data-Fields defined below, such as interface identifiers or IOAM-Namespace specific data, are defined in both "short format" as well as "wide format". The use of "short format" or "wide format" is not mutually exclusive. A deployment could choose to leverage both. For example, `ingress_if_id`(short format) could be an identifier for the physical interface, whereas `ingress_if_id`(wide format) could be an identifier for a logical sub-interface of that physical interface.

Data fields and associated data types for each of the IOAM-Data-Fields are specified in the following sections. The definition of IOAM-Data-Fields focuses on the syntax of the data-fields and avoids specifying the semantics where feasible. This is why no units are defined for data-fields like e.g., "buffer occupancy" or "queue depth". With this approach, nodes can supply the information in their native format and are not required to perform unit or format conversions. Systems that further process IOAM information, like e.g., a network management system are assumed to also handle unit conversions as part of their IOAM data-fields processing. The combination of a particular data-field and the namespace-id provides for the context to interpret the provided data appropriately.

#### 5.4.2.1. Hop\_Lim and node\_id short format

The "Hop\_Lim and node\_id short format" field is a 4-octet field that is defined as follows:

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Hop_Lim | node_id |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

**Hop\_Lim:** 1-octet unsigned integer. It is set to the Hop Limit value in the packet at egress from the node that records this data. Hop Limit information is used to identify the location of the node in the communication path. This is copied from the lower layer, e.g., TTL value in IPv4 header or hop limit field from IPv6 header of the packet when the packet is ready for transmission. The semantics of the Hop\_Lim field depend on the lower layer protocol that IOAM is encapsulated into, and therefore its specific semantics are outside the scope of this memo. The value of this field MUST be set to 0xff when the lower level does not have a TTL/Hop limit equivalent field.

**node\_id:** 3-octet unsigned integer. Node identifier field to uniquely identify a node within the IOAM-Namespace and associated

IOAM-Domain. The procedure to allocate, manage and map the `node_ids` is beyond the scope of this document. See [I-D.ietf-ippm-ioam-deployment] for a discussion of deployment related aspects of the `node_id`.

#### 5.4.2.2. `ingress_if_id` and `egress_if_id`

The "`ingress_if_id` and `egress_if_id`" field is a 4-octet field that is defined as follows:

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           ingress_if_id           |           egress_if_id           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

`ingress_if_id`: 2-octet unsigned integer. Interface identifier to record the ingress interface the packet was received on.

`egress_if_id`: 2-octet unsigned integer. Interface identifier to record the egress interface the packet is forwarded out of.

Note that due to the fact that IOAM uses its own IOAM-Namespaces for IOAM-Data-Fields, data fields like interface identifiers can be used in a flexible way to represent system resources that are associated with ingressing or egressing packets, i.e., `ingress_if_id` could represent a physical interface, a virtual or logical interface, or even a queue.

#### 5.4.2.3. `timestamp seconds`

The "`timestamp seconds`" field is a 4-octet unsigned integer field. It contains the absolute timestamp in seconds that specifies the time at which the packet was received by the node. This field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp Seconds field contains the 32 most significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value that is valid for 1 second in approximately 136 years; the analyzer has to correlate several packets or compare the timestamp value to its own time-of-day in order to detect the error indication.

## 5.4.2.4. timestamp fraction

The "timestamp fraction" field is a 4-octet unsigned integer field. Fraction specifies the fractional portion of the number of seconds since the NTP epoch [RFC8877]. The field specifies the time at which the packet was received by the node. This field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp fraction field contains the 32 least significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value in the NTP format, valid for approximately 233 picoseconds in every second. If the NTP format is used the analyzer has to correlate several packets in order to detect the error indication.

## 5.4.2.5. transit delay

The "transit delay" field is a 4-octet unsigned integer in the range 0 to  $2^{31}-1$ . It is the time in nanoseconds the packet spent in the transit node. This can serve as an indication of the queuing delay at the node. If the transit delay exceeds  $2^{31}-1$  nanoseconds then the top bit 'O' is set to indicate overflow and value set to 0x80000000. When this field is part of the data field but a node populating the field is not able to fill it, the field position in the field MUST be filled with value 0xFFFFFFFF to mean not populated.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|O|                                     transit delay                      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

## 5.4.2.6. namespace specific data

The "namespace specific data" field is a 4-octet field which can be used by the node to add IOAM-Namespace specific data. This represents a "free-format" 4-octet bit field with its semantics defined in the context of a specific IOAM-Namespace.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     namespace specific data                      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

## 5.4.2.7. queue depth

The "queue depth" field is a 4-octet unsigned integer field. This field indicates the current length of the egress interface queue of the interface from where the packet is forwarded out. The queue depth is expressed as the current amount of memory buffers used by the queue (a packet could consume one or more memory buffers, depending on its size).

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     queue depth                             |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## 5.4.2.8. Checksum Complement

The "Checksum Complement" field is a 4-octet node data which contains a 4-octet Checksum Complement field. The Checksum Complement is useful when IOAM is transported over encapsulations that make use of a UDP transport, such as VXLAN-GPE or Geneve. Without the Checksum Complement, nodes adding IOAM node data update the UDP Checksum field following the recommendation of the encapsulation protocols. When the Checksum Complement is present, an IOAM encapsulating node or IOAM transit node adding node data MUST carry out one of the following two alternatives in order to maintain the correctness of the UDP Checksum value:

1. Recompute the UDP Checksum field.
2. Use the Checksum Complement to make a checksum-neutral update in the UDP payload; the Checksum Complement is assigned a value that complements the rest of the node data fields that were added by the current node, causing the existing UDP Checksum field to remain correct.

IOAM decapsulating nodes MUST recompute the UDP Checksum field, since they do not know whether previous hops modified the UDP Checksum field or the Checksum Complement field.

Checksum Complement fields are used in a similar manner in [RFC7820] and [RFC7821].

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Checksum Complement                             |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## 5.4.2.9. Hop\_Lim and node\_id wide

The "Hop\_Lim and node\_id wide" field is an 8-octet field defined as follows:

```

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |   Hop_Lim   |                               node_id           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ~                               node_id (contd)                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

```

Hop\_Lim: 1-octet unsigned integer. See Section 5.4.2.1 for the definition of the field.

node\_id: 7-octet unsigned integer. Node identifier field to uniquely identify a node within the IOAM-Namespace and associated IOAM-Domain. The procedure to allocate, manage and map the node\_ids is beyond the scope of this document.

## 5.4.2.10. ingress\_if\_id and egress\_if\_id wide

The "ingress\_if\_id and egress\_if\_id wide" field is an 8-octet field which is defined as follows:

```

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                               ingress_if_id                   |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                               egress_if_id                    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

```

ingress\_if\_id: 4-octet unsigned integer. Interface identifier to record the ingress interface the packet was received on.

egress\_if\_id: 4-octet unsigned integer. Interface identifier to record the egress interface the packet is forwarded out of.

## 5.4.2.11. namespace specific data wide

The "namespace specific data wide" field is an 8-octet field which can be used by the node to add IOAM-Namespace specific data. This represents a "free-format" 8-octet bit field with its semantics defined in the context of a specific IOAM-Namespace.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     namespace specific data                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     namespace specific data (contd)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 5.4.2.12. buffer occupancy

The "buffer occupancy" field is a 4-octet unsigned integer field. This field indicates the current status of the occupancy of the common buffer pool used by a set of queues. The units of this field are implementation specific. Hence, the units are interpreted within the context of an IOAM-Namespace and/or node-id if used. The authors acknowledge that in some operational cases there is a need for the units to be consistent across a packet path through the network, hence it is recommended for implementations to use standard units such as Bytes.

```

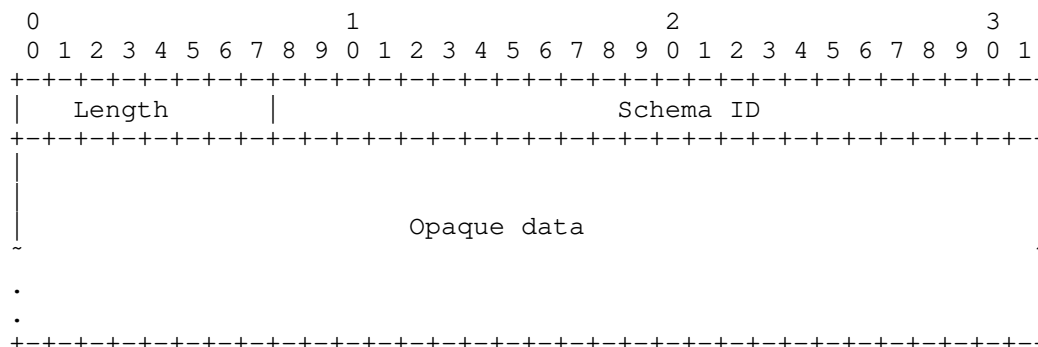
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     buffer occupancy                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

#### 5.4.2.13. Opaque State Snapshot

The "Opaque State Snapshot" is a variable length field and follows the fixed length IOAM-Data-Fields defined above. It allows the network element to store an arbitrary state in the node data field, without a pre-defined schema. The schema is to be defined within the context of an IOAM-Namespace. The schema needs to be made known to the analyzer by some out-of-band mechanism. The specification of this mechanism is beyond the scope of this document. A 24-bit "Schema Id" field, interpreted within the context of an IOAM-Namespace, indicates which particular schema is used, and has to be configured on the network element by the operator.





Length: 1-octet unsigned integer. It is the length in multiples of 4-octets of the Opaque data field that follows Schema Id.

Schema ID: 3-octet unsigned integer identifying the schema of Opaque data.

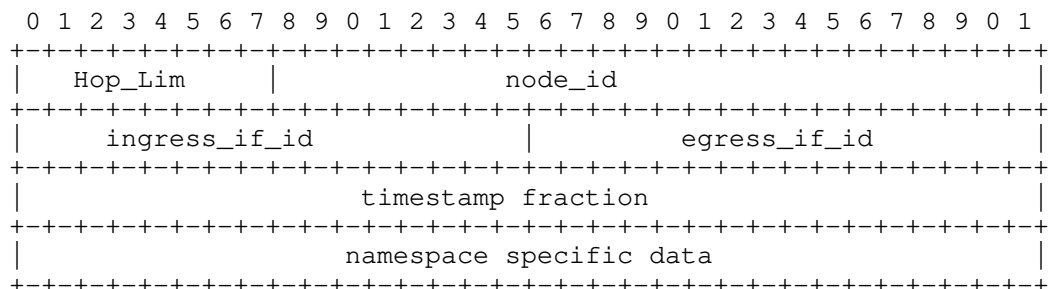
Opaque data: Variable length field. This field is interpreted as specified by the schema identified by the Schema ID.

When this field is part of the data field but a node populating the field has no opaque state data to report, the Length MUST be set to 0 and the Schema ID MUST be set to 0xFFFFF to mean no schema.

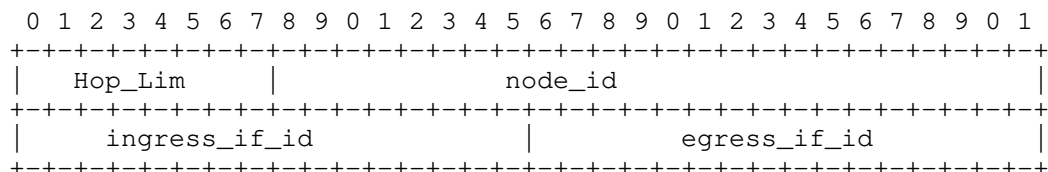
### 5.4.3. Examples of IOAM node data

The format used for the entries in a packet's "node data list" array can vary from packet to packet and deployment to deployment". Some deployments might only be interested in recording the node identifiers, whereas others might be interested in recording node identifiers and timestamps. This section provides example entries of the "node data list".

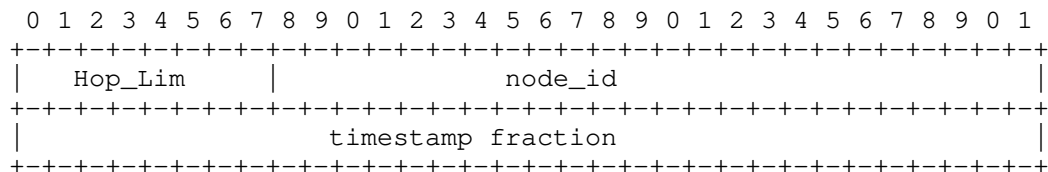
0xD40000: IOAM-Trace-Type is 0xD40000 (0b110101000000000000000000)  
then the format of node data is:



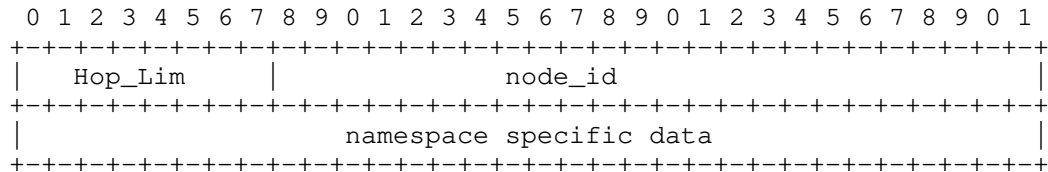
0xC00000: IOAM-Trace-Type is 0xC00000 (0b110000000000000000000000)  
then the format is:



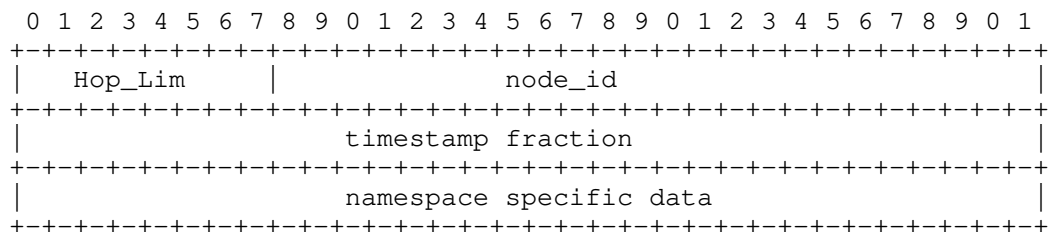
0x900000: IOAM-Trace-Type is 0x900000 (0b100100000000000000000000)  
then the format is:



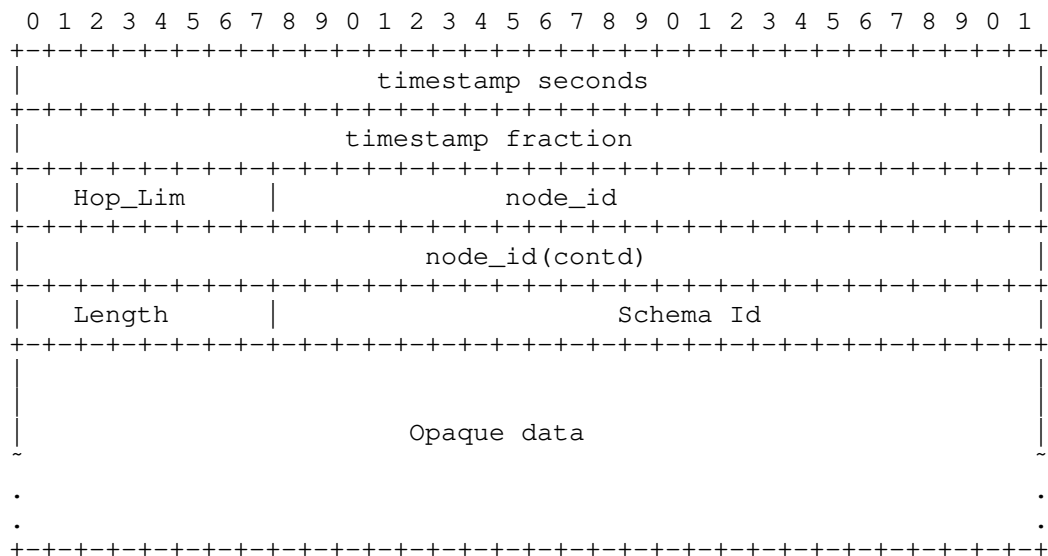
0x840000: IOAM-Trace-Type is 0x840000 (0b100001000000000000000000)  
then the format is:



0x940000: IOAM-Trace-Type is 0x940000 (0b100101000000000000000000)  
then the format is:



0x308002: IOAM-Trace-Type is 0x308002 (0b001100000100000000000000010)  
then the format is:



### 5.5. IOAM Proof of Transit Option-Type

IOAM Proof of Transit Option-Type is used to support path or service function chain [RFC7665] verification use cases, i.e., prove that traffic transited a defined path. While details on how the IOAM data for the Proof-of-transit option is processed at IOAM encapsulating, decapsulating and transit nodes are outside the scope of the document, proof of transit approaches share the need to uniquely identify a packet as well as iteratively operate on a set of information that is handed from node to node. Correspondingly, two pieces of information are added as IOAM-Data-Fields to the packet:

- o PktID: Unique identifier for the packet.

- o Cumulative: Information which is handed from node to node and updated by every node according to a verification algorithm.

The IOAM Proof-of-Transit Option-Type consist of a fixed size "IOAM proof of transit option header" and "IOAM proof of transit option data fields":

IOAM proof of transit option header:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Namespace-ID           | IOAM POT Type | IOAM POT flags |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IOAM proof of transit Option-Type IOAM-Data-Fields MUST be 4-octet aligned:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           POT Option data field determined by IOAM-POT-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Namespace-ID:** 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, the node MUST NOT change the contents of the IOAM-Data-Fields.

**IOAM POT Type:** 8-bit identifier of a particular POT variant that specifies the POT data that is included. This document defines POT Type 0:

0: POT data is a 16 Octet field to carry data associated to POT procedures.

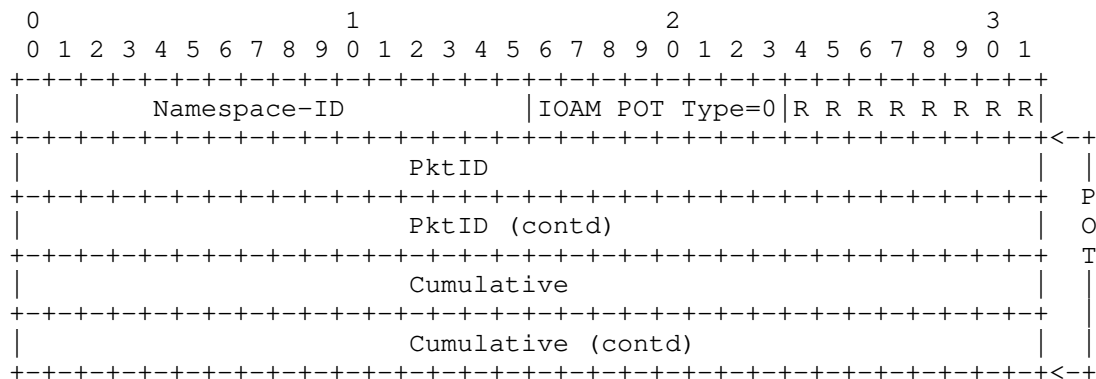
If a node receives an IOAM POT Type value that it does not understand, the node MUST NOT change, add to, or remove the contents of the OAM-Data-Fields.

**IOAM POT flags:** 8-bit. This document does not define any flags. Bits 0-7 These bits are available for assignment, see Section 8.5. Bits which have not been assigned MUST be set to zero upon transmission and ignored upon receipt.

POT Option data: Variable-length field. The type of which is determined by the IOAM-POT-Type.

#### 5.5.1. IOAM Proof of Transit Type 0

IOAM proof of transit option of IOAM POT Type 0:



Namespace-ID: 16-bit identifier of an IOAM-Namespace (see Section 5.5 above).

IOAM POT Type: 8-bit identifier of a particular POT variant that specifies the POT data that is included (see Section 5.5 above). For this case here, IOAM POT Type is set to the value 0.

Bit 0-7: Undefined (see Section 5.5 above).

PktID: 64-bit packet identifier.

Cumulative: 64-bit Cumulative that is updated at specific nodes by processing per packet PktID field and configured parameters.

Note: Larger or smaller sizes of "PktID" and "Cumulative" data are feasible and could be required for certain deployments, e.g., in case of space constraints in the encapsulation protocols used. Future documents could introduce different sizes of data for "proof of transit".

#### 5.6. IOAM Edge-to-Edge Option-Type

The IOAM Edge-to-Edge Option-Type is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating

node. The IOAM transit nodes MAY process the data but MUST NOT modify it.

The IOAM Edge-to-Edge Option-Type consist of a fixed size "IOAM Edge-to-Edge Option-Type header" and "IOAM Edge-to-Edge Option-Type data fields":

IOAM Edge-to-Edge Option-Type header:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Namespace-ID           |           IOAM-E2E-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IOAM Edge-to-Edge Option-Type IOAM-Data-Fields MUST be 4-octet aligned:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           E2E Option data field determined by IOAM-E2E-Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Namespace-ID:** 16-bit identifier of an IOAM-Namespace. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" (see Section 5.3) and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, then the node MUST NOT change the contents of the IOAM-Data-Fields.

**IOAM-E2E-Type:** A 16-bit identifier which specifies which data types are used in the E2E option data. The IOAM-E2E-Type value is a bit field. The order of packing the E2E option data field elements follows the bit order of the IOAM-E2E-Type field, as follows:

- Bit 0 (Most significant bit) When set indicates presence of a 64-bit sequence number added to a specific "packet group" which is used to detect packet loss, packet reordering, or packet duplication within the group. The "packet group" is deployment dependent and defined at the IOAM encapsulating node, e.g., by n-tuple based classification of packets. When this bit is set, "Bit 1" (for 32-bit sequence number, see below) MUST be zero.
- Bit 1 When set indicates presence of a 32-bit sequence number added to a specific "packet group" which is used to

detect packet loss, packet reordering, or packet duplication within that group. The "packet group" is deployment dependent and defined at the IOAM encapsulating node, e.g., by n-tuple based classification of packets. When this bit is set, "Bit 0" (for 64-bit sequence number, see above) MUST be zero.

- Bit 2      When set indicates presence of timestamp seconds, representing the time at which the packet entered the IOAM-domain. Within the IOAM encapsulating node, the time that the timestamp is retrieved can depend on the implementation. Some possibilities are: 1) the time at which the packet was received by the node, 2) the time at which the packet was transmitted by the node, 3) when a tunnel encapsulation is used, the point at which the packet is encapsulated into the tunnel. Each implementation has to document when the E2E timestamp that is going to be put in the packet is retrieved. This 4-octet field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp Seconds field contains the 32 most significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value that is valid for 1 second in approximately 136 years; the analyzer has to correlate several packets or compare the timestamp value to its own time-of-day in order to detect the error indication.
- Bit 3      When set indicates presence of timestamp fraction, representing the time at which the packet entered the IOAM-domain. This 4-octet field has three possible formats; based on either PTP (see e.g., [RFC8877]), NTP [RFC5905], or POSIX [POSIX]. The three timestamp formats are specified in Section 6. In all three cases, the Timestamp fraction field contains the 32 least significant bits of the timestamp format that is specified in Section 6. If a node is not capable of populating this field, it assigns the value 0xFFFFFFFF. Note that this is a legitimate value in the NTP format, valid for approximately 233 picoseconds in every second. If the NTP format is used the analyzer has to correlate several packets in order to detect the error indication.

Bit 4-15 Undefined. An IOAM encapsulating node MUST set the value of these bits to zero upon transmission and ignore upon receipt.

E2E Option data: Variable-length field. The type of which is determined by the IOAM-E2E-Type.

## 6. Timestamp Formats

The IOAM-Data-Fields include a timestamp field which is represented in one of three possible timestamp formats. It is assumed that the management plane is responsible for determining which timestamp format is used.

### 6.1. PTP Truncated Timestamp Format

The Precision Time Protocol (PTP) uses an 80-bit timestamp format. The truncated timestamp format is a 64-bit field, which is the 64 least significant bits of the 80-bit PTP timestamp. The PTP truncated format is specified in Section 4.3 of [RFC8877], and the details are presented below for the sake of completeness.

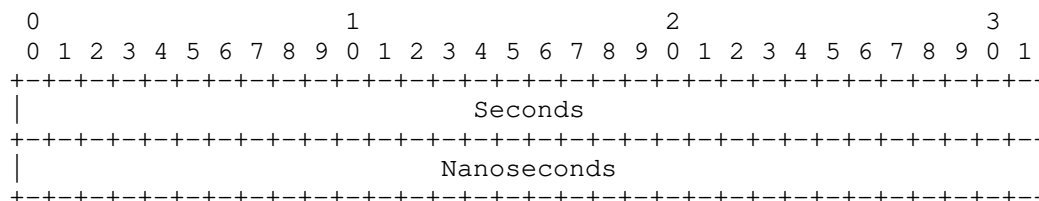


Figure 1: PTP Truncated Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the PTP epoch.

+ Size: 32 bits.

+ Units: seconds.

Nanoseconds: specifies the fractional portion of the number of seconds since the PTP epoch.

+ Size: 32 bits.



+ Units: nanoseconds. The value of this field is in the range 0 to  $(10^9)-1$ .

Epoch:

PTP epoch. For details see e.g., [RFC8877].

Resolution:

The resolution is 1 nanosecond.

Wraparound:

This time format wraps around every  $2^{32}$  seconds, which is roughly 136 years. The next wraparound will occur in the year 2106.

Synchronization Aspects:

It is assumed that nodes that run this protocol are synchronized among themselves. Nodes MAY be synchronized to a global reference time. Note that if PTP is used for synchronization, the timestamp MAY be derived from the PTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an PTP Grandmaster clock.

## 6.2. NTP 64-bit Timestamp Format

The Network Time Protocol (NTP) [RFC5905] timestamp format is 64 bits long. This specification uses the NTP timestamp format that is specified in Section 4.2.1 of [RFC8877], and the details are presented below for the sake of completeness.

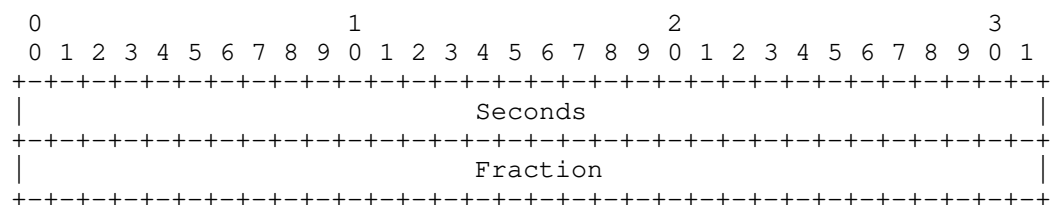


Figure 2: NTP [RFC5905] 64-bit Timestamp Format

Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the NTP epoch.

+ Size: 32 bits.

+ Units: seconds.

Fraction: specifies the fractional portion of the number of seconds since the NTP epoch.

+ Size: 32 bits.

+ Units: the unit is  $2^{-32}$  seconds, which is roughly equal to 233 picoseconds.

Epoch:

NTP Epoch. For details see [RFC5905].

Resolution:

The resolution is  $2^{-32}$  seconds.

Wraparound:

This time format wraps around every  $2^{32}$  seconds, which is roughly 136 years. The next wraparound will occur in the year 2036.

Synchronization Aspects:

Nodes that use this timestamp format will typically be synchronized to UTC using NTP [RFC5905]. Thus, the timestamp MAY be derived from the NTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an NTP server.

### 6.3. POSIX-based Timestamp Format

This timestamp format is based on the POSIX time format [POSIX]. The detailed specification of the timestamp format used in this document is presented below.

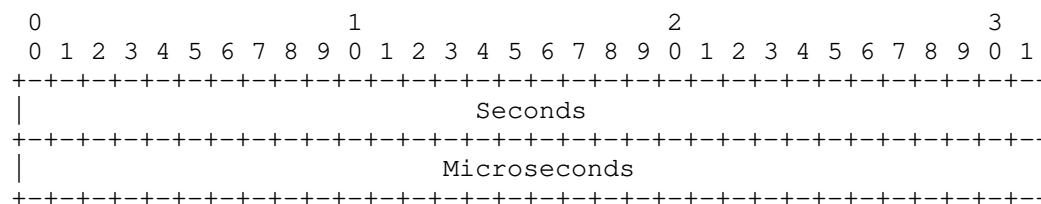


Figure 3: POSIX-based Timestamp Format

#### Timestamp field format:

Seconds: specifies the integer portion of the number of seconds since the POSIX epoch.

+ Size: 32 bits.

+ Units: seconds.

Microseconds: specifies the fractional portion of the number of seconds since the POSIX epoch.

+ Size: 32 bits.

+ Units: the unit is microseconds. The value of this field is in the range 0 to  $(10^6)-1$ .

#### Epoch:

POSIX epoch. For details, see [POSIX], appendix A.4.16.

#### Resolution:

The resolution is 1 microsecond.

#### Wraparound:

This time format wraps around every  $2^{32}$  seconds, which is roughly 136 years. The next wraparound will occur in the year 2106.

#### Synchronization Aspects:

It is assumed that nodes that use this timestamp format run the Linux operating system, and hence use the POSIX time. In some cases nodes MAY be synchronized to UTC using a synchronization mechanism that is outside the scope of this document, such as NTP [RFC5905]. Thus, the timestamp MAY be derived from the NTP-synchronized clock, allowing the timestamp to be measured with respect to the clock of an NTP server.

## 7. IOAM Data Export

IOAM nodes collect information for packets traversing a domain that supports IOAM. IOAM decapsulating nodes as well as IOAM transit nodes can choose to retrieve IOAM information from the packet, process the information further and export the information using e.g., IPFIX. The mechanisms and associated data formats for exporting IOAM data is outside the scope of this document.

A way to perform raw data export of IOAM data using IPFIX is discussed in [I-D.spiegel-ippm-ioam-rawexport].

## 8. IANA Considerations

This document requests the following IANA Actions.

IANA is requested to define a registry group named "In-Situ OAM (IOAM) Protocol Parameters".

This group will include the following registries:

- IOAM Option-Type

- IOAM Trace-Type

- IOAM Trace-Flags

- IOAM POT-Type

- IOAM POT-Flags

- IOAM E2E-Type

- IOAM Namespace-ID

The subsequent sub-sections detail the registries herein contained.

### 8.1. IOAM Option-Type Registry

This registry defines 128 code points for the IOAM Option-Type field for identifying IOAM Option-Types as explained in Section 5. The following code points are defined in this draft:

- 0 IOAM Pre-allocated Trace Option-Type

- 1 IOAM Incremental Trace Option-Type

- 2 IOAM POT Option-Type

- 3 IOAM E2E Option-Type

4 - 127 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Name: Name of the newly registered Option-Type.

Code point: Desired value of the requested code point.

Description: Brief description of the newly registered Option-Type.

Reference: Reference to the document that defines the new Option-Type.

The evaluation of a new registration request MUST also include checking whether the new IOAM Option-Type includes an IOAM-Namespace field and that the IOAM-Namespace field is the first field in the newly defined header of the new Option-Type.

## 8.2. IOAM Trace-Type Registry

This registry defines code point for each bit in the 24-bit IOAM-Trace-Type field for Pre-allocated Trace-Option-Type and Incremental Trace-Option-Type defined in Section 5.4. The meaning of Bits 0 - 11 is defined in this document in Paragraph 5 of Section 5.4.1:

Bit 0 hop\_Lim and node\_id in short format

Bit 1 ingress\_if\_id and egress\_if\_id in short format

Bit 2 timestamp seconds

Bit 3 timestamp fraction

Bit 4 transit delay

Bit 5 namespace specific data in short format

Bit 6 queue depth

Bit 7 checksum complement

Bit 8 hop\_Lim and node\_id in wide format

Bit 9 ingress\_if\_id and egress\_if\_id in wide format

Bit 10 namespace specific data in wide format

Bit 11 buffer occupancy

Bit 22 variable length Opaque State Snapshot

Bit 23 reserved

The meaning for Bits 12 - 21 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 24-bit IOAM Trace-Option-Type field for Pre-allocated Trace-Option-Type and Incremental Trace-Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

### 8.3. IOAM Trace-Flags Registry

This registry defines code points for each bit in the 4 bit flags for the Pre-allocated trace option and for the Incremental trace option defined in Section 5.4. The meaning of Bit 0 (the most significant bit) for trace flags is defined in this document in Paragraph 3 of Section 5.4.1:

Bit 0 "Overflow" (O-bit)

Bit 1 - 3 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit flags field of the Pre-allocated Trace-Option-Type and for the Incremental Trace-Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

### 8.4. IOAM POT-Type Registry

This registry defines 256 code points to define IOAM POT Type for IOAM proof of transit option Section 5.5. The code point value 0 is defined in this document:

0: 16 Octet POT data

1 - 255 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Name: Name of the newly registered POT-Type.

Code point: Desired value of the requested code point.

Description: Brief description of the newly registered POT-Type.

Reference: Reference to the document that defines the new POT-Type.

#### 8.5. IOAM POT-Flags Registry

This registry defines code points for each bit in the 8 bit flags for IOAM POT Option-Type defined in Section 5.5.

The meaning for Bits 0 - 7 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 8 bit flags field of the IOAM POT Option-Type.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

#### 8.6. IOAM E2E-Type Registry

This registry defines code points for each bit in the 16 bit IOAM-E2E-Type field for IOAM E2E option Section 5.6. The meaning of Bit 0 - 3 are defined in this document:

Bit 0 64-bit sequence number

Bit 1 32-bit sequence number

Bit 2 timestamp seconds

Bit 3 timestamp fraction

The meaning of Bits 4 - 15 are available for assignment via "IETF Review" process as per [RFC8126].

New registration requests MUST use the following template:

Bit: Desired bit to be allocated in the 16 bit IOAM-E2E-Type field.

Description: Brief description of the newly registered bit.

Reference: Reference to the document that defines the new bit.

#### 8.7. IOAM Namespace-ID Registry

IANA is requested to set up an "IOAM Namespace-ID Registry", containing 16-bit values and following the template for requests shown below. The meaning of 0x0000 is defined in this document. IANA is requested to reserve the values 0x0001 to 0x7FFF for private use (managed by operators), as specified in Section 5.3 of the current document. Registry entries for the values 0x8000 to 0xFFFF are to be assigned via the "Expert Review" policy defined in [RFC8126].

Upon receiving a new allocation request, a designated expert will perform the following:

- o Review whether the request is complete, i.e., the registration template has been filled in. The expert will send incomplete requests back to the requestor.
- o Check whether the request is neither a duplicate of nor conflicting with either an already existing allocation or a pending allocation. In case of duplicates or conflicts, the expert will ask the requestor to update the allocation request accordingly.
- o Solicit feedback from relevant working groups and communities to ensure that the new allocation request has been properly reviewed and that rough consensus on the request exists. At a minimum, the expert will solicit feedback from the IPPM working group in the IETF by posting the request to the `ippm@ietf.org` mailing list. The expert will allow for a 3-week review period on the mailing lists. If the feedback received from the relevant working groups and communities within the review period indicates rough consensus on the request, the expert will approve the request and ask IANA for allocating the new Namespace-ID. In case the expert senses a lack of consensus from the feedback received, the expert will ask the requestor to engage with the corresponding working groups and communities to further review and refine the request.

It is intended that any allocation will be accompanied by a published RFC. In order to allow for the allocation of code points prior to the RFC being approved for publication, the designated expert can approve allocations once it seems clear that an RFC will be published.

0x0000: default namespace (known to all IOAM nodes)



0x0001 - 0x7FFF: reserved for private use

0x8000 - 0xFFFF: unassigned

New registration requests MUST use the following template:

Name: Name of the newly registered Namespace-ID.

Code point: Desired value of the requested Namespace-ID.

Description: Brief description of the newly registered Namespace-ID.

Reference: Reference to the document that defines the new Namespace-ID.

Status of the registration: Status can be either "permanent" or "provisional". Namespace-ID registrations following a successful expert review will have the status "provisional". Once the RFC, which defines the new Namespace-ID is published, the status is changed to "permanent".

## 9. Management and Deployment Considerations

This document defines the structure and use of IOAM data fields. This document does not define the encapsulation of IOAM data fields into different protocols. Management and deployment aspects for IOAM have to be considered within the context of the protocol IOAM data fields are encapsulated into and as such, are out of scope for this document. For a discussion of IOAM deployment, please also refer to [I-D.ietf-ippm-ioam-deployment], which outlines a framework for IOAM deployment and provides best current practices.

## 10. Security Considerations

As discussed in [RFC7276], a successful attack on an OAM protocol in general, and specifically on IOAM, can prevent the detection of failures or anomalies, or create a false illusion of nonexistent ones. In particular, these threats are applicable by compromising the integrity of IOAM data, either by maliciously modifying IOAM options in transit, or by injecting packets with maliciously generated IOAM options. All nodes in the path of a IOAM carrying packet can perform such an attack.

The Proof of Transit Option-Type (see Section 5.5) is used for verifying the path of data packets, i.e., proving that packets transited through a defined set of nodes.

In case an attacker gains access to several nodes in a network and would be able to change the system software of these nodes, IOAM data fields could be misused and repurposed for a use different from what is specified in this document. One type of misuse is the implementation of a covert channel between network nodes.

From a confidentiality perspective, although IOAM options are not expected to contain user data, they can be used for network reconnaissance, allowing attackers to collect information about network paths, performance, queue states, buffer occupancy and other information. Moreover, if IOAM data leaks from the IOAM-domain it could enable reconnaissance beyond the scope of the IOAM-domain. One possible application of such reconnaissance is to gauge the effectiveness of an ongoing attack, e.g., if buffers and queues are overflowing.

IOAM can be used as a means for implementing Denial of Service (DoS) attacks, or for amplifying them. For example, a malicious attacker can add an IOAM header to packets in order to consume the resources of network devices that take part in IOAM or entities that receive, collect or analyze the IOAM data. Another example is a packet length attack, in which an attacker pushes headers associated with IOAM Option-Types into data packets, causing these packets to be increased beyond the MTU size, resulting in fragmentation or in packet drops. In case POT is used, an attacker could corrupt the POT data fields in the packet, resulting in a verification failure of the POT data, even if the packet followed the correct path.

Since IOAM options can include timestamps, if network devices use synchronization protocols then any attack on the time protocol [RFC7384] can compromise the integrity of the timestamp-related data fields.

At the management plane, attacks can be set up by misconfiguring or by maliciously configuring IOAM-enabled nodes in a way that enables other attacks. IOAM configuration should only be managed by authorized processes or users.

IETF protocols require features to ensure their security. While IOAM data fields don't represent a protocol by themselves, the IOAM data fields add to the protocol that the IOAM data fields are encapsulated into. Any specification that defines how IOAM data fields are carried in an encapsulating protocol MUST provide for a mechanism for cryptographic integrity protection of the IOAM data fields. Cryptographic integrity protection could be either achieved through a mechanism of the encapsulating protocol or it could incorporate the mechanisms specified in [I-D.ietf-ippm-ioam-data-integrity].

The current document does not define a specific IOAM encapsulation. It has to be noted that some IOAM encapsulation types can introduce specific security considerations. A specification that defines an IOAM encapsulation is expected to address the respective encapsulation-specific security considerations.

Notably, IOAM is expected to be deployed in limited domains, thus confining the potential attack vectors to within the limited domain. A limited administrative domain provides the operator with the means to select, monitor, and control the access of all the network devices, making these devices trusted by the operator. Indeed, in order to limit the scope of threats mentioned above to within the current limited domain the network operator is expected to enforce policies that prevent IOAM traffic from leaking outside of the IOAM domain, and prevent IOAM data from outside the domain to be processed and used within the domain.

This document does not define the data contents of custom fields like "Opaque State Snapshot" and "namespace specific data" IOAM data fields. These custom data fields will have security considerations corresponding to their defined data contents that need to be described where those formats are defined.

IOAM deployments which leverage both IOAM Trace Option-Types, i.e., the Pre-allocated Trace Option-Type and Incremental Trace Option-Type can suffer from incomplete visibility if the information gathered via the two Trace Option-Types is not correlated and aggregated appropriately. If IOAM transit nodes leverage the IOAM data fields in the packet for further actions or insights, then IOAM transit nodes which only support one IOAM Trace Option-Type in an IOAM deployment which leverages both Trace Option-Types, have limited visibility and thus can draw inappropriate conclusions or take wrong actions.

The security considerations of a system that deploys IOAM, much like any system, has to be reviewed on a per-deployment-scenario basis, based on a systems-specific threat analysis, which can lead to specific security solutions that are beyond the scope of the current document. Specifically, in an IOAM deployment that is not confined to a single LAN, but spans multiple inter-connected sites (for example, using an overlay network), the inter-site links can be secured (e.g., by IPsec) in order to avoid external threats.

IOAM deployment considerations, including approaches to mitigate the above discussed threads and potential attacks are outside the scope of this document. IOAM deployment considerations are discussed in [I-D.ietf-ippm-ioam-deployment].

## 11. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, LJ Wobker, Erik Nordmark, Vengada Prasad Govindan, Andrew Yourtchenko, Aviv Kfir, Tianran Zhou, Zhenbin (Robin) and Greg Mirsky for the comments and advice.

This document leverages and builds on top of several 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.

The authors would like to gracefully acknowledge useful review and insightful comments received from Joe Clarke, Al Morton, Tom Herbert, Carlos Bernardos, Haoyu Song, Mickey Spiegel, Roman Danyliw, Benjamin Kaduk, Murray S. Kucherawy, Ian Swett, Martin Duke, Francesca Palombini, Lars Eggert, Alvaro Retana, Erik Kline, Robert Wilton, Zaheduzzaman Sarker, Dan Romascanu and Barak Gafni.

## 12. References

### 12.1. Normative References

- [POSIX] Institute of Electrical and Electronics Engineers, "IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2017) - IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(TM) Base Specifications, Issue 7)", IEEE Std 1003.1-2017, 2017, <<https://standards.ieee.org/findstds/standard/1003.1-2017.html>>.
- [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>>.
- [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>>.
- [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>>.

## 12.2. Informative References

- [I-D.ietf-ippm-ioam-data-integrity]  
Brockners, F., Bhandari, S., and T. Mizrahi, "Integrity of In-situ OAM Data Fields", draft-ietf-ippm-ioam-data-integrity-00 (work in progress), October 2021.
- [I-D.ietf-ippm-ioam-deployment]  
Brockners, F., Bhandari, S., Bernier, D., and T. Mizrahi, "In-situ OAM Deployment", draft-ietf-ippm-ioam-deployment-00 (work in progress), October 2021.
- [I-D.ietf-nvo3-vxlan-gpe]  
(Editor), F. M., (editor), L. K., and U. E. (editor), "Generic Protocol Extension for VXLAN (VXLAN-GPE)", draft-ietf-nvo3-vxlan-gpe-12 (work in progress), September 2021.
- [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.
- [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-05 (work in progress), July 2021.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.

- [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>>.
- [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>>.
- [RFC7821] Mizrahi, T., "UDP Checksum Complement in the Network Time Protocol (NTP)", RFC 7821, DOI 10.17487/RFC7821, March 2016, <<https://www.rfc-editor.org/info/rfc7821>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8877] Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", RFC 8877, DOI 10.17487/RFC8877, September 2020, <<https://www.rfc-editor.org/info/rfc8877>>.
- [RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <<https://www.rfc-editor.org/info/rfc8926>>.

#### Contributors' Addresses

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

Email: [cpignata@cisco.com](mailto:cpignata@cisco.com)

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

US

Email: mickey.spiegel@intel.com

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

Email: gbarak@nvidia.com

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

Email: jennifer.lemon@broadcom.com

Hannes Gredler  
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy  
United States

Email: john@leddy.net

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

Email: stephen.youell@jpmorgan.com

David Mozes

Email: mosesster@gmail.com

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

Email: petr@fb.com

Remy Chang  
Barefoot Networks  
4750 Patrick Henry Drive  
Santa Clara, CA 95054  
US

Email: remy@barefootnetworks.com

Daniel Bernier  
Bell Canada  
Canada

Email: daniel.bernier@bell.ca

#### Authors' Addresses

Frank Brockners (editor)  
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



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

Email: tal.mizrahi.phd@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 10, 2020

M. Bagnulo  
UC3M  
B. Claise  
Cisco Systems, Inc.  
P. Eardley  
BT  
A. Morton  
AT&T Labs  
A. Akhter  
Consultant  
March 9, 2020

Registry for Performance Metrics  
draft-ietf-ippm-metric-registry-24

Abstract

This document defines the format for the IANA Performance Metrics Registry. This document also gives a set of guidelines for Registered Performance Metric requesters and reviewers.

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 September 10, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	5
3. Scope . . . . .	7
4. Motivation for a Performance Metrics Registry . . . . .	8
4.1. Interoperability . . . . .	8
4.2. Single point of reference for Performance Metrics . . . . .	9
4.3. Side benefits . . . . .	9
5. Criteria for Performance Metrics Registration . . . . .	9
6. Performance Metric Registry: Prior attempt . . . . .	10
6.1. Why this Attempt Should Succeed . . . . .	11
7. Definition of the Performance Metric Registry . . . . .	11
7.1. Summary Category . . . . .	13
7.1.1. Identifier . . . . .	13
7.1.2. Name . . . . .	13
7.1.3. URI . . . . .	17
7.1.4. Description . . . . .	17
7.1.5. Reference . . . . .	17
7.1.6. Change Controller . . . . .	17
7.1.7. Version (of Registry Format) . . . . .	18
7.2. Metric Definition Category . . . . .	18
7.2.1. Reference Definition . . . . .	18
7.2.2. Fixed Parameters . . . . .	18
7.3. Method of Measurement Category . . . . .	19
7.3.1. Reference Method . . . . .	19
7.3.2. Packet Stream Generation . . . . .	19
7.3.3. Traffic Filter . . . . .	20
7.3.4. Sampling Distribution . . . . .	20
7.3.5. Run-time Parameters . . . . .	21
7.3.6. Role . . . . .	22
7.4. Output Category . . . . .	22
7.4.1. Type . . . . .	22
7.4.2. Reference Definition . . . . .	23
7.4.3. Metric Units . . . . .	23
7.4.4. Calibration . . . . .	23
7.5. Administrative information . . . . .	24
7.5.1. Status . . . . .	24
7.5.2. Requester . . . . .	24
7.5.3. Revision . . . . .	24
7.5.4. Revision Date . . . . .	24
7.6. Comments and Remarks . . . . .	24

8. Processes for Managing the Performance Metric Registry Group	24
8.1. Adding new Performance Metrics to the Performance Metrics Registry	25
8.2. Revising Registered Performance Metrics	26
8.3. Deprecating Registered Performance Metrics	28
9. Security considerations	28
10. IANA Considerations	29
10.1. Registry Group	29
10.2. Performance Metric Name Elements	29
10.3. New Performance Metrics Registry	30
11. Blank Registry Template	32
11.1. Summary	32
11.1.1. ID (Identifier)	32
11.1.2. Name	32
11.1.3. URI	32
11.1.4. Description	32
11.1.5. Change Controller	32
11.1.6. Version (of Registry Format)	32
11.2. Metric Definition	32
11.2.1. Reference Definition	32
11.2.2. Fixed Parameters	32
11.3. Method of Measurement	33
11.3.1. Reference Method	33
11.3.2. Packet Stream Generation	33
11.3.3. Traffic Filtering (observation) Details	33
11.3.4. Sampling Distribution	33
11.3.5. Run-time Parameters and Data Format	33
11.3.6. Roles	33
11.4. Output	33
11.4.1. Type	34
11.4.2. Reference Definition	34
11.4.3. Metric Units	34
11.4.4. Calibration	34
11.5. Administrative items	34
11.5.1. Status	34
11.5.2. Requester	34
11.5.3. Revision	34
11.5.4. Revision Date	34
11.6. Comments and Remarks	34
12. Acknowledgments	34
13. References	35
13.1. Normative References	35
13.2. Informative References	36
Authors' Addresses	37

## 1. Introduction

The IETF specifies and uses Performance Metrics of protocols and applications transported over its protocols. Performance metrics are important part of network operations using IETF protocols, and [RFC6390] specifies guidelines for their development.

The definition and use of Performance Metrics in the IETF has been fostered in various working groups (WG), most notably:

The "IP Performance Metrics" (IPPM) WG is the WG primarily focusing on Performance Metrics definition at the IETF.

The "Benchmarking Methodology" WG (BMWG) defines many Performance Metrics for use in laboratory benchmarking of inter-networking technologies.

The "Metric Blocks for use with RTCP's Extended Report Framework" (XRBLOCK) WG (concluded) specified many Performance Metrics related to "RTP Control Protocol Extended Reports (RTCP XR)" [RFC3611], which establishes a framework to allow new information to be conveyed in RTCP, supplementing the original report blocks defined in "RTP: A Transport Protocol for Real-Time Applications", [RFC3550].

The "IP Flow Information eXport" (IPFIX) concluded WG specified an IANA process for new Information Elements. Some Performance Metrics related Information Elements are proposed on regular basis.

The "Performance Metrics for Other Layers" (PMOL) a concluded WG defined some Performance Metrics related to Session Initiation Protocol (SIP) voice quality [RFC6035].

It is expected that more Performance Metrics will be defined in the future, not only IP-based metrics, but also metrics which are protocol-specific and application-specific.

Despite the importance of Performance Metrics, there are two related problems for the industry. First, ensuring that when one party requests another party to measure (or report or in some way act on) a particular Performance Metric, then both parties have exactly the same understanding of what Performance Metric is being referred to. Second, discovering which Performance Metrics have been specified, to avoid developing a new Performance Metric that is very similar, but not quite inter-operable. These problems can be addressed by creating a registry of performance metrics. The usual way in which the IETF organizes registries is with Internet Assigned Numbers

Authority (IANA), and there is currently no Performance Metrics Registry maintained by the IANA.

This document requests that IANA create and maintain a Performance Metrics Registry, according to the maintenance procedures and the Performance Metrics Registry format defined in this memo. The resulting Performance Metrics Registry is for use by the IETF and others. Although the Registry formatting specifications herein are primarily for registry creation by IANA, any other organization that wishes to create a performance metrics registry may use the same formatting specifications for their purposes. The authors make no guarantee of the registry format's applicability to any possible set of Performance Metrics envisaged by other organizations, but encourage others to apply it. In the remainder of this document, unless we explicitly say otherwise, we will refer to the IANA-maintained Performance Metrics Registry as simply the Performance Metrics Registry.

## 2. Terminology

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.

**Performance Metric:** A Performance Metric is a quantitative measure of performance, targeted to an IETF-specified protocol or targeted to an application transported over an IETF-specified protocol. Examples of Performance Metrics are the FTP response time for a complete file download, the DNS response time to resolve the IP address(es), a database logging time, etc. This definition is consistent with the definition of metric in [RFC2330] and broader than the definition of performance metric in [RFC6390].

**Registered Performance Metric:** A Registered Performance Metric is a Performance Metric expressed as an entry in the Performance Metrics Registry, administered by IANA. Such a performance metric has met all the registry review criteria defined in this document in order to be included in the registry.

**Performance Metrics Registry:** The IANA registry containing Registered Performance Metrics.

**Proprietary Registry:** A set of metrics that are registered in a proprietary registry, as opposed to Performance Metrics Registry.

**Performance Metrics Experts:** The Performance Metrics Experts is a group of designated experts [RFC8126] selected by the IESG to validate the Performance Metrics before updating the Performance Metrics Registry. The Performance Metrics Experts work closely with IANA.

**Parameter:** A Parameter is an input factor defined as a variable in the definition of a Performance Metric. A Parameter is a numerical or other specified factor forming one of a set that defines a metric or sets the conditions of its operation. All Parameters must be known in order to make a measurement using a metric and interpret the results. There are two types of Parameters: Fixed and Run-time parameters. For the Fixed Parameters, the value of the variable is specified in the Performance Metrics Registry entry and different Fixed Parameter values results in different Registered Performance Metrics. For the Run-time Parameters, the value of the variable is defined when the metric measurement method is executed and a given Registered Performance Metric supports multiple values for the parameter. Although Run-time Parameters do not change the fundamental nature of the Performance Metric's definition, some have substantial influence on the network property being assessed and interpretation of the results.

Note: Consider the case of packet loss in the following two Active Measurement Method cases. The first case is packet loss as background loss where the Run-time Parameter set includes a very sparse Poisson stream, and only characterizes the times when packets were lost. Actual user streams likely see much higher loss at these times, due to tail drop or radio errors. The second case is packet loss as inverse of throughput where the Run-time Parameter set includes a very dense, bursty stream, and characterizes the loss experienced by a stream that approximates a user stream. These are both "loss metrics", but the difference in interpretation of the results is highly dependent on the Run-time Parameters (at least), to the extreme where we are actually using loss to infer its compliment: delivered throughput.

**Active Measurement Method:** Methods of Measurement conducted on traffic which serves only the purpose of measurement and is generated for that reason alone, and whose traffic characteristics are known a priori. The complete definition of Active Methods is specified in section 3.4 of [RFC7799]. Examples of Active Measurement Methods are the measurement methods for the One way delay metric defined in [RFC7679] and the one for round trip delay defined in [RFC2681].

**Passive Measurement Method:** Methods of Measurement conducted on network traffic, generated either from the end users or from network elements that would exist regardless whether the measurement was being conducted or not. The complete definition of Passive Methods is specified in section 3.6 of [RFC7799]. One characteristic of Passive Measurement Methods is that sensitive information may be observed, and as a consequence, stored in the measurement system.

**Hybrid Measurement Method:** Hybrid Methods are Methods of Measurement that use a combination of Active Methods and Passive Methods, to assess Active Metrics, Passive Metrics, or new metrics derived from the a priori knowledge and observations of the stream of interest. The complete definition of Hybrid Methods is specified in section 3.8 of [RFC7799].

### 3. Scope

This document is intended for two different audiences:

1. For those defining new Registered Performance Metrics, it provides specifications and best practices to be used in deciding which Registered Performance Metrics are useful for a measurement study, instructions for writing the text for each column of the Registered Performance Metrics, and information on the supporting documentation required for the new Performance Metrics Registry entry (up to and including the publication of one or more immutable documents such as an RFC).
2. For the appointed Performance Metrics Experts and for IANA personnel administering the new IANA Performance Metrics Registry, it defines a set of acceptance criteria against which these proposed Registered Performance Metrics should be evaluated.

In addition, this document may be useful for other organizations who are defining a Performance Metric registry of their own, and may re-use the features of the Performance Metrics Registry defined in this document.

This Performance Metrics Registry is applicable to Performance Metrics issued from Active Measurement, Passive Measurement, and any other form of Performance Metric. This registry is designed to encompass Performance Metrics developed throughout the IETF and especially for the technologies specified in the following working groups: IPPM, XRBLOCK, IPFIX, and BMWG. This document analyzes a prior attempt to set up a Performance Metrics Registry, and the reasons why this design was inadequate [RFC6248]. Finally, this



document gives a set of guidelines for requesters and expert reviewers of candidate Registered Performance Metrics.

This document makes no attempt to populate the Performance Metrics Registry with initial entries; the related memo [I-D.ietf-ippm-initial-registry] proposes the initial set of registry entries.

#### 4. Motivation for a Performance Metrics Registry

In this section, we detail several motivations for the Performance Metrics Registry.

##### 4.1. Interoperability

As with any IETF registry, the primary intention is to manage registration of identifiers for use within one or more protocols. In the particular case of the Performance Metrics Registry, there are two types of protocols that will use the Performance Metrics in the Performance Metrics Registry during their operation (by referring to the Index values):

- o Control protocol: This type of protocol used to allow one entity to request another entity to perform a measurement using a specific metric defined by the Performance Metrics Registry. One particular example is the LMAP framework [RFC7594]. Using the LMAP terminology, the Performance Metrics Registry is used in the LMAP Control protocol to allow a Controller to schedule a measurement task for one or more Measurement Agents. In order to enable this use case, the entries of the Performance Metrics Registry must be sufficiently defined to allow a Measurement Agent implementation to trigger a specific measurement task upon the reception of a control protocol message. This requirement heavily constrains the type of entries that are acceptable for the Performance Metrics Registry.
- o Report protocol: This type of protocol is used to allow an entity to report measurement results to another entity. By referencing to a specific Performance Metrics Registry, it is possible to properly characterize the measurement result data being reported. Using the LMAP terminology, the Performance Metrics Registry is used in the Report protocol to allow a Measurement Agent to report measurement results to a Collector.

It should be noted that the LMAP framework explicitly allows for using not only the IANA-maintained Performance Metrics Registry but also other registries containing Performance Metrics, either defined by other organizations or private ones. However, others who are

creating Registries to be used in the context of an LMAP framework are encouraged to use the Registry format defined in this document, because this makes it easier for developers of LMAP Measurement Agents (MAs) to programmatically use information found in those other Registries' entries.

#### 4.2. Single point of reference for Performance Metrics

A Performance Metrics Registry serves as a single point of reference for Performance Metrics defined in different working groups in the IETF. As we mentioned earlier, there are several WGs that define Performance Metrics in the IETF and it is hard to keep track of all them. This results in multiple definitions of similar Performance Metrics that attempt to measure the same phenomena but in slightly different (and incompatible) ways. Having a registry would allow the IETF community and others to have a single list of relevant Performance Metrics defined by the IETF (and others, where appropriate). The single list is also an essential aspect of communication about Performance Metrics, where different entities that request measurements, execute measurements, and report the results can benefit from a common understanding of the referenced Performance Metric.

#### 4.3. Side benefits

There are a couple of side benefits of having such a registry. First, the Performance Metrics Registry could serve as an inventory of useful and used Performance Metrics, that are normally supported by different implementations of measurement agents. Second, the results of measurements using the Performance Metrics should be comparable even if they are performed by different implementations and in different networks, as the Performance Metric is properly defined. BCP 176 [RFC6576] examines whether the results produced by independent implementations are equivalent in the context of evaluating the completeness and clarity of metric specifications. This BCP defines the standards track advancement testing for (active) IPPM metrics, and the same process will likely suffice to determine whether Registered Performance Metrics are sufficiently well specified to result in comparable (or equivalent) results. Registered Performance Metrics which have undergone such testing SHOULD be noted, with a reference to the test results.

#### 5. Criteria for Performance Metrics Registration

It is neither possible nor desirable to populate the Performance Metrics Registry with all combinations of Parameters of all Performance Metrics. The Registered Performance Metrics SHOULD be:

1. interpretable by the user.
2. implementable by the software or hardware designer,
3. deployable by network operators,
4. accurate in terms of producing equivalent results, and for interoperability and deployment across vendors,
5. Operationally useful, so that it has significant industry interest and/or has seen deployment,
6. Sufficiently tightly defined, so that different values for the Run-time Parameters does not change the fundamental nature of the measurement, nor change the practicality of its implementation.

In essence, there needs to be evidence that a candidate Registered Performance Metric has significant industry interest, or has seen deployment, and there is agreement that the candidate Registered Performance Metric serves its intended purpose.

#### 6. Performance Metric Registry: Prior attempt

There was a previous attempt to define a metric registry RFC 4148 [RFC4148]. However, it was obsoleted by RFC 6248 [RFC6248] because it was "found to be insufficiently detailed to uniquely identify IPPM metrics... [there was too much] variability possible when characterizing a metric exactly" which led to the RFC4148 registry having "very few users, if any".

A couple of interesting additional quotes from RFC 6248 [RFC6248] might help to understand the issues related to that registry.

1. "It is not believed to be feasible or even useful to register every possible combination of Type P, metric parameters, and Stream parameters using the current structure of the IPPM Metrics Registry."
2. "The registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics."
3. "Despite apparent efforts to find current or even future users, no one responded to the call for interest in the RFC 4148 registry during the second half of 2010."

The current approach learns from this by tightly defining each Registered Performance Metric with only a few variable (Run-time) Parameters to be specified by the measurement designer, if any. The

idea is that entries in the Performance Metrics Registry stem from different measurement methods which require input (Run-time) parameters to set factors like source and destination addresses (which do not change the fundamental nature of the measurement). The downside of this approach is that it could result in a large number of entries in the Performance Metrics Registry. There is agreement that less is more in this context - it is better to have a reduced set of useful metrics rather than a large set of metrics, some with questionable usefulness.

#### 6.1. Why this Attempt Should Succeed

As mentioned in the previous section, one of the main issues with the previous registry was that the metrics contained in the registry were too generic to be useful. This document specifies stricter criteria for performance metric registration (see section 5), and imposes a group of Performance Metrics Experts that will provide guidelines to assess if a Performance Metric is properly specified.

Another key difference between this attempt and the previous one is that in this case there is at least one clear user for the Performance Metrics Registry: the LMAP framework and protocol. Because the LMAP protocol will use the Performance Metrics Registry values in its operation, this actually helps to determine if a metric is properly defined. In particular, since we expect that the LMAP control protocol will enable a controller to request a measurement agent to perform a measurement using a given metric by embedding the Performance Metrics Registry identifier in the protocol. Such a metric and method are properly specified if they are defined well-enough so that it is possible (and practical) to implement them in the measurement agent. This was the failure of the previous attempt: a registry entry with an undefined Type-P (section 13 of RFC 2330 [RFC2330]) allows implementation to be ambiguous.

### 7. Definition of the Performance Metric Registry

This Performance Metrics Registry is applicable to Performance Metrics used for Active Measurement, Passive Measurement, and any other form of Performance Measurement. Each category of measurement has unique properties, so some of the columns defined below are not applicable for a given metric category. In this case, the column(s) SHOULD be populated with the "NA" value (Non Applicable). However, the "NA" value MUST NOT be used by any metric in the following columns: Identifier, Name, URI, Status, Requester, Revision, Revision Date, Description. In the future, a new category of metrics could require additional columns, and adding new columns is a recognized form of registry extension. The specification defining the new

column(s) MUST give general guidelines for populating the new column(s) for existing entries.

The columns of the Performance Metrics Registry are defined below. The columns are grouped into "Categories" to facilitate the use of the registry. Categories are described at the 7.x heading level, and columns are at the 7.x.y heading level. The Figure below illustrates this organization. An entry (row) therefore gives a complete description of a Registered Performance Metric.

Each column serves as a check-list item and helps to avoid omissions during registration and expert review.

=====

Legend:

Registry Categories and Columns are shown below as:

Category

-----...

Column | Column |...

=====

Summary

-----

Identifier	Name	URI	Desc.	Reference	Change Controller	Ver
------------	------	-----	-------	-----------	-------------------	-----

Metric Definition

-----

Reference Definition	Fixed Parameters
----------------------	------------------

Method of Measurement

-----

Reference Method	Packet Stream Generation	Traffic Filter	Sampling Distribution	Run-time Parameters	Role
------------------	--------------------------	----------------	-----------------------	---------------------	------

Output

-----

Type	Reference Definition	Units	Calibration
------	----------------------	-------	-------------

Administrative Information

-----

Status	Requester	Rev	Rev.Date
--------	-----------	-----	----------

Comments and Remarks

-----

There is a blank template of the Registry template provided in Section 11 of this memo.

## 7.1. Summary Category

### 7.1.1. Identifier

A numeric identifier for the Registered Performance Metric. This identifier **MUST** be unique within the Performance Metrics Registry.

The Registered Performance Metric unique identifier is an unbounded integer (range 0 to infinity).

The Identifier 0 should be Reserved. The Identifier values from 64512 to 65536 are reserved for private or experimental use, and the user may encounter overlapping uses.

When adding newly Registered Performance Metrics to the Performance Metrics Registry, IANA **SHOULD** assign the lowest available identifier to the new Registered Performance Metric.

If a Performance Metrics Expert providing review determines that there is a reason to assign a specific numeric identifier, possibly leaving a temporary gap in the numbering, then the Performance Expert **SHALL** inform IANA of this decision.

### 7.1.2. Name

As the name of a Registered Performance Metric is the first thing a potential human implementor will use when determining whether it is suitable for their measurement study, it is important to be as precise and descriptive as possible. In future, users will review the names to determine if the metric they want to measure has already been registered, or if a similar entry is available as a basis for creating a new entry.

Names are composed of the following elements, separated by an underscore character "\_":

MetricType\_Method\_SubTypeMethod\_... Spec\_Units\_Output

- o MetricType: a combination of the directional properties and the metric measured, such as and not limited to:

- RTDelay (Round Trip Delay)

- RTDNS (Response Time Domain Name Service)

- RLDNS (Response Loss Domain Name Service)

- OWDelay (One Way Delay)

RTLoss (Round Trip Loss)

OWLoss (One Way Loss)

OWPDV (One Way Packet Delay Variation)

OWIPDV (One Way Inter-Packet Delay Variation)

OWReorder (One Way Packet Reordering)

OWDuplic (One Way Packet Duplication)

OWBTC (One Way Bulk Transport Capacity)

OWMBM (One Way Model Based Metric)

SPMonitor (Single Point Monitor)

MPMonitor (Multi-Point Monitor)

- o Method: One of the methods defined in [RFC7799], such as and not limited to:

Active (depends on a dedicated measurement packet stream and observations of the stream)

Passive (depends *\*solely\** on observation of one or more existing packet streams)

HybridType1 (observations on one stream that combine both active and passive methods)

HybridType2 (observations on two or more streams that combine both active and passive methods)

Spatial (Spatial Metric of RFC5644)

- o SubTypeMethod: One or more sub-types to further describe the features of the entry, such as and not limited to:

ICMP (Internet Control Message Protocol)

IP (Internet Protocol)

DSCPxx (where xx is replaced by a Diffserv code point)

UDP (User Datagram Protocol)

TCP (Transport Control Protocol)

QUIC (QUIC transport protocol)

HS (Hand-Shake, such as TCP's 3-way HS)

Poisson (Packet generation using Poisson distribution)

Periodic (Periodic packet generation)

SendOnRcv (Sender keeps one packet in-transit by sending when previous packet arrives)

PayloadxxxxB (where xxxx is replaced by an integer, the number of octets in the Payload))

SustainedBurst (Capacity test, worst case)

StandingQueue (test of bottleneck queue behavior)

SubTypeMethod values are separated by a hyphen "-" character, which indicates that they belong to this element, and that their order is unimportant when considering name uniqueness.

- o Spec: An immutable document identifier combined with a document section identifier. For RFCs, this consists of the RFC number and major section number that specifies this Registry entry in the form RFCXXXXsecY, such as RFC7799sec3. Note: the RFC number is not the Primary Reference specification for the metric definition, such as [RFC7679] for One-way Delay; it will contain the placeholder "RFCXXXXsecY" until the RFC number is assigned to the specifying document, and would remain blank in private registry entries without a corresponding RFC. Anticipating the "RFC10K" problem, the number of the RFC continues to replace RFCXXXX regardless of the number of digits in the RFC number. Anticipating Registry Entries from other standards bodies, the form of this Name Element MUST be proposed and reviewed for consistency and uniqueness by the Expert Reviewer.
- o Units: The units of measurement for the output, such as and not limited to:

Seconds

Ratio (unitless)



Percent (value multiplied by 100%)

Logical (1 or 0)

Packets

BPS (Bits per Second)

PPS (Packets per Second)

EventTotal (for unit-less counts)

Multiple (more than one type of unit)

Enumerated (a list of outcomes)

Unitless

- o Output: The type of output resulting from measurement, such as and not limited to:

Singleton

Raw (multiple Singletons)

Count

Minimum

Maximum

Median

Mean

95Percentile (95th Percentile)

99Percentile (99th Percentile)

StdDev (Standard Deviation)

Variance

PFI (Pass, Fail, Inconclusive)

FlowRecords (descriptions of flows observed)

LossRatio (lost packets to total packets, <=1)

An example is:

RTDelay\_Active\_IP-UDP-Periodic\_RFCXXXXsecY\_Seconds\_95Percentile

as described in section 4 of [I-D.ietf-ippm-initial-registry].

Note that private registries following the format described here SHOULD use the prefix "Priv\_" on any name to avoid unintended conflicts (further considerations are described in section 10). Private registry entries usually have no specifying RFC, thus the Spec: element has no clear interpretation.

#### 7.1.3. URI

The URIs column MUST contain a URL [RFC3986] that uniquely identifies and locates the metric entry so it is accessible through the Internet. The URL points to a file containing all the human-readable information for one registry entry. The URL SHALL reference a target file that is preferably HTML-formatted and contains URLs to referenced sections of HTML-ized RFCs, or other reference specifications. These target files for different entries can be more easily edited and re-used when preparing new entries. The exact form of the URL for each target file, and the target file itself, will be determined by IANA and reside on "iana.org". The major sections of [I-D.ietf-ippm-initial-registry] provide an example of a target file in HTML form (sections 4 and higher).

#### 7.1.4. Description

A Registered Performance Metric description is a written representation of a particular Performance Metrics Registry entry. It supplements the Registered Performance Metric name to help Performance Metrics Registry users select relevant Registered Performance Metrics.

#### 7.1.5. Reference

This entry gives the specification containing the candidate registry entry which was reviewed and agreed, if such an RFC or other specification exists.

#### 7.1.6. Change Controller

This entry names the entity responsible for approving revisions to the registry entry, and SHALL provide contact information (for an individual, where appropriate).

#### 7.1.7. Version (of Registry Format)

This entry gives the version number for the registry format used. Formats complying with this memo MUST use 1.0. The version number SHALL NOT change unless a new RFC is published that changes the registry format. The version number of registry entries SHALL NOT change unless the registry entry is updated (following procedures in section 8).

#### 7.2. Metric Definition Category

This category includes columns to prompt all necessary details related to the metric definition, including the immutable document reference and values of input factors, called fixed parameters, which are left open in the immutable document, but have a particular value defined by the performance metric.

##### 7.2.1. Reference Definition

This entry provides a reference (or references) to the relevant section(s) of the document(s) that define the metric, as well as any supplemental information needed to ensure an unambiguous definition for implementations. The reference needs to be an immutable document, such as an RFC; for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification.

##### 7.2.2. Fixed Parameters

Fixed Parameters are Parameters whose value must be specified in the Performance Metrics Registry. The measurement system uses these values.

Where referenced metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Fixed Parameters. As an example for active metrics, Fixed Parameters determine most or all of the IPPM Framework convention "packets of Type-P" as described in [RFC2330], such as transport protocol, payload length, TTL, etc. An example for passive metrics is for RTP packet loss calculation that relies on the validation of a packet as RTP which is a multi-packet validation controlled by MIN\_SEQUENTIAL as defined by [RFC3550]. Varying MIN\_SEQUENTIAL values can alter the loss report and this value could be set as a Fixed Parameter.

Parameters MUST have well-defined names. For human readers, the hanging indent style is preferred, and any Parameter names and

definitions that do not appear in the Reference Method Specification MUST appear in this column (or Run-time Parameters column).

Parameters MUST have a well-specified data format.

A Parameter which is a Fixed Parameter for one Performance Metrics Registry entry may be designated as a Run-time Parameter for another Performance Metrics Registry entry.

### 7.3. Method of Measurement Category

This category includes columns for references to relevant sections of the immutable document(s) and any supplemental information needed to ensure an unambiguous method for implementations.

#### 7.3.1. Reference Method

This entry provides references to relevant sections of immutable documents, such as RFC(s) (for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification) describing the method of measurement, as well as any supplemental information needed to ensure unambiguous interpretation for implementations referring to the immutable document text.

Specifically, this section should include pointers to pseudocode or actual code that could be used for an unambiguous implementation.

#### 7.3.2. Packet Stream Generation

This column applies to Performance Metrics that generate traffic as part of their Measurement Method, including but not necessarily limited to Active metrics. The generated traffic is referred as a stream and this column describes its characteristics.

Each entry for this column contains the following information:

- o Value: The name of the packet stream scheduling discipline
- o Reference: the specification where the parameters of the stream are defined

The packet generation stream may require parameters such as the average packet rate and distribution truncation value for streams with Poisson-distributed inter-packet sending times. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

The simplest example of stream specification is Singleton scheduling (see [RFC2330]), where a single atomic measurement is conducted. Each atomic measurement could consist of sending a single packet (such as a DNS request) or sending several packets (for example, to request a webpage). Other streams support a series of atomic measurements in a "sample", with a schedule defining the timing between each transmitted packet and subsequent measurement. Principally, two different streams are used in IPPM metrics, Poisson distributed as described in [RFC2330] and Periodic as described in [RFC3432]. Both Poisson and Periodic have their own unique parameters, and the relevant set of parameters names and values should be included either in the Fixed Parameters column or in the Run-time parameter column.

### 7.3.3. Traffic Filter

This column applies to Performance Metrics that observe packets flowing through (the device with) the measurement agent i.e. that is not necessarily addressed to the measurement agent. This includes but is not limited to Passive Metrics. The filter specifies the traffic that is measured. This includes protocol field values/ranges, such as address ranges, and flow or session identifiers.

The traffic filter itself depends on needs of the metric itself and a balance of an operator's measurement needs and a user's need for privacy. Mechanics for conveying the filter criteria might be the BPF (Berkley Packet Filter) or PSAMP [RFC5475] Property Match Filtering which reuses IPFIX [RFC7012]. An example BPF string for matching TCP/80 traffic to remote destination net 192.0.2.0/24 would be "dst net 192.0.2.0/24 and tcp dst port 80". More complex filter engines might be supported by the implementation that might allow for matching using Deep Packet Inspection (DPI) technology.

The traffic filter includes the following information:

Type: the type of traffic filter used, e.g. BPF, PSAMP, OpenFlow rule, etc. as defined by a normative reference

Value: the actual set of rules expressed

### 7.3.4. Sampling Distribution

The sampling distribution defines out of all the packets that match the traffic filter, which one of those are actually used for the measurement. One possibility is "all" which implies that all packets matching the Traffic filter are considered, but there may be other sampling strategies. It includes the following information:

Value: the name of the sampling distribution

Reference definition: pointer to the specification where the sampling distribution is properly defined.

The sampling distribution may require parameters. In case such parameters are needed, they should be included either in the Fixed parameter column or in the run time parameter column, depending on whether they will be fixed or will be an input for the metric.

Sampling and Filtering Techniques for IP Packet Selection are documented in the PSAMP (Packet Sampling) [RFC5475], while the Framework for Packet Selection and Reporting, [RFC5474] provides more background information. The sampling distribution parameters might be expressed in terms of the Information Model for Packet Sampling Exports, [RFC5477], and the Flow Selection Techniques, [RFC7014].

#### 7.3.5. Run-time Parameters

Run-Time Parameters are Parameters that must be determined, configured into the measurement system, and reported with the results for the context to be complete. However, the values of these parameters is not specified in the Performance Metrics Registry (like the Fixed Parameters), rather these parameters are listed as an aid to the measurement system implementer or user (they must be left as variables, and supplied on execution).

Where metrics supply a list of Parameters as part of their descriptive template, a sub-set of the Parameters will be designated as Run-Time Parameters.

Parameters MUST have well defined names. For human readers, the hanging indent style is preferred, and the names and definitions that do not appear in the Reference Method Specification MUST appear in this column.

A Data Format for each Run-time Parameter MUST be specified in this column, to simplify the control and implementation of measurement devices. For example, parameters that include an IPv4 address can be encoded as a 32 bit integer (i.e. binary base64 encoded value) or ip-address as defined in [RFC6991]. The actual encoding(s) used must be explicitly defined for each Run-time parameter. IPv6 addresses and options MUST be accommodated, allowing Registered Metrics to be used in that address family. Other address families are permissable.

Examples of Run-time Parameters include IP addresses, measurement point designations, start times and end times for measurement, and other information essential to the method of measurement.

#### 7.3.6. Role

In some methods of measurement, there may be several roles defined, e.g., for a one-way packet delay active measurement there is one measurement agent that generates the packets and another agent that receives the packets. This column contains the name of the Role(s) for this particular entry. In the one-way delay example above, there should be two entries in the Role registry column, one for each Role (Source and Destination). When a measurement agent is instructed to perform the "Source" Role for one-way delay metric, the agent knows that it is required to generate packets. The values for this field are defined in the reference method of measurement (and this frequently results in abbreviated role names such as "Src").

When the Role column of a registry entry defines more than one Role, then the Role SHALL be treated as a Run-time Parameter and supplied for execution. It should be noted that the LMAP framework [RFC7594] distinguishes the Role from other Run-time Parameters, and defines a special parameter "Roles" inside the registry-grouping function list in the LMAP YANG model[RFC8194].

#### 7.4. Output Category

For entries which involve a stream and many singleton measurements, a statistic may be specified in this column to summarize the results to a single value. If the complete set of measured singletons is output, this will be specified here.

Some metrics embed one specific statistic in the reference metric definition, while others allow several output types or statistics.

##### 7.4.1. Type

This column contains the name of the output type. The output type defines a single type of result that the metric produces. It can be the raw results (packet send times and singleton metrics), or it can be a summary statistic. The specification of the output type MUST define the format of the output. In some systems, format specifications will simplify both measurement implementation and collection/storage tasks. Note that if two different statistics are required from a single measurement (for example, both "Xth percentile mean" and "Raw"), then a new output type must be defined ("Xth percentile mean AND Raw"). See the Naming section above for a list of Output Types.

#### 7.4.2. Reference Definition

This column contains a pointer to the specification(s) where the output type and format are defined.

#### 7.4.3. Metric Units

The measured results must be expressed using some standard dimension or units of measure. This column provides the units.

When a sample of singletons (see Section 11 of [RFC2330] for definitions of these terms) is collected, this entry will specify the units for each measured value.

#### 7.4.4. Calibration

Some specifications for Methods of Measurement include the possibility to perform an error calibration. Section 3.7.3 of [RFC7679] is one example. In the registry entry, this field will identify a method of calibration for the metric, and when available, the measurement system SHOULD perform the calibration when requested and produce the output with an indication that it is the result of a calibration method. In-situ calibration could be enabled with an internal loopback that includes as much of the measurement system as possible, performs address manipulation as needed, and provides some form of isolation (e.g., deterministic delay) to avoid send-receive interface contention. Some portion of the random and systematic error can be characterized this way.

For one-way delay measurements, the error calibration must include an assessment of the internal clock synchronization with its external reference (this internal clock is supplying timestamps for measurement). In practice, the time offsets of clocks at both the source and destination are needed to estimate the systematic error due to imperfect clock synchronization (the time offsets are smoothed, thus the random variation is not usually represented in the results).

Both internal loopback calibration and clock synchronization can be used to estimate the \*available accuracy\* of the Output Metric Units. For example, repeated loopback delay measurements will reveal the portion of the Output result resolution which is the result of system noise, and thus inaccurate.



## 7.5. Administrative information

### 7.5.1. Status

The status of the specification of this Registered Performance Metric. Allowed values are 'current' and 'deprecated'. All newly defined Information Elements have 'current' status.

### 7.5.2. Requester

The requester for the Registered Performance Metric. The requester MAY be a document, such as RFC, or person.

### 7.5.3. Revision

The revision number of a Registered Performance Metric, starting at 0 for Registered Performance Metrics at time of definition and incremented by one for each revision.

### 7.5.4. Revision Date

The date of acceptance or the most recent revision for the Registered Performance Metric. The date SHALL be determined by IANA and the reviewing Performance Metrics Expert.

## 7.6. Comments and Remarks

Besides providing additional details which do not appear in other categories, this open Category (single column) allows for unforeseen issues to be addressed by simply updating this informational entry.

## 8. Processes for Managing the Performance Metric Registry Group

Once a Performance Metric or set of Performance Metrics has been identified for a given application, candidate Performance Metrics Registry entry specifications prepared in accordance with Section 7 should be submitted to IANA to follow the process for review by the Performance Metric Experts, as defined below. This process is also used for other changes to the Performance Metrics Registry, such as deprecation or revision, as described later in this section.

It is desirable that the author(s) of a candidate Performance Metrics Registry entry seek review in the relevant IETF working group, or offer the opportunity for review on the working group mailing list.

### 8.1. Adding new Performance Metrics to the Performance Metrics Registry

Requests to add Registered Performance Metrics in the Performance Metrics Registry SHALL be submitted to IANA, which forwards the request to a designated group of experts (Performance Metric Experts) appointed by the IESG; these are the reviewers called for by the Specification Required [RFC8126] policy defined for the Performance Metrics Registry. The Performance Metric Experts review the request for such things as compliance with this document, compliance with other applicable Performance Metric-related RFCs, and consistency with the currently defined set of Registered Performance Metrics. The most efficient path for submission begins with preparation of an Internet Draft containing the proposed Performance Metrics Registry entry using the template in Section 11, so that the submission formatting will benefit from the normal IETF Internet Draft submission processing (including HTML-ization).

Submission to IANA may be during IESG review (leading to IETF Standards Action), where an Internet Draft proposes one or more Registered Performance Metrics to be added to the Performance Metrics Registry, including the text of the proposed Registered Performance Metric(s).

If an RFC-to-be includes a Performance Metric and a proposed Performance Metrics Registry entry, but the Performance Metric Expert review determines that one or more of the Section 5 criteria have not been met, then the proposed Performance Metrics Registry entry MUST be removed from the text. Once evidence exists that the Performance Metric meets the criteria in section 5, the proposed Performance Metrics Registry entry SHOULD be submitted to IANA to be evaluated in consultation with the Performance Metric Experts for registration at that time.

Authors of proposed Registered Performance Metrics SHOULD review compliance with the specifications in this document to check their submissions before sending them to IANA.

At least one Performance Metric Expert should endeavor to complete referred reviews in a timely manner. If the request is acceptable, the Performance Metric Experts signify their approval to IANA, and IANA updates the Performance Metrics Registry. If the request is not acceptable, the Performance Metric Experts MAY coordinate with the requester to change the request to be compliant, otherwise IANA SHALL coordinate resolution of issues on behalf of the expert. The Performance Metric Experts MAY choose to reject clearly frivolous or inappropriate change requests outright, but such exceptional circumstances should be rare.

This process should not in any way be construed as allowing the Performance Metric Experts to overrule IETF consensus. Specifically, any Registered Performance Metrics that were added to the Performance Metrics Registry with IETF consensus require IETF consensus for revision or deprecation.

Decisions by the Performance Metric Experts may be appealed as in Section 7 of [RFC8126].

## 8.2. Revising Registered Performance Metrics

A request for Revision is only permitted when the requested changes maintain backward-compatibility with implementations of the prior Performance Metrics Registry entry describing a Registered Performance Metric (entries with lower revision numbers, but the same Identifier and Name).

The purpose of the Status field in the Performance Metrics Registry is to indicate whether the entry for a Registered Performance Metric is 'current' or 'deprecated'.

In addition, no policy is defined for revising the Performance Metric entries in the IANA Registry or addressing errors therein. To be clear, changes and deprecations within the Performance Metrics Registry are not encouraged, and should be avoided to the extent possible. However, in recognition that change is inevitable, the provisions of this section address the need for revisions.

Revisions are initiated by sending a candidate Registered Performance Metric definition to IANA, as in Section 8.1, identifying the existing Performance Metrics Registry entry, and explaining how and why the existing entry should be revised.

The primary requirement in the definition of procedures for managing changes to existing Registered Performance Metrics is avoidance of measurement interoperability problems; the Performance Metric Experts must work to maintain interoperability above all else. Changes to Registered Performance Metrics may only be done in an interoperable way; necessary changes that cannot be done in a way to allow interoperability with unchanged implementations MUST result in the creation of a new Registered Performance Metric (with a new Name, replacing the RFCXXXXsecY portion of the name) and possibly the deprecation of the earlier metric.

A change to a Registered Performance Metric SHALL be determined to be backward-compatible when:

1. it involves the correction of an error that is obviously only editorial; or
2. it corrects an ambiguity in the Registered Performance Metric's definition, which itself leads to issues severe enough to prevent the Registered Performance Metric's usage as originally defined; or
3. it corrects missing information in the metric definition without changing its meaning (e.g., the explicit definition of 'quantity' semantics for numeric fields without a Data Type Semantics value); or
4. it harmonizes with an external reference that was itself corrected.

If a Performance Metric revision is deemed permissible and backward-compatible by the Performance Metric Experts, according to the rules in this document, IANA SHOULD execute the change(s) in the Performance Metrics Registry. The requester of the change is appended to the original requester in the Performance Metrics Registry. The Name of the revised Registered Performance Metric, including the RFCXXXsecY portion of the name, SHALL remain unchanged (even when the change is the result of IETF Standards Action; the revised registry entry SHOULD reference the new immutable document, such as an RFC or for other standards bodies, it is likely to be necessary to reference a specific, dated version of a specification, in an appropriate category and column).

Each Registered Performance Metric in the Performance Metrics Registry has a revision number, starting at zero. Each change to a Registered Performance Metric following this process increments the revision number by one.

When a revised Registered Performance Metric is accepted into the Performance Metrics Registry, the date of acceptance of the most recent revision is placed into the revision Date column of the registry for that Registered Performance Metric.

Where applicable, additions to Registered Performance Metrics in the form of text Comments or Remarks should include the date, but such additions may not constitute a revision according to this process.

Older version(s) of the updated metric entries are kept in the registry for archival purposes. The older entries are kept with all fields unmodified (version, revision date) except for the status field that SHALL be changed to "Deprecated".

### 8.3. Deprecating Registered Performance Metrics

Changes that are not permissible by the above criteria for Registered Performance Metric's revision may only be handled by deprecation. A Registered Performance Metric MAY be deprecated and replaced when:

1. the Registered Performance Metric definition has an error or shortcoming that cannot be permissibly changed as in Section 8.2 Revising Registered Performance Metrics; or
2. the deprecation harmonizes with an external reference that was itself deprecated through that reference's accepted deprecation method.

A request for deprecation is sent to IANA, which passes it to the Performance Metric Experts for review. When deprecating an Performance Metric, the Performance Metric description in the Performance Metrics Registry must be updated to explain the deprecation, as well as to refer to any new Performance Metrics created to replace the deprecated Performance Metric.

The revision number of a Registered Performance Metric is incremented upon deprecation, and the revision Date updated, as with any revision.

The intentional use of deprecated Registered Performance Metrics should result in a log entry or human-readable warning by the respective application.

Names and Metric IDs of deprecated Registered Performance Metrics must not be reused.

The deprecated entries are kept with all fields unmodified, except the version, revision date, and the status field (changed to "Deprecated").

## 9. Security considerations

This draft defines a registry structure, and does not itself introduce any new security considerations for the Internet. The definition of Performance Metrics for this registry may introduce some security concerns, but the mandatory references should have their own considerations for security, and such definitions should be reviewed with security in mind if the security considerations are not covered by one or more reference standards.

The aggregated results of the performance metrics described in this registry might reveal network topology information that may be

considered sensitive. If such cases are found, then access control mechanisms should be applied.

## 10. IANA Considerations

With the background and processes described in earlier sections, this document requests the following IANA Actions.

Editor's Note: Mock-ups of the implementation of this set of requests have been prepared with IANA's help during development of this memo, and have been captured in the Proceedings of IPPM working group sessions. IANA is currently preparing a mock-up. A recent version is available here: <http://encrypted.net/IETFMetricsRegistry-106.html>

### 10.1. Registry Group

The new registry group SHALL be named, "PERFORMANCE METRICS Group".

Registration Procedure: Specification Required

Reference: <This RFC>

Experts: Performance Metrics Experts

Note: TBD

### 10.2. Performance Metric Name Elements

This document specifies the procedure for Performance Metrics Name Element Registry setup. IANA is requested to create a new set of registries for Performance Metric Name Elements called "Registered Performance Metric Name Elements". Each Registry, whose names are listed below:

MetricType:

Method:

SubTypeMethod:

Spec:

Units:

Output:

will contain the current set of possibilities for Performance Metrics Registry Entry Names.

To populate the Registered Performance Metric Name Elements at creation, the IANA is asked to use the lists of values for each name element listed in Section 7.1.2. The Name Elements in each registry are case-sensitive.

When preparing a Metric entry for Registration, the developer SHOULD choose Name elements from among the registered elements. However, if the proposed metric is unique in a significant way, it may be necessary to propose a new Name element to properly describe the metric, as described below.

A candidate Metric Entry RFC or immutable document for IANA and Expert Review would propose one or more new element values required to describe the unique entry, and the new name element(s) would be reviewed along with the metric entry. New assignments for Registered Performance Metric Name Elements will be administered by IANA through Specification Required policy (which includes Expert Review) [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors.

### 10.3. New Performance Metrics Registry

This document specifies the procedure for Performance Metrics Registry setup. IANA is requested to create a new registry for Performance Metrics called "Performance Metrics Registry". This Registry will contain the following Summary columns:

Identifier:

Name:

URI:

Description:

Reference:

Change Controller:

Version:

Descriptions of these columns and additional information found in the template for registry entries (categories and columns) are further defined in section Section 7.

The Identifier 0 should be Reserved. The Registered Performance Metric unique identifier is an unbounded integer (range 0 to

infinity). The Identifier values from 64512 to 65536 are reserved for private or experimental use, and the user may encounter overlapping uses. When adding newly Registered Performance Metrics to the Performance Metrics Registry, IANA SHOULD assign the lowest available identifier to the new Registered Performance Metric. If a Performance Metrics Expert providing review determines that there is a reason to assign a specific numeric identifier, possibly leaving a temporary gap in the numbering, then the Performance Expert SHALL inform IANA of this decision.

Names starting with the prefix Priv\_ are reserved for private use, and are not considered for registration. The "Name" column entries are further defined in section Section 7.

The "URI" column will have a URL to the full template of each registry entry. The Registry Entry text SHALL be HTML-ized to aid the reader, with links to reference RFCs (similar to the way that Internet Drafts are HTML-ized, the same tool can perform the function) or immutable document.

The "Reference" column will include an RFC number, an approved specification designator from another standards body, or other immutable document.

New assignments for Performance Metrics Registry will be administered by IANA through Specification Required policy (which includes Expert Review) [RFC8126], i.e., review by one of a group of experts, the Performance Metric Experts, who are appointed by the IESG upon recommendation of the Transport Area Directors, or by Standards Action. The experts can be initially drawn from the Working Group Chairs, document editors, and members of the Performance Metrics Directorate, among other sources of experts.

Extensions of the Performance Metrics Registry require IETF Standards Action. Only one form of registry extension is envisaged:

1. Adding columns, or both categories and columns, to accommodate unanticipated aspects of new measurements and metric categories.

If the Performance Metrics Registry is extended in this way, the Version number of future entries complying with the extension SHALL be incremented (either in the unit or tenths digit, depending on the degree of extension).



## 11. Blank Registry Template

This section provides a blank template to help IANA and registry entry writers.

### 11.1. Summary

This category includes multiple indexes to the registry entry: the element ID and metric name.

#### 11.1.1. ID (Identifier)

<insert a numeric identifier, an integer, TBD>

#### 11.1.2. Name

<insert name according to metric naming convention>

#### 11.1.3. URI

URL: <https://www.iana.org/> ... <name>

#### 11.1.4. Description

<provide a description>

#### 11.1.5. Change Controller

#### 11.1.6. Version (of Registry Format)

### 11.2. Metric Definition

This category includes columns to prompt the entry of all necessary details related to the metric definition, including the immutable document reference and values of input factors, called fixed parameters.

#### 11.2.1. Reference Definition

<Full bibliographic reference to an immutable doc.>

<specific section reference and additional clarifications, if needed>

#### 11.2.2. Fixed Parameters

<list and specify Fixed Parameters, input factors that must be determined and embedded in the measurement system for use when needed>

### 11.3. Method of Measurement

This category includes columns for references to relevant sections of the immutable documents(s) and any supplemental information needed to ensure an unambiguous methods for implementations.

#### 11.3.1. Reference Method

<for metric, insert relevant section references and supplemental info>

#### 11.3.2. Packet Stream Generation

<list of generation parameters and section/spec references if needed>

#### 11.3.3. Traffic Filtering (observation) Details

The measured results based on a filtered version of the packets observed, and this section provides the filter details (when present).

<section reference>.

#### 11.3.4. Sampling Distribution

<insert time distribution details, or how this is diff from the filter>

#### 11.3.5. Run-time Parameters and Data Format

Run-time Parameters are input factors that must be determined, configured into the measurement system, and reported with the results for the context to be complete.

<list of run-time parameters, and their data formats>

#### 11.3.6. Roles

<lists the names of the different roles from the measurement method>

### 11.4. Output

This category specifies all details of the Output of measurements using the metric.

#### 11.4.1. Type

<insert name of the output type, raw or a selected summary statistic>

#### 11.4.2. Reference Definition

<describe the reference data format for each type of result>

#### 11.4.3. Metric Units

<insert units for the measured results, and the reference specification>.

#### 11.4.4. Calibration

<insert information on calibration>

#### 11.5. Administrative items

##### 11.5.1. Status

<current or deprecated>

##### 11.5.2. Requester

<name or RFC, etc.>

##### 11.5.3. Revision

<1.0>

##### 11.5.4. Revision Date

<format YYYY-MM-DD>

#### 11.6. Comments and Remarks

<Additional (Informational) details for this entry>

#### 12. Acknowledgments

Thanks to Brian Trammell and Bill Cervený, IPPM chairs, for leading some brainstorming sessions on this topic. Thanks to Barbara Stark and Juergen Schoenwaelder for the detailed feedback and suggestions. Thanks to Andrew McGregor for suggestions on metric naming. Thanks to Michelle Cotton for her early IANA review, and to Amanda Barber for answering questions related to the presentation of the registry and accessibility of the complete template via URL. Thanks to Roni

Even for his review and suggestions to generalize the procedures.  
Thanks to ~all the Area Directors for their reviews.

### 13. References

#### 13.1. Normative References

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [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>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [RFC6576] Geib, R., Ed., Morton, A., Fardid, R., and A. Steinmitz, "IP Performance Metrics (IPPM) Standard Advancement Testing", BCP 176, RFC 6576, DOI 10.17487/RFC6576, March 2012, <<https://www.rfc-editor.org/info/rfc6576>>.
- [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>>.
- [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>>.

## 13.2. Informative References

- [I-D.ietf-ippm-initial-registry]  
Morton, A., Bagnulo, M., Eardley, P., and K. D'Souza,  
"Initial Performance Metrics Registry Entries", draft-  
ietf-ippm-initial-registry-15 (work in progress), December  
2019.
- [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>>.
- [RFC3432] Raisanen, V., Grotefeld, G., and A. Morton, "Network  
performance measurement with periodic streams", RFC 3432,  
DOI 10.17487/RFC3432, November 2002,  
<<https://www.rfc-editor.org/info/rfc3432>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.  
Jacobson, "RTP: A Transport Protocol for Real-Time  
Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,  
July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed.,  
"RTP Control Protocol Extended Reports (RTCP XR)",  
RFC 3611, DOI 10.17487/RFC3611, November 2003,  
<<https://www.rfc-editor.org/info/rfc3611>>.
- [RFC4148] Stephan, E., "IP Performance Metrics (IPPM) Metrics  
Registry", BCP 108, RFC 4148, DOI 10.17487/RFC4148, August  
2005, <<https://www.rfc-editor.org/info/rfc4148>>.
- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A.,  
Grossglauser, M., and J. Rexford, "A Framework for Packet  
Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474,  
March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F.  
Raspall, "Sampling and Filtering Techniques for IP Packet  
Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009,  
<<https://www.rfc-editor.org/info/rfc5475>>.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G.  
Carle, "Information Model for Packet Sampling Exports",  
RFC 5477, DOI 10.17487/RFC5477, March 2009,  
<<https://www.rfc-editor.org/info/rfc5477>>.

- [RFC6035] Pendleton, A., Clark, A., Johnston, A., and H. Sinnreich, "Session Initiation Protocol Event Package for Voice Quality Reporting", RFC 6035, DOI 10.17487/RFC6035, November 2010, <<https://www.rfc-editor.org/info/rfc6035>>.
- [RFC6248] Morton, A., "RFC 4148 and the IP Performance Metrics (IPPM) Registry of Metrics Are Obsolete", RFC 6248, DOI 10.17487/RFC6248, April 2011, <<https://www.rfc-editor.org/info/rfc6248>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<https://www.rfc-editor.org/info/rfc7012>>.
- [RFC7014] D'Antonio, S., Zseby, T., Henke, C., and L. Peluso, "Flow Selection Techniques", RFC 7014, DOI 10.17487/RFC7014, September 2013, <<https://www.rfc-editor.org/info/rfc7014>>.
- [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>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <<https://www.rfc-editor.org/info/rfc8194>>.

Authors' Addresses

Marcelo Bagnulo  
Universidad Carlos III de Madrid  
Av. Universidad 30  
Leganes, Madrid 28911  
SPAIN

Phone: 34 91 6249500  
Email: marcelo@it.uc3m.es  
URI: <http://www.it.uc3m.es>

Benoit Claise  
Cisco Systems, Inc.  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

Email: [bclaise@cisco.com](mailto:bclaise@cisco.com)

Philip Eardley  
BT  
Adastral Park, Martlesham Heath  
Ipswich  
ENGLAND

Email: [philip.eardley@bt.com](mailto:philip.eardley@bt.com)

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

Email: [acmorton@att.com](mailto:acmorton@att.com)

Aamer Akhter  
Consultant  
118 Timber Hitch  
Cary, NC  
USA

Email: [aakhter@gmail.com](mailto:aakhter@gmail.com)

IPPM Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: September 24, 2020

G. Fioccola, Ed.  
Huawei Technologies  
M. Cociglio  
Telecom Italia  
A. Sapio  
R. Sisto  
Politecnico di Torino  
March 23, 2020

Multipoint Alternate Marking method for passive and hybrid performance  
monitoring  
draft-ietf-ippm-multipoint-alt-mark-09

Abstract

The Alternate Marking method, as presented in RFC 8321, can be applied only to point-to-point flows because it assumes that all the packets of the flow measured on one node are measured again by a single second node. This document generalizes and expands this methodology to measure any kind of unicast flows, whose packets can follow several different paths in the network, in wider terms a multipoint-to-multipoint network. For this reason the technique here described is called Multipoint Alternate Marking.

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 September 24, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.



This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
2.1. Correlation with RFC5644 . . . . .	5
3. Flow classification . . . . .	5
4. Multipoint Performance Measurement . . . . .	8
4.1. Monitoring Network . . . . .	8
5. Multipoint Packet Loss . . . . .	10
6. Network Clustering . . . . .	11
6.1. Algorithm for Cluster partition . . . . .	11
7. Timing Aspects . . . . .	15
8. Multipoint Delay and Delay Variation . . . . .	17
8.1. Delay measurements on multipoint paths basis . . . . .	17
8.1.1. Single Marking measurement . . . . .	17
8.2. Delay measurements on single packets basis . . . . .	17
8.2.1. Single and Double Marking measurement . . . . .	17
8.2.2. Hashing selection method . . . . .	18
9. A Closed Loop Performance Management approach . . . . .	20
10. Examples of application . . . . .	21
11. Security Considerations . . . . .	22
12. Acknowledgements . . . . .	22
13. IANA Considerations . . . . .	22
14. References . . . . .	22
14.1. Normative References . . . . .	22
14.2. Informative References . . . . .	23
Authors' Addresses . . . . .	24

## 1. Introduction

The Alternate Marking method, as described in RFC 8321 [RFC8321], is applicable to a point-to-point path. The extension proposed in this document applies to the most general case of multipoint-to-multipoint path and enables flexible and adaptive performance measurements in a managed network.

The Alternate Marking methodology described in RFC 8321 [RFC8321] allows the synchronization of the measurements in different points by

dividing the packet flow into batches. So it is possible to get coherent counters and show what is happening in every marking period for each monitored flow. The monitoring parameters are the packet counter and timestamps of a flow for each marking period. Note that additional details about the applicability of the Alternate Marking methodology are described both in RFC 8321 [RFC8321] and in the paper [IEEE-Network-PNPM].

There are some applications of the Alternate Marking method where there are a lot of monitored flows and nodes. Multipoint Alternate Marking aims to reduce these values and makes the performance monitoring more flexible in case a detailed analysis is not needed. For instance, by considering  $n$  measurement points and  $m$  monitored flows, the order of magnitude of the packet counters for each time interval is  $n*m*2$  (1 per color). The number of measurement points and monitored flows may vary and depends on the portion of the network we are monitoring (core network, metro network, access network) and on the granularity (for each service, each customer). So if both  $n$  and  $m$  are high values the packet counters increase a lot and Multipoint Alternate Marking offers a tool to control these parameters.

The approach presented in this document is applied only to unicast flows and not to multicast. Broadcast, Unknown-unicast, and Multicast (BUM) traffic is not considered here, because traffic replication is not covered by the Multipoint Alternate Marking method. Furthermore it can be applicable to anycast flows and Equal-Cost MultiPath (ECMP) paths can also be easily monitored with this technique.

In short, RFC 8321 [RFC8321] applies to point-to-point unicast flows and BUM traffic while this document and its Clustered Alternate Marking method is valid for multipoint-to-multipoint unicast flows, anycast and ECMP flows.

The Alternate Marking method can therefore be extended to any kind of multipoint to multipoint paths, and the network clustering approach presented in this document is the formalization of how to implement this property and allow a flexible and optimized performance measurement support for network management in every situation.

Without network clustering, it is possible to apply Alternate Marking only for all the network or per single flow. Instead, with network clustering, it is possible to use the partition of the network into clusters at different levels in order to perform the needed degree of detail. In some circumstances it is possible to monitor a Multipoint Network by analysing the Network Clustering, without examining in depth. In case of problems (packet loss is measured or the delay is

too high) the filtering criteria could be specified more in order to perform a detailed analysis by using a different combination of clusters up to a per-flow measurement as described in RFC 8321 [RFC8321].

This approach fits very well with the Closed Loop Network and Software Defined Network (SDN) paradigm where the SDN Orchestrator and the SDN Controllers are the brains of the network and can manage flow control to the switches and routers and, in the same way, can calibrate the performance measurements depending on the desired accuracy. An SDN Controller Application can orchestrate how accurate the network performance monitoring is setup by applying the Multipoint Alternate Marking as described in this document.

It is important to underline that, as extension of RFC 8321 [RFC8321], this is a methodology draft, so the mechanism that can be used to transmit the counters and the timestamps is out of scope here and the implementation is open. Several options are possible, e.g. [I-D.zhou-ippm-enhanced-alternate-marking].

Note that, as for RFC 8321 [RFC8321], the fragmented packets case can be managed with this methodology if fragmentation happens outside the portion of the monitored network.

## 2. Terminology

The definitions of the basic terms are identical to those found in Alternate Marking (RFC 8321 [RFC8321]). It is to be remembered that RFC 8321 [RFC8321] is valid for point-to-point unicast flows and BUM traffic.

The important new terms that need to be explained are listed below:

**Multipoint Alternate Marking:** Extension to RFC 8321 [RFC8321], valid for multipoint-to-multipoint unicast flows, anycast and ECMP flows. It can also be referred as Clustered Alternate Marking;

**Flow definition:** The concept of flow is generalized in this document. The identification fields are selected without any constraints and, in general, the flow can be a multipoint-to-multipoint flow, as a result of aggregate point-to-point flows;

**Monitoring Network:** it is identified with the nodes of the network that are the measurement points (MPs) and the links that are the connections between MPs. The Monitoring Network graph depends on the flow definition, so it can represent a specific flow or the the entire network topology as aggregate of all the flows;

Cluster: smallest identifiable subnetwork of the entire Monitoring Network graph that still satisfies the condition that the number of packets that goes in is the same that goes out;

Multipoint metrics: packet loss, delay and delay variation are extended to the case of multipoint flows. It is possible to compute these metrics on multipoint paths basis in order to associate the measurements to a cluster, to a combination of clusters or to the entire monitored network. For delay and delay variation, it is also possible to define the metrics on a single packet basis and it means that the multipoint path is used to easily couple packets between input and output nodes of a multipoint path.

The next section highlights the correlation with the terms used in RFC 5644 [RFC5644].

### 2.1. Correlation with RFC5644

RFC 5644 [RFC5644] is limited to active measurements using a single source packet or stream, and observations of corresponding packets along the path (spatial), at one or more destinations (one-to-group), or both.

Instead, the scope of this memo is to define multiparty metrics for passive and hybrid measurements in a group-to-group topology with multiple sources and destinations.

RFC 5644 [RFC5644] introduces metric names that can be reused also here but have to be extended and rephrased to be applied to the Alternate Marking schema:

- a. the multiparty metrics are not only one-to-group metrics but can be also group-to-group metrics;
- b. the spatial metrics, used for measuring the performance of segments of a source to destination path, are applied here to group-to-group segments (called Clusters).

### 3. Flow classification

An unicast flow is identified by all the packets having a set of common characteristics. This definition is inspired by RFC 7011 [RFC7011].

As an example, by considering a flow as all the packets sharing the same source IP address or the same destination IP address, it is easy to understand that the resulting pattern will not be a point-to-point

connection, but a point-to-multipoint or multipoint-to-point connection.

In general a flow can be defined by a set of selection rules used to match a subset of the packets processed by the network device. These rules specify a set of layer-3 and layer-4 headers fields (Identification Fields) and the relative values that must be found in matching packets.

The choice of the identification fields directly affects the type of paths that the flow would follow in the network. In fact, it is possible to relate a set of identification fields with the pattern of the resulting graphs, as listed in Figure 1.

A TCP 5-tuple usually identifies flows following either a single path or a point-to-point multipath (in case of load balancing). On the contrary, a single source address selects aggregate flows following a point-to-multipoint, while a multipoint-to-point can be the result of a matching on a single destination address. In case a selection rule and its reverse are used for bidirectional measurements, they can correspond to a point-to-multipoint in one direction and a multipoint-to-point in the opposite direction.

So the flows to be monitored are selected into the monitoring points using packet selection rules, that can also change the pattern of the monitored network.

Note that, more in general, the flow can be defined at different levels based on the encapsulation considered and additional conditions that are not in the packet header can also be included as part of matching criteria.

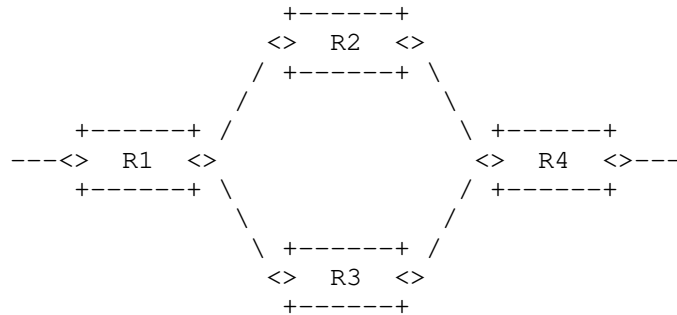
The Alternate Marking method is applicable only to a single path (and partially to a one-to-one multipath), so the extension proposed in this document is suitable also for the most general case of multipoint-to-multipoint, which embraces all the other patterns of Figure 1.

```

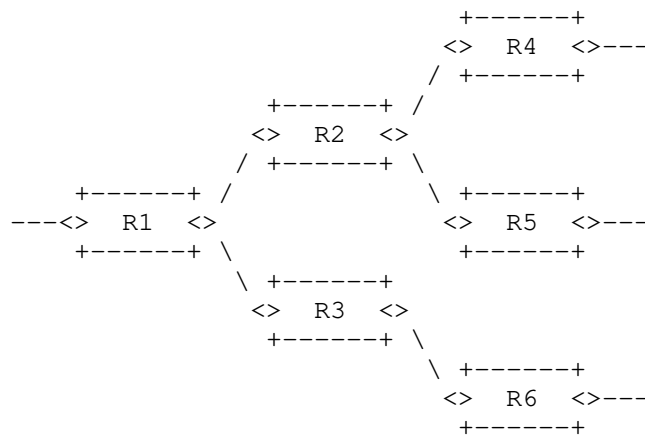
point-to-point single path
  +-----+      +-----+      +-----+
---<>  R1  <>---<>  R2  <>---<>  R3  <>---
  +-----+      +-----+      +-----+

```

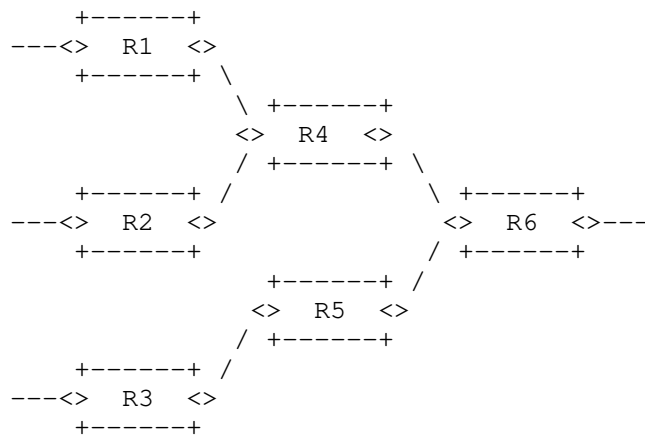
## point-to-point multipath



## point-to-multipoint



## multipoint-to-point



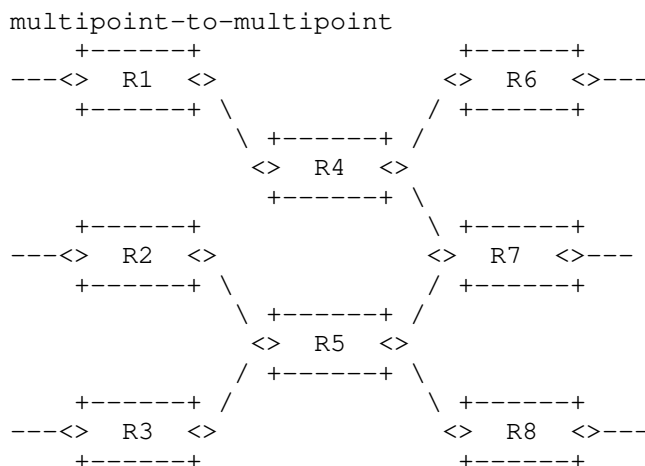


Figure 1: Flow classification

The case of unicast flow is considered in the previous figure. Anyway the anycast flow is also in scope because there is no replication and only a single node from the anycast group receives the traffic, so it can be viewed as a special case of unicast flow. Furthermore, an ECMP flow is in scope by definition, since it is a point-to-multipoint unicast flow.

#### 4. Multipoint Performance Measurement

By Using the Alternate Marking method only point-to-point paths can be monitored. To have an IP (TCP/UDP) flow that follows a point-to-point path we have to define, with a specific value, 5 identification fields (IP Source, IP Destination, Transport Protocol, Source Port, Destination Port).

Multipoint Alternate Marking enables the performance measurement for multipoint flows selected by identification fields without any constraints (even the entire network production traffic). It is also possible to use multiple marking points for the same monitored flow.

##### 4.1. Monitoring Network

The Monitoring Network is deduced from the Production Network, by identifying the nodes of the graph that are the measurement points, and the links that are the connections between measurement points.

There are some techniques that can help with the building of the monitoring network (as an example it is possible to mention

[I-D.ietf-ippm-route]). In general there are different options: the monitoring network can be obtained by considering all the possible paths for the traffic or also by periodically checking the traffic (e.g. daily, weekly, monthly) and update the graph as appropriate, but this is up to the Network Management System (NMS) configuration.

So a graph model of the monitoring network can be built according to the Alternate Marking method: the monitored interfaces and links are identified. Only the measurement points and links where the traffic has flowed have to be represented in the graph.

The following figure shows a simple example of a Monitoring Network graph:

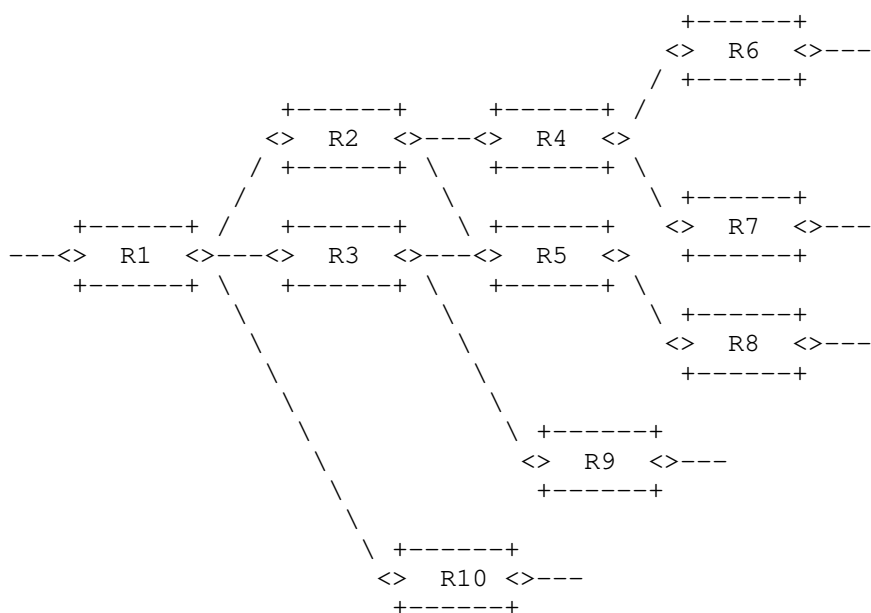


Figure 2: Monitoring Network Graph

Each monitoring point is characterized by the packet counter that refers only to a marking period of the monitored flow.

The same is applicable also for the delay but it will be described in the following sections.



## 5. Multipoint Packet Loss

Since all the packets of the considered flow leaving the network have previously entered the network, the number of packets counted by all the input nodes is always greater or equal than the number of packets counted by all the output nodes. Non-initial fragments are not considered here.

The assumption is the use of the Alternate Marking method. And in case of no packet loss occurring in the marking period, if all the input and output points of the network domain to be monitored are measurement points, the sum of the number of packets on all the ingress interfaces equals the number on egress interfaces for the monitored flow. In this circumstance, if no packet loss occurs, the intermediate measurement points have only the task to split the measurement.

It is possible to define the Network Packet Loss of one monitored flow for a single period: <<In a packet network, the number of lost packets is the number of packets counted by the input nodes minus the number of packets counted by the output nodes>>. This is true for every packet flow in each marking period.

The Monitored Network Packet Loss with n input nodes and m output nodes is given by:

$$PL = (PI1 + PI2 + \dots + PIn) - (PO1 + PO2 + \dots + POm)$$

where:

PL is the Network Packet Loss (number of lost packets)

PIi is the Number of packets flowed through the i-th Input node in this period

POj is the Number of packets flowed through the j-th Output node in this period

The equation is applied on a per-time-interval basis and on an per-flow basis:

The reference interval is the Alternate Marking period as defined in RFC 8321 [RFC8321].

The flow definition is generalized here, indeed, as described before, a multipoint packet flow is considered and the identification fields can be selected without any constraints.

## 6. Network Clustering

The previous Equation can determine the number of packets lost globally in the monitored network, exploiting only the data provided by the counters in the input and output nodes.

In addition it is also possible to leverage the data provided by the other counters in the network to converge on the smallest identifiable subnetworks where the losses occur. These subnetworks are named Clusters.

A Cluster graph is a subnetwork of the entire Monitoring Network graph that still satisfies the packet loss equation (introduced in the previous section) where PL in this case is the number of packets lost in the Cluster. As for the entire Monitoring Network graph, the Cluster is defined on a per-flow basis.

For this reason a Cluster should contain all the arcs emanating from its input nodes and all the arcs terminating at its output nodes. This ensures that we can count all the packets (and only those) exiting an input node again at the output node, whatever path they follow.

In a completely monitored unidirectional network (a network where every network interface is monitored), each network device corresponds to a Cluster and each physical link corresponds to two Clusters (one for each device).

Clusters can have different sizes depending on flow filtering criteria adopted.

Moreover, sometimes Clusters can be optionally simplified. For example when two monitored interfaces are divided by a single router (one is the input interface and the other is the output interface and the router has only these two interfaces), instead of counting exactly twice, upon entering and leaving, it is possible to consider a single measurement point (in this case we do not care of the internal packet loss of the router).

It is worth highlighting that it might also be convenient to define Clusters based on the topological information and applicable to all the possible flows in the monitored network.

### 6.1. Algorithm for Cluster partition

A simple algorithm can be applied in order to split our monitoring network into Clusters. This can be done for each direction separately. The Cluster partition is based on the Monitoring Network

Graph that can be valid for a specific flow or can also be general and valid for the entire network topology.

It is a two-step algorithm:

- o Group the links where there is the same starting node;
- o Join the grouped links with at least one ending node in common.

Considering that the links are unidirectional, the first step implies to list all the links as connection between two nodes and to group the different links if they have the same starting node. Note that it is possible to start from any link and the procedure works anyway. Following this classification, the second step implies to eventually join the groups classified in the first step by looking at the ending nodes. If different groups have at least one common ending node, they are put together and belong to the same set. After the application of the two steps of the algorithm, each one of the composed sets of links together with the endpoint nodes constitutes a Cluster.

In our monitoring network graph example it is possible to identify the Clusters partition by applying this two-step algorithm.

The first step identifies the following groups:

1. Group 1: (R1-R2), (R1-R3), (R1-R10)
2. Group 2: (R2-R4), (R2-R5)
3. Group 3: (R3-R5), (R3-R9)
4. Group 4: (R4-R6), (R4-R7)
5. Group 5: (R5-R8)

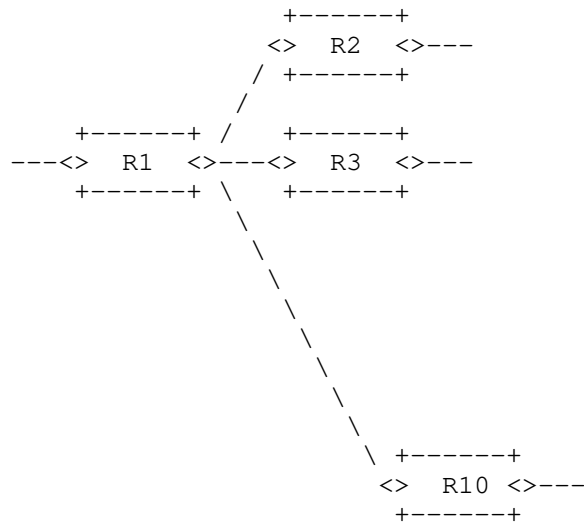
And then, the second step builds the Clusters partition (in particular we can underline that Group 2 and Group 3 connect together, since R5 is in common):

1. Cluster 1: (R1-R2), (R1-R3), (R1-R10)
2. Cluster 2: (R2-R4), (R2-R5), (R3-R5), (R3-R9)
3. Cluster 3: (R4-R6), (R4-R7)
4. Cluster 4: (R5-R8)

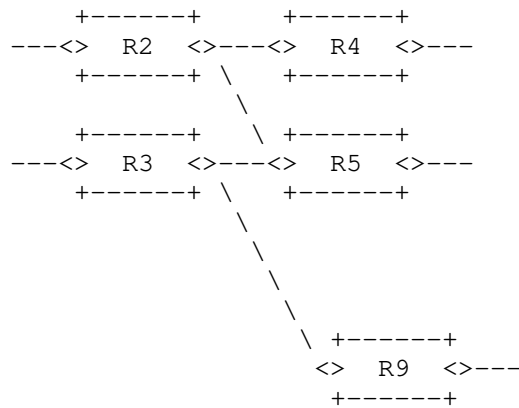
The flow direction here considered is from left to right. For the opposite direction the same way of reasoning can be applied and, in this example, you get the same Clusters partition.

In the end the following 4 Clusters are obtained:

Cluster 1



Cluster 2



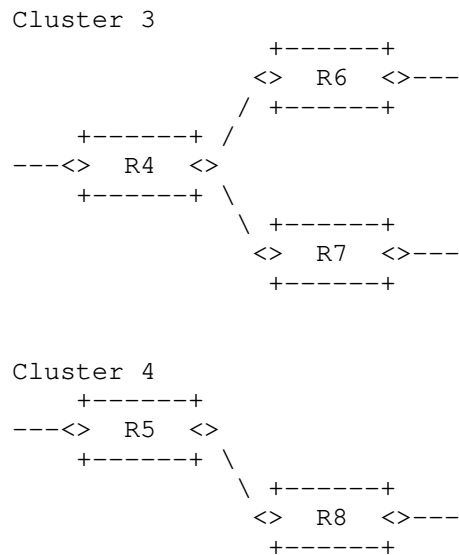


Figure 3: Clusters example

There are Clusters with more than 2 nodes and two-nodes Clusters. In the two-nodes Clusters the loss is on the link (Cluster 4). In more-than-2-nodes Clusters the loss is on the Cluster but we cannot know in which link (Cluster 1, 2, 3).

In this way the calculation of packet loss can be made on Cluster basis. Note that the packet counters for each marking period permit to calculate the packet rate on Cluster basis, so Committed Information Rate (CIR) and Excess Information Rate (EIR) could also be deduced on Cluster basis.

Obviously, by combining some Clusters in a new connected subnetwork (called Super Cluster) the Packet Loss Rule is still true.

In this way, in a very large network there is no need to configure detailed filter criteria to inspect the traffic. You can check a multipoint network and, in case of problems, you can go deep with a step-by-step cluster analysis, but only for the cluster or combination of clusters where the problem happens.

In summary, once defined a flow, the algorithm to build the Cluster Partition considers all the possible links and nodes crossed by the given flow, even if there is no traffic. It is based on topological information. So, if the flow does not enter or traverse all the nodes, the counters have a non-zero value for the involved nodes,

while a zero value for the other nodes without traffic, but, in the end all the formulas are still valid.

The algorithm described above is an Iterative clustering algorithm, but it is also possible to apply a Recursive clustering algorithm by using the node-node adjacency matrix representation ([IEEE-ACM-ToN-MPNPM]).

The complete and mathematical analysis of the possible Algorithms for Cluster partition, including the considerations in terms of efficiency and a comparison between the different methods, is in the paper [IEEE-ACM-ToN-MPNPM].

## 7. Timing Aspects

It is important to consider the timing aspects, since out of order packets happen and have to be handled as well as described in RFC 8321 [RFC8321]. But, in a multi-source situation an additional issue has to be considered. With multipoint path, the egress nodes will receive alternate marked packets in random order from different ingress nodes, and this must not affect the measurement.

So, if we analyse a multipoint-to-multipoint path with more than one marking node, it is important to recognize the reference measurement interval. In general the measurement interval for describing the results is the interval of the marking node that is more aligned with the start of the measurement, as reported in the following figure.

Note that the mark switching approach based on a fixed timer is considered in this document.

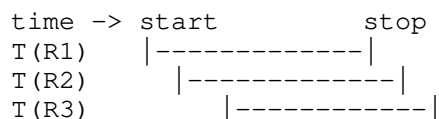


Figure 4: Measurement Interval

In the figure it is assumed that the node with the earliest clock (R1) identifies the right starting and ending time of the measurement, but it is just an assumption and other possibilities could occur. So, in this case, T(R1) is the measurement interval and its recognition is essential in order to be compatible and make comparison with other active/passive/hybrid Packet Loss metrics.

When we expand to multipoint-to-multipoint flows, we have to consider that all source nodes mark the traffic and this adds more complexity.

Regarding the timing aspects of the methodology, RFC 8321 [RFC8321] already describes two contributions that are taken into account: the clock error between network devices and the network delay between measurement points.

But we should now consider an additional contribution. Since all source nodes mark the traffic, the source measurement intervals can be of different lengths and with different offsets and this mismatch  $m$  can be added to  $d$ , as shown in figure.

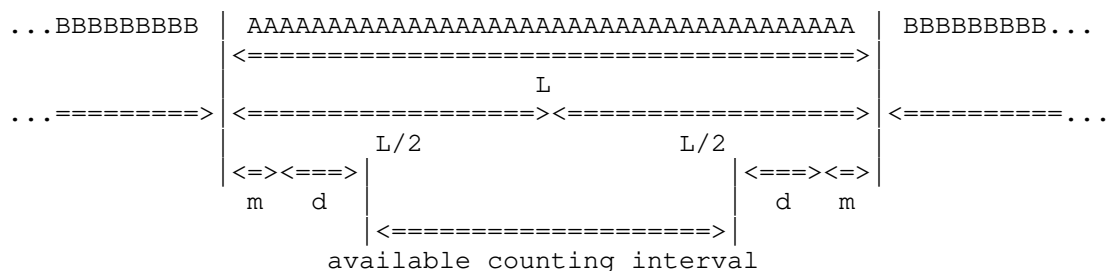


Figure 5: Timing Aspects for Multipoint paths

So the misalignment between the marking source routers gives an additional constraint and the value of  $m$  is added to  $d$  (that already includes clock error and network delay).

Thus, three different possible contributions are considered: clock error between network devices, network delay between measurement points and the misalignment between the marking source routers.

In the end, the condition that must be satisfied to enable the method to function properly is that the available counting interval must be  $> 0$ , and that means:

$$L - 2m - 2d > 0.$$

This formula needs to be verified for each measurement point on the multipoint path, where  $m$  is misalignment between the marking source routers, while  $d$ , already introduced in RFC 8321 [RFC8321], takes into account clock error and network delay between network nodes. Therefore, the mismatch between measurement intervals must satisfy this condition.

Note that the timing considerations are valid for both packet loss and delay measurements.

## 8. Multipoint Delay and Delay Variation

The same line of reasoning can be applied to Delay and Delay Variation. Similarly to the delay measurements defined in RFC 8321 [RFC8321], the marking batches anchor the samples to a particular period and this is the time reference that can be used. It is important to highlight that both delay and delay variation measurements make sense in a multipoint path. The Delay Variation is calculated by considering the same packets selected for measuring the Delay.

In general, it is possible to perform delay and delay variation measurements on multipoint paths basis or on single packets basis:

- o Delay measurements on multipoint paths basis means that the delay value is representative of an entire multipoint path (e.g. whole multipoint network, a cluster or a combination of clusters).
- o Delay measurements on a single packet basis means that you can use multipoint path just to easily couple packets between input and output nodes of a multipoint path, as it is described in the following sections.

### 8.1. Delay measurements on multipoint paths basis

#### 8.1.1. Single Marking measurement

Mean delay and mean delay variation measurements can also be generalized to the case of multipoint flows. It is possible to compute the average one-way delay of packets, in one block, in a cluster or in the entire monitored network.

The average latency can be measured as the difference between the weighted averages of the mean timestamps of the sets of output and input nodes. This means that, in the calculation, it is possible to weigh the timestamps by considering the number of packets for each endpoints.

### 8.2. Delay measurements on single packets basis

#### 8.2.1. Single and Double Marking measurement

Delay and delay variation measurements relative to only one picked packet per period (both single and double marked) can be performed in the Multipoint scenario with some limitations:



Single marking based on the first/last packet of the interval would not work, because it would not be possible to agree on the first packet of the interval.

Double marking or multiplexed marking would work, but each measurement would only give information about the delay of a single path. However, by repeating the measurement multiple times, it is possible to get information about all the paths in the multipoint flow. This can be done in case of point-to-multipoint path but it is more difficult to achieve in case of multipoint-to-multipoint path because of the multiple source routers.

If we would perform a delay measurement for more than one picked packet in the same marking period and, especially, if we want to get delay measurements on multipoint-to-multipoint basis, both single and double marking method are not useful in the Multipoint scenario, since they would not be representative of the entire flow. The packets can follow different paths with various delays, and in general it can be very difficult to recognize marked packets in a multipoint-to-multipoint path especially in the case when there is more than one per period.

A desirable option is to monitor simultaneously all the paths of a multipoint path in the same marking period and, for this purpose, hashing can be used as reported in the next Section.

#### 8.2.2. Hashing selection method

RFC 5474 [RFC5474] and RFC 5475 [RFC5475] introduce sampling and filtering techniques for IP Packet Selection.

The hash-based selection methodologies for delay measurement can work in a multipoint-to-multipoint path and can be used both coupled to mean delay or stand alone.

[I-D.mizrahi-ippm-compact-alternate-marking] introduces how to use the Hash method (RFC 5474 [RFC5474] and RFC 5475 [RFC5475]) combined with Alternate Marking method for point-to-point flows. It is also called Mixed Hashed Marking: the coupling of marking method and hashing technique is very useful because the marking batches anchor the samples selected with hashing and this simplifies the correlation of the hashing packets along the path.

It is possible to use a basic hash or a dynamic hash method. One of the challenges of the basic approach is that the frequency of the sampled packets may vary considerably. For this reason the dynamic approach has been introduced for point-to-point flow in order to have

the desired and almost fixed number of samples for each measurement period. In the hash-based sampling, Alternate Marking is used to create periods, so that hash-based samples are divided into batches, allowing to anchor the selected samples to their period. Moreover in the dynamic hash-based sampling, by dynamically adapting the length of the hash value, the number of samples is bounded in each marking period. This can be realized by choosing the maximum number of samples (NMAX) to be caught in a marking period. The algorithm starts with only few hash bits, that permit to select a greater percentage of packets (e.g. with 0 bit of hash all the packets are sampled, with 1 bit of hash half of the packets are sampled, and so on). When the number of selected packets reaches NMAX, a hashing bit is added. As a consequence, the sampling proceeds at half of the original rate and also the packets already selected that do not match the new hash are discarded. This step can be repeated iteratively. It is assumed that each sample includes the timestamp (used for delay measurement) and the hash value, allowing the management system to match the samples received from the two measurement points. The dynamic process statistically converges at the end of a marking period and the final number of selected samples is between NMAX/2 and NMAX. Therefore, the dynamic approach paces the sampling rate, allowing to bound the number of sampled packets per sampling period.

In a multipoint environment the behaviour is similar to a point-to-point flow. In particular, in the context of a multipoint-to-multipoint flow, the dynamic hash could be the solution to perform delay measurements on specific packets and to overcome the single and double marking limitations.

The management system receives the samples including the timestamps and the hash value from all the MPs, and this happens both for point-to-point and for multipoint-to-multipoint flows. Then the longest hash used by MPs is deduced and it is applied to couple timestamps of the same packets of 2 MPs of a point-to-point path or of input and output MPs of a Cluster (or a Super Cluster or the entire network). But some considerations are needed: if there isn't packet loss the set of input samples is always equal to the set of output samples. In case of packet loss the set of output samples can be a subset of input samples but the method still works because, at the end, it is easy to couple the input and output timestamps of each caught packet using the hash (in particular the "unused part of the hash" that should be different for each packet).

Therefore, the basic hash is logically similar to the double marking method, and in case of point-to-point path double marking and basic hash selection are equivalent. The dynamic approach scales the number of measurements per interval, and it would seem that double marking would also work well if we reduced the interval length, but

this can be done only for point-to-point path and not for multipoint path, where we cannot couple the picked packets in a multipoint paths. So, in general, if we want to get delay measurements on multipoint-to-multipoint path basis and want to select more than one packet per period, double marking cannot be used because we could not be able to couple the picked packets between input and output nodes. On the other hand we can do that by using hashing selection.

#### 9. A Closed Loop Performance Management approach

The Multipoint Alternate Marking framework that is introduced in this document adds flexibility to Performance Management (PM) because it can reduce the order of magnitude of the packet counters. This allows an SDN Orchestrator to supervise, control and manage PM in large networks.

The monitoring network can be considered as a whole or can be split in Clusters, that are the smallest subnetworks (group-to-group segments), maintaining the packet loss property for each subnetwork. They can also be combined in new connected subnetworks at different levels depending on the detail we want to achieve.

An SDN Controller or a Network Management System (NMS) can calibrate Performance Measurements since they are aware of the network topology. They can start without examining in depth. In case of necessity (packet loss is measured or the delay is too high), the filtering criteria could be immediately reconfigured in order to perform a partition of the network by using Clusters and/or different combinations of Clusters. In this way the problem can be localized in a specific Cluster or in a single combination of Clusters and a more detailed analysis can be performed step-by-step by successive approximation up to a point-to-point flow detailed analysis. This is the so called Closed Loop.

This approach can be called Network Zooming and can be performed in two different ways:

- 1) change the traffic filter and select more detailed flows;
- 2) activate new measurement points by defining more specified clusters.

The Network Zooming approach implies that the some filters or rules are changed and there is a transient time to wait once the new network configuration takes effect and it can be determined by the Network Orchestrator/Controller, based on the network conditions.

For example, if the Network Zooming identifies the performance problem for the traffic coming from a specific source, we need to recognize the marked signal from this specific source node and its relative path. For this purpose we can activate all the available measurement points and specify better the flow filter criteria (i.e. 5-tuple). As an alternative, it can be enough to select packets from the specific source for delay measurements, and in this case it is possible to apply the hashing technique as mentioned in the previous sections.

[I-D.song-opsawg-ifit-framework] defines an architecture where the centralized Data Collector and Network Management can apply the intelligent and flexible Alternate Marking algorithm as previously described.

As for RFC 8321 [RFC8321], it is possible to classify the traffic and mark a portion of the total traffic. For each period the packet rate and bandwidth are calculated from the number of packets. In this way the Network Orchestrator becomes aware if the traffic rate overcomes limits. In addition more precision can be obtained by reducing the marking period, indeed some implementations use a marking period of 1 sec and less.

In addition an SDN Controller could also collect the measurement history.

It is important to mention that the Multipoint Alternate Marking framework also helps Traffic Visualization. Indeed this methodology is very useful to identify which path or which cluster is crossed by the flow.

## 10. Examples of application

There are application fields where it may be useful to take into consideration the Multipoint Alternate Marking:

- o VPN: The IP traffic is selected on IP source basis in both directions. At the endpoint WAN interface all the output traffic is counted in a single flow. The input traffic is composed by all the other flows aggregated for source address. So, by considering  $n$  end-points, the monitored flows are  $n$  (each flow with 1 ingress point and  $(n-1)$  egress points) instead of  $n*(n-1)$  flows (each flow, with 1 ingress point and 1 egress point);
- o Mobile Backhaul: LTE traffic is selected, in the Up direction, by the ENodeB source address and, in Down direction, by the ENodeB destination address because the packets are sent from the Mobile

Packet Core to the EnodeB. So the monitored flow is only one per EnodeB in both directions;

- o Over The Top (OTT) services: The traffic is selected, in the Down direction by the source addresses of the packets sent by OTT Servers. In the opposite direction (Up) by the destination IP addresses of the same Servers. So the monitoring is based on a single flow per OTT Servers in both directions.
- o Enterprise SD-WAN: SD-WAN allows to connect remote branch offices to Data Centers and build higher-performance WANs. A centralized controller is used to set policies and prioritize traffic. The SD-WAN takes into account these policies and the availability of network bandwidth to route traffic. This helps ensure that application performance meets service level agreements (SLAs). This methodology can also help the path selection for the WAN connection based on per Cluster and per flow performance.

Note that the list is just an example and it is not exhaustive. More applications are possible.

## 11. Security Considerations

This document specifies a method to perform measurements that does not directly affect Internet security nor applications that run on the Internet. However, implementation of this method must be mindful of security and privacy concerns, as explained in RFC 8321 [RFC8321].

## 12. Acknowledgements

The authors would like to thank Al Morton, Tal Mizrahi, Rachel Huang for the precious contribution.

## 13. IANA Considerations

This memo makes no requests of IANA.

## 14. References

### 14.1. Normative References

- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.

- [RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <<https://www.rfc-editor.org/info/rfc5475>>.
- [RFC5644] Stephan, E., Liang, L., and A. Morton, "IP Performance Metrics (IPPM): Spatial and Multicast", RFC 5644, DOI 10.17487/RFC5644, October 2009, <<https://www.rfc-editor.org/info/rfc5644>>.
- [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>>.

#### 14.2. Informative References

- [I-D.ietf-ippm-route] Alvarez-Hamelin, J., Morton, A., Fabini, J., Pignataro, C., and R. Geib, "Advanced Unidirectional Route Assessment (AURA)", draft-ietf-ippm-route-07 (work in progress), December 2019.
- [I-D.mizrahi-ippm-compact-alternate-marking] Mizrahi, T., Arad, C., Fioccola, G., Cociglio, M., Chen, M., Zheng, L., and G. Mirsky, "Compact Alternate Marking Methods for Passive and Hybrid Performance Monitoring", draft-mizrahi-ippm-compact-alternate-marking-05 (work in progress), July 2019.
- [I-D.song-opsawg-ifit-framework] Song, H., Qin, F., Chen, H., Jin, J., and J. Shin, "In-situ Flow Information Telemetry", draft-song-opsawg-ifit-framework-11 (work in progress), March 2020.
- [I-D.zhou-ippm-enhanced-alternate-marking] Zhou, T., Fioccola, G., Li, Z., Lee, S., and M. Cociglio, "Enhanced Alternate Marking Method", draft-zhou-ippm-enhanced-alternate-marking-04 (work in progress), October 2019.
- [IEEE-ACM-ToN-MPNPM] IEEE/ACM TRANSACTION ON NETWORKING, "Multipoint Passive Monitoring in Packet Networks", DOI 10.1109/TNET.2019.2950157, 2019.

[IEEE-Network-PNPM]

IEEE Network, "AM-PM: Efficient Network Telemetry using Alternate Marking", DOI 10.1109/MNET.2019.1800152, 2019.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

#### Authors' Addresses

Giuseppe Fioccola (editor)  
Huawei Technologies  
Riesstrasse, 25  
Munich 80992  
Germany

Email: [giuseppe.fioccola@huawei.com](mailto:giuseppe.fioccola@huawei.com)

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

Email: [mauro.cociglio@telecomitalia.it](mailto:mauro.cociglio@telecomitalia.it)

Amedeo Sapio  
Politecnico di Torino  
Corso Duca degli Abruzzi, 24  
Torino 10129  
Italy

Email: [amedeo.sapio@polito.it](mailto:amedeo.sapio@polito.it)

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

Email: [riccardo.sisto@polito.it](mailto:riccardo.sisto@polito.it)

Network Working Group  
Internet-Draft  
Updates: 2330 (if approved)  
Intended status: Standards Track  
Expires: February 14, 2021

J. Alvarez-Hamelin  
Universidad de Buenos Aires  
A. Morton  
AT&T Labs  
J. Fabini  
TU Wien  
C. Pignataro  
Cisco Systems, Inc.  
R. Geib  
Deutsche Telekom  
August 13, 2020

Advanced Unidirectional Route Assessment (AURA)  
draft-ietf-ippm-route-10

Abstract

This memo introduces an advanced unidirectional route assessment (AURA) metric and associated measurement methodology, based on the IP Performance Metrics (IPPM) Framework RFC 2330. This memo updates RFC 2330 in the areas of path-related terminology and path description, primarily to include the possibility of parallel subpaths between a given Source and Destination pair, owing to the presence of multi-path technologies.

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 February 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.



This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Issues with Earlier Work to Define a Route Metric . . . . .	3
1.2. Requirements Language . . . . .	4
2. Scope . . . . .	4
3. Route Metric Specifications . . . . .	5
3.1. Terms and Definitions . . . . .	5
3.2. Formal Name . . . . .	6
3.3. Parameters . . . . .	6
3.4. Metric Definitions . . . . .	7
3.5. Related Round-Trip Delay and Loss Definitions . . . . .	9
3.6. Discussion . . . . .	10
3.7. Reporting the Metric . . . . .	10
4. Route Assessment Methodologies . . . . .	11
4.1. Active Methodologies . . . . .	11
4.1.1. Temporal Composition for Route Metrics . . . . .	13
4.1.2. Routing Class Identification . . . . .	15
4.1.3. Intermediate Observation Point Route Measurement . . . . .	16
4.2. Hybrid Methodologies . . . . .	16
4.3. Combining Different Methods . . . . .	17
5. Background on Round-Trip Delay Measurement Goals . . . . .	17
6. RTD Measurements Statistics . . . . .	18
7. Security Considerations . . . . .	20
8. IANA Considerations . . . . .	21
9. Acknowledgements . . . . .	21
10. Appendix I MPLS Methods for Route Assessment . . . . .	21
11. References . . . . .	22
11.1. Normative References . . . . .	22
11.2. Informative References . . . . .	24
Authors' Addresses . . . . .	26

## 1. Introduction

The IETF IP Performance Metrics (IPPM) working group first created a framework for metric development in [RFC2330]. This framework has stood the test of time and enabled development of many fundamental metrics. It has been updated in the area of metric composition

[RFC5835], and in several areas related to active stream measurement of modern networks with reactive properties [RFC7312].

The [RFC2330] framework motivated the development of "performance and reliability metrics for paths through the Internet," and Section 5 of [RFC2330] defines terms that support description of a path under test. However, metrics for assessment of paths and related performance aspects had not been attempted in IPPM when the [RFC2330] framework was written.

This memo takes up the route measurement challenge and specifies a new route metric, two practical frameworks for methods of measurement (using either active or hybrid active-passive methods [RFC7799]), and Round-Trip Delay and link information discovery using the results of measurements. All route measurements are limited by the willingness of hosts along the path to be discovered, to cooperate with the methods used, or to recognize that the measurement operation is taking place (such as when tunnels are present).

#### 1.1. Issues with Earlier Work to Define a Route Metric

Section 7 of [RFC2330] presented a simple example of a "route" metric along with several other examples. The example is reproduced below (where the reference is to Section 5 of [RFC2330]):

"route: The path, as defined in Section 5, from A to B at a given time."

This example provides a starting point to develop a more complete definition of route. Areas needing clarification include:

Time: In practice, the route will be assessed over a time interval, because active path detection methods like Paris Traceroute [PT] rely on hop limits for their operation and cannot accomplish discovery of all hosts using a single packet.

Type-P: The legacy route definition lacks the option to cater for packet-dependent routing. In this memo, we assess the route for a specific packet of Type-P, and reflect this in the metric definition. The methods of measurement determine the specific Type-P used.

Parallel Paths: Parallel paths are a reality of the Internet and a strength of advanced route assessment methods, so the metric must acknowledge this possibility. Use of Equal Cost Multi-Path (ECMP) and Unequal Cost Multi-Path (UCMP) technologies are common sources of parallel subpaths.

Cloud Subpath: May contain hosts that do not decrement hop limit, but may have two or more exchange links connecting "discoverable" hosts or routers. Parallel subpaths contained within clouds cannot be discovered. The assessment methods only discover hosts or routers on the path that decrement hop limit, or cooperate with interrogation protocols. The presence of tunnels and nested tunnels further complicate assessment by hiding hops.

Hop: Although the [RFC2330] definition of a hop was a link-host pair, only hosts that are discoverable or have the capability to cooperate with interrogation protocols where link information may be exposed.

The refined definition of Route metrics begins in the sections that follow.

## 1.2. 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.

## 2. Scope

The purpose of this memo is to add new route metrics and methods of measurement to the existing set of IPPM metrics.

The scope is to define route metrics that can identify the path taken by a packet or a flow traversing the Internet between two hosts. Although primarily intended for hosts communicating on the Internet, the definitions and metrics are constructed to be applicable to other network domains, if desired. The methods of measurement to assess the path may not be able to discover all hosts comprising the path, but such omissions are often deterministic and explainable sources of error.

This memo also specifies a framework for active methods of measurement which uses the techniques described in [PT], as well as a framework for hybrid active-passive methods of measurement such as the Hybrid Type I method [RFC7799] described in [I-D.ietf-ippm-ioam-data]. Methods using [I-D.ietf-ippm-ioam-data] are intended only for single administrative domains that provide a protocol for explicit interrogation of nodes on a path. Combinations of active methods and hybrid active-passive methods are also in-scope.

Further, this memo provides additional analysis of the round-trip delay measurements made possible by the methods, in an effort to discover more details about the path, such as the link technology in use.

This memo updates Section 5 of [RFC2330] in the areas of path-related terminology and path description, primarily to include the possibility of parallel subpaths between a given Source and Destination address pair (possibly resulting from Equal Cost Multi-Path (ECMP) and Unequal Cost Multi-Path (UCMP) technologies).

There are several simple non-goals of this memo. There is no attempt to assess the reverse path from any host on the path to the host attempting the path measurement. The reverse path contribution to delay will be that experienced by ICMP packets (in active methods), and may be different from delays experienced by UDP or TCP packets. Also, the round trip delay will include an unknown contribution of processing time at the host that generates the ICMP response. Therefore, the ICMP-based active methods are not supposed to yield accurate, reproducible estimations of the Round-Trip Delay that UDP or TCP packets will experience.

### 3. Route Metric Specifications

This section sets requirements for the components of the Route Metric.

#### 3.1. Terms and Definitions

**Host** A Host as defined in [RFC2330] (a computer capable of IP communication, includes routers), a.k.a. RFC 2330 Host.

**Node** A Node is any network function on the path capable of IP-layer Communication, includes RFC 2330 Hosts.

**Node Identity** The unique address for Nodes communicating within the network domain. For Nodes communicating on the Internet with IP, it is the globally routable IP address which the Node uses when communicating with other Nodes under normal or error conditions. The Node Identity revealed (and its connection to a Node Name through reverse DNS) determines whether interfaces to parallel links can be associated with a single Node, or appear to identify unique Nodes.

**Discoverable Node** Nodes that convey their Node Identity according to the requirements of their network domain, such as when error conditions are detected by that Node. For Nodes communicating with IP packets, compliance with Section 3.2.2.4 of [RFC1122] when

discarding a packet due to TTL or Hop Limit Exceeded condition, MUST result in sending the corresponding Time Exceeded message (containing a form of Node identity) to the source. This requirement is also consistent with section 5.3.1 of [RFC1812] for routers.

**Cooperating Node** Nodes that respond to direct queries for their Node identity as part of a previously agreed and established interrogation protocol. Nodes SHOULD also provide information such as arrival/departure interface identification, arrival timestamp, and any relevant information about the Node or specific link which delivered the query to the Node.

**Hop specification** A Hop specification MUST contain a Node Identity, and MAY contain arrival and/or departure interface identification, round trip delay, and an arrival timestamp.

**Routing Class** A route that treats equally a class of different types of packets, designated "C" (unrelated to address classes of the past) [RFC2330] [RFC8468]. Knowledge of such a class allows any one of the types of packets within that class to be used for subsequent measurement of the route. The designator "class C" is used for historical reasons, see [RFC2330].

### 3.2. Formal Name

The formal name of the metric is:

Type-P-Route-Ensemble-Method-Variant

abbreviated as Route Ensemble.

Note that Type-P depends heavily on the chosen method and variant.

### 3.3. Parameters

This section lists the REQUIRED input factors to define and measure a Route metric, as specified in this memo.

- o Src, the address of a Node (such as the globally routable IP address).
- o Dst, the address of a Node (such as the globally routable IP address).
- o i, the limit on the number of Hops a specific packet may visit as it traverses from the Node at Src to the Node at Dst (such as the TTL or Hop Limit).

- o MaxHops, the maximum value of  $i$  used, ( $i=1,2,3,\dots\text{MaxHops}$ ).
- o  $T_0$ , a time (start of measurement interval)
- o  $T_f$ , a time (end of measurement interval)
- o  $\text{MP}(\text{address})$ , Measurement Point at address, such as Src or Dst, usually at the same node stack layer as "address".
- o  $T$ , the Node time of a packet as measured at  $\text{MP}(\text{Src})$ , meaning Measurement Point at the Source.
- o  $T_a$ , the Node time of a reply packet's \*arrival\* as measured at  $\text{MP}(\text{Src})$ , assigned to packets that arrive within a "reasonable" time (see parameter below).
- o  $T_{\text{max}}$ , a maximum waiting time for reply packets to return to the source, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost), such that the distribution of Round-Trip Delay is not truncated.
- o  $F$ , the number of different flows simulated by the method and variant.
- o 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 if the  $\text{MP}(\text{Src})$  is a Tunnel End Point (TEP) complying with [RFC6438] guidelines.
- o Type-P, the complete description of the packets for which this assessment applies (including the flow-defining fields).

### 3.4. Metric Definitions

This section defines the REQUIRED measurement components of the Route metrics (unless otherwise indicated):

$M$ , the total number of packets sent between  $T_0$  and  $T_f$ .

$N$ , the smallest value of  $i$  needed for a packet to be received at Dst (sent between  $T_0$  and  $T_f$ ).

$N_{\text{max}}$ , the largest value of  $i$  needed for a packet to be received at Dst (sent between  $T_0$  and  $T_f$ ).  $N_{\text{max}}$  may be equal to  $N$ .

Next define a *\*singleton\** definition for a Node on the path, with sufficient indexes to identify all Nodes identified in a measurement interval (where *\*singleton\** is part of the IPPM Framework [RFC2330]).

A Hop Specification, designated  $h(i,j)$ , the IP address and/or identity of Discoverable Nodes (or Cooperating Nodes) that are  $i$  hops away from the Node with address = Src and part of Route  $j$  during the measurement interval,  $T_0$  to  $T_f$ . As defined here, a Hop singleton measurement MUST contain a Node Identity,  $hid(i,j)$ , and MAY contain one or more of the following attributes:

- o  $a(i,j)$  Arrival Interface ID (e.g., when [RFC5837] is supported)
- o  $d(i,j)$  Departure Interface ID (e.g., when [RFC5837] is supported)
- o  $t(i,j)$  Arrival Timestamp, where  $t(i,j)$  is ideally supplied by the Hop. (Note that  $t(i,j)$  might be approximated from the sending time of the packet that revealed the Hop, e.g., when the round trip response time is available and divided by 2.)
- o Measurements of Round-Trip Delay (for each packet that reveals the same Node Identity and flow attributes, then this attribute is computed, see next section)

Node Identities and related information can be ordered by their distance from the Node with address Src in Hops  $h(i,j)$ . Based on this, two forms of Routes are distinguished:

A Route Ensemble is defined as the combination of all routes traversed by different flows from the Node at Src address to the Node at Dst address. A single Route traversed by a single flow (determined by an unambiguous tuple of addresses Src and Dst, and other identical flow criteria) is a member of the Route Ensemble and called a Member Route.

Using  $h(i,j)$  and components and parameters, further define:

When considering the set of Hops in the context of a single flow, a Member Route  $j$  is an ordered list  $\{h(1,j), \dots, h(N_j, j)\}$  where  $h(i-1, j)$  and  $h(i, j)$  are 1 hop away from each other and  $N_j$  satisfying  $h(N_j, j) = \text{Dst}$  is the minimum count of Hops needed by the packet on Member Route  $j$  to reach Dst. Member Routes must be unique. The uniqueness property requires that any two Member routes  $j$  and  $k$  that are part of the same Route Ensemble differ either in terms of minimum hop count  $N_j$  and  $N_k$  to reach the destination Dst, or, in the case of identical hop count  $N_j = N_k$ , they have at least one distinct Hop:  $h(i, j) \neq h(i, k)$  for at least one  $i$  ( $i=1..N_j$ ).

All the optional information collected to describe a Member Route, such as the arrival interface, departure interface, and Round Trip Delay at each Hop, turns each list item into a rich structure. There may be information on the links between Hops, possibly information on the routing (arrival interface and departure interface), an estimate of distance between Hops based on Round-Trip Delay measurements and calculations, and a time stamp indicating when all these additional details were valid.

The Route Ensemble from Src to Dst, during the measurement interval  $T_0$  to  $T_f$ , is the aggregate of all  $m$  distinct Member Routes discovered between the two Nodes with Src and Dst addresses. More formally, with the Node having address Src omitted:

```
Route Ensemble = {
{h(1,1), h(2,1), h(3,1), ... h(N1,1)=Dst},
{h(1,2), h(2,2), h(3,2), ..., h(N2,2)=Dst},
...
{h(1,m), h(2,m), h(3,m), ....h(Nm,m)=Dst}
}
```

where the following conditions apply:  $i \leq N_j \leq N_{max}$  ( $j=1..m$ )

Note that some  $h(i,j)$  may be empty (null) in the case that systems do not reply (not discoverable, or not cooperating).

$h(i-1,j)$  and  $h(i,j)$  are the Hops on the same Member Route one hop away from each other.

Hop  $h(i,j)$  may be identical with  $h(k,l)$  for  $i \neq k$  and  $j \neq l$  ; which means there may be portions shared among different Member Routes (parts of Member Routes may overlap).

### 3.5. Related Round-Trip Delay and Loss Definitions

RTD( $i,j,T$ ) is defined as a singleton of the [RFC2681] Round-Trip Delay between the Node with address = Src and the Node at Hop  $h(i,j)$  at time  $T$ .

RTL( $i,j,T$ ) is defined as a singleton of the [RFC6673] Round-trip Loss between the Node with address = Src and the Node at Hop  $h(i,j)$  at time  $T$ .



### 3.6. Discussion

Depending on the way that Node Identity is revealed, it may be difficult to determine parallel subpaths between the same pair of Nodes (i.e. multiple parallel links). It is easier to detect parallel subpaths involving different Nodes.

- o If a pair of discovered Nodes identify two different addresses (IP or not), then they will appear to be different Nodes. See item below.
- o If a pair of discovered Nodes identify two different IP addresses, and the IP addresses resolve to the same Node name (in the DNS), then they will appear to be the same Nodes.
- o If a discovered Node always replies using the same network address, regardless of the interface a packet arrives on, then multiple parallel links cannot be detected in that network domain. This condition may apply to traceroute-style methods, but may not apply to other hybrid methods based on In-situ Operations, Administration, and Maintenance (IOAM). For example, if the [RFC5837] ICMP extension mechanism is implemented, then parallel links can be detected with the discovery traceroute-style methods.
- o If parallel links between routers are aggregated below the IP layer, then from Node point of view, all these links share the same pair of IP addresses. The existence of these parallel links can't be detected at the IP layer. This applies to other network domains with layers below them, as well. This condition may apply to traceroute-style methods, but may not apply to other hybrid methods based on IOAM.

When a route assessment employs IP packets (for example), the reality of flow assignment to parallel subpaths involves layers above IP. Thus, the measured Route Ensemble is applicable to IP and higher layers (as described in the methodology's packet of Type-P and flow parameters).

### 3.7. Reporting the Metric

An Information Model and an XML Data Model for Storing Traceroute Measurements is available in [RFC5388]. The measured information at each hop includes four pieces of information: a one-dimensional hop index, Node symbolic address, Node IP address, and RTD for each response.

The description of Hop information that may be collected according to this memo covers more dimensions, as defined in Section 3.4 above.

For example, the Hop index is two-dimensional to capture the complexity of a Route Ensemble, and it contains corresponding Node identities at a minimum. The models need to be expanded to include these features, as well as Arrival Interface ID, Departure Interface ID, and Arrival Timestamp, when available. The original sending Timestamp from the Src Node anchors a particular measurement in time.

#### 4. Route Assessment Methodologies

There are two classes of methods described in this section, active methods relying on the reaction to TTL or Hop Limit Exceeded condition to discover Nodes on a path, and Hybrid active-passive methods that involve direct interrogation of cooperating Nodes (usually within a single domain). Description of these methods follow.

##### 4.1. Active Methodologies

This section describes the method employed by current open source tools, thereby providing a practical framework for further advanced techniques to be included as method variants. This method is applicable for use across multiple administrative domains.

Internet routing is complex because it depends on the policies of thousands of Autonomous Systems (AS). Most routers perform load balancing on flows using a form of Equal Cost Multiple Path (ECMP). [RFC2991] describes a number of flow-based or hashed approaches (e.g., Modulo-N Hash, Hash-Threshold, Highest Random Weight (HRW)), and makes some good suggestions. Flow-based ECMP avoids increased packet delay variation and possibly overwhelming levels of packet reordering in flows.

A few routers still divide the workload through packet-based techniques, such as a round-robin scheme to distribute every new outgoing packet to multiple links, as explained in [RFC2991]. The methods described in this section assume flow-based ECMP.

Taking into account that Internet protocol was designed under the "end-to-end" principle, the IP payload and its header do not provide any information about the routes or path necessary to reach some destination. For this reason, the popular tool traceroute was developed to gather the IP addresses of each hop along a path using the ICMP protocol [RFC0792]. Traceroute also measures RTD from each hop. However, the growing complexity of the Internet makes it more challenging to develop an accurate traceroute implementation. For instance, the early traceroute tools would be inaccurate in the current network, mainly because they were not designed to retain a flow state. However, evolved traceroute tools, such as Paris-

traceroute [PT] [MLB] and Scamper [SCAMPER], expect to encounter ECMP and achieve more accurate results when they do, where Scamper ensures traceroute packets will follow the same path in 98% of cases[SCAMPER].

Today's traceroute tools send Type-P of packets, either ICMP, UDP, or TCP. UDP and TCP are used when a particular characteristic needs to be verified, such as filtering or traffic shaping on specific ports (i.e., services). UDP and TCP traceroute are also used when ICMP responses are not received. [SCAMPER] supports IPv6 traceroute measurements, keeping the FlowLabel constant in all packets.

Paris-traceroute allows its users to measure the RTD to every Node of the path for a particular flow. Furthermore, either Paris-traceroute or Scamper is capable of unveiling the many available paths between a source and destination (which are visible to active methods). This task is accomplished by repeating complete traceroute measurements with different flow parameters for each measurement; Paris-traceroute provides "exhaustive" mode while scamper provides "tracelb" (stands for traceroute load balance). The Framework for IP Performance Metrics (IPPM) ([RFC2330] updated by[RFC7312]) has the flexibility to require that the Round-Trip Delay measurement [RFC2681] uses packets with the constraints to assure that all packets in a single measurement appear as the same flow. This flexibility covers ICMP, UDP, and TCP. The accompanying methodology of [RFC2681] needs to be expanded to report the sequential hop identifiers along with RTD measurements, but no new metric definition is needed.

The advanced route assessment methods used in Paris-traceroute [PT] keep the critical fields constant for every packet to maintain the appearance of the same flow. When considering IPv6 headers, it is necessary to ensure that the IP source and destination addresses and the FlowLabel are constant (but note that many routers ignore the FlowLabel field at this time), see [RFC6437]. Use of IPv6 Extension Headers may add critical fields, and SHOULD be avoided. In IPv4, certain fields of the IP header and the first four bytes of the IP payload should remain constant in a flow. In the IPv4 header, the IP source and destination addresses, protocol number, and Diffserv fields identify flows. The first four payload bytes include the UDP and TCP ports, and the ICMP type, code, and checksum fields.

Maintaining a constant ICMP checksum in IPv4 is most challenging, as the ICMP sequence number or identifier fields will usually change for different probes of the same path. Probes should use arbitrary bytes in the ICMP data field to offset changes to sequence number and identifier, thus keeping the checksum constant.

Finally, it is also essential to route the resulting ICMP Time Exceeded messages along a consistent path. In IPv6, the fields above are sufficient. In IPv4, the ICMP Time Exceeded message will contain the IP header and the first eight bytes of the IP payload, which affects its ICMP checksum. The TCP sequence number, UDP Length, and UDP checksum will affect this value, and should remain constant.

Formally, to maintain the same flow in the measurements to a particular hop, the Type-P-Route-Ensemble-Method-Variant packets should be[PT]:

- o TCP case: For IPv4, the fields Src, Dst, port-Src, port-Dst, sequence number, and Diffserv Field SHOULD be the same. For IPv6, the field FlowLabel, Src and Dst SHOULD be the same.
- o UDP case: For IPv4, the fields Src, Dst, port-Src, port-Dst, Diffserv should be the same, and the UDP-checksum SHOULD change to keep the IP checksum of the ICMP time exceeded reply constant. Then, the data length should be fixed, and the data field is used to make it so (consider that ICMP checksum uses its data field, which contains the original IP header plus 8 bytes of UDP, where TTL, IP identification, IP checksum, and UDP checksum changes). For IPv6, the field FlowLabel, and Source and Destination addresses SHOULD be the same.
- o ICMP case: For IPv4, the Data field SHOULD compensate variations on TTL or Hop Limit, IP identification, and IP checksum for every packet. There is no need to consider ICMPv6 because only FlowLabel of IPv6 and Source and Destination addresses are used, and all of them SHOULD be constant.

Then, the way to identify different hops and attempts of the same IPv4 flow is:

- o TCP case: The IP identification field.
- o UDP case: The IP identification field.
- o ICMP case: The IP identification field, and ICMP Sequence number.

#### 4.1.1. Temporal Composition for Route Metrics

The Active Route Assessment Methods described above have the ability to discover portions of a path where ECMP load balancing is present, observed as two or more unique Member Routes having one or more distinct Hops which are part of the Route Ensemble. Likewise, attempts to deliberately vary the flow characteristics to discover

all Member Routes will reveal portions of the path which are flow-invariant.

Section 9.2 of [RFC2330] describes Temporal Composition of metrics, and introduces the possibility of a relationship between earlier measurement results and the results for measurement at the current time (for a given metric). There is value in establishing a Temporal Composition relationship for Route Metrics. However, this relationship does not represent a forecast of future route conditions in any way.

For Route Metric measurements, the value of Temporal Composition is to reduce the measurement iterations required with repeated measurements. Reduced iterations are possible by inferring that current measurements using fixed and previously measured flow characteristics:

- o will have many common hops with previous measurements.
- o will have relatively time-stable results at the ingress and egress portions of the path when measured from user locations, as opposed to measurements of backbone networks and across inter-domain gateways.
- o may have greater potential for time-variation in path portions where ECMP load balancing is observed (because increasing or decreasing the pool of links changes the hash calculations).

Optionally, measurement systems may take advantage of the inferences above when seeking to reduce measurement iterations, after exhaustive measurements indicate that the time-stable properties are present. Repetitive Active Route measurement systems:

1. SHOULD occasionally check path portions which have exhibited stable results over time, particularly ingress and egress portions of the path (e.g., daily checks if measuring many times during a day).
2. SHOULD continue testing portions of the path that have previously exhibited ECMP load balancing.
3. SHALL trigger re-assessment of the complete path and Route Ensemble, if any change in hops is observed for a specific (and previously tested) flow.

#### 4.1.1.2. Routing Class Identification

There is an opportunity to apply the [RFC2330] notion of equal treatment for a class of packets, "...very useful to know if a given Internet component treats equally a class C of different types of packets", as it applies to Route measurements. The notion of class C was examined further in [RFC8468] as it applied to load-balancing flows over parallel paths, which is the case we develop here. Knowledge of class C parameters (unrelated to address classes of the past) on a path potentially reduces the number of flows required for a given method to assess a Route Ensemble over time.

First, recognize that each Member Route of a Route Ensemble will have a corresponding class C. Class C can be discovered by testing with multiple flows, all of which traverse the unique set of hops that comprise a specific Member Route.

Second, recognize that the different classes depend primarily on the hash functions used at each instance of ECMP load balancing on the path.

Third, recognize the synergy with Temporal Composition methods (described above), where evaluation intends to discover time-stable portions of each Member Route, so that more emphasis can be placed on ECMP portions that also determine class C.

The methods to assess the various class C characteristics benefit from the following measurement capabilities:

- o flows designed to determine which n-tuple header fields are considered by a given hash function and ECMP hop on the path, and which are not. This operation immediately narrows the search space, where possible, and partially defines a class C.
- o a priori knowledge of the possible types of hash functions in use also helps to design the flows for testing (major router vendors publish information about these hash functions, examples are here [LOAD\_BALANCE]).
- o ability to direct the emphasis of current measurements on ECMP portions of the path, based on recent past measurement results (the Routing Class of some portions of the path is essentially "all packets").

#### 4.1.3. Intermediate Observation Point Route Measurement

There are many examples where passive monitoring of a flow at an Observation Point within the network can detect unexpected Round Trip Delay or Delay Variation. But how can the cause of the anomalous delay be investigated further \*from the Observation Point\* possibly located at an intermediate point on the path?

In this case, knowledge that the flow of interest belongs to a specific Routing Class C will enable measurement of the route where anomalous delay has been observed. Specifically, Round-Trip Delay assessment to each Hop on the path between the Observation Point and the Destination for the flow of interest may discover high or variable delay on a specific link and Hop combination.

The determination of a Routing Class C which includes the flow of interest is as described in the section above, aided by computation of the relevant hash function output as the target.

#### 4.2. Hybrid Methodologies

The Hybrid Type I methods provide an alternative method for Route Member assessment. As mentioned in the Scope section, [I-D.ietf-ippm-ioam-data] provides a possible set of data fields that would support route identification.

In general, nodes in the measured domain would be equipped with specific abilities:

- o Store the identity of nodes that a packet has visited in header data fields, in the order the packet visited the nodes.
- o Support of a "Loopback" capability, where a copy of the packet is returned to the encapsulating node, and the packet is processed like any other IOAM packet on the return transfer.

In addition to node identity, nodes may also identify the ingress and egress interfaces utilized by the tracing packet, the absolute time when the packet was processed, and other generic data (as described in section 4 of [I-D.ietf-ippm-ioam-data]). Interface identification isn't necessarily limited to IP, i.e. different links in a bundle (LACP) could be identified. Equally well, links without explicit IP addresses can be identified (like with unnumbered interfaces in an IGP deployment).

Note that the Type-P packet specification for this method will likely be a partial specification, because most of the packet fields are determined by the user traffic. The packet (encapsulation) header(s)

added by the Hybrid method can certainly be specified in Type-P, in unpopulated form.

#### 4.3. Combining Different Methods

In principle, there are advantages if the entity conducting Route measurements can utilize both forms of advanced methods (active and hybrid), and combine the results. For example, if there are Nodes involved in the path that qualify as Cooperating Nodes, but not as Discoverable Nodes, then a more complete view of Hops on the path is possible when a hybrid method (or interrogation protocol) is applied and the results are combined with the active method results collected across all other domains.

In order to combine the results of active and hybrid/interrogation methods, the network Nodes that are part of a domain supporting an interrogation protocol have the following attributes:

1. Nodes at the ingress to the domain SHOULD be both Discoverable and Cooperating.
2. Any Nodes within the domain that are both Discoverable and Cooperating SHOULD reveal the same Node Identity in response to both active and hybrid methods.
3. Nodes at the egress to the domain SHOULD be both Discoverable and Cooperating, and SHOULD reveal the same Node Identity in response to both active and hybrid methods.

When Nodes follow these requirements, it becomes a simple matter to match single domain measurements with the overlapping results from a multidomain measurement.

In practice, Internet users do not typically have the ability to utilize the OAM capabilities of networks that their packets traverse, so the results from a remote domain supporting an interrogation protocol would not normally be accessible. However, a network operator could combine interrogation results from their access domain with other measurements revealing the path outside their domain.

#### 5. Background on Round-Trip Delay Measurement Goals

The aim of this method is to use packet probes to unveil the paths between any two end-Nodes of the network. Moreover, information derived from RTD measurements might be meaningful to identify:

1. Intercontinental submarine links



2. Satellite communications
3. Congestion
4. Inter-domain paths

This categorization is widely accepted in the literature and among operators alike, and it can be trusted with empirical data and several sources as ground of truth (e.g., [RTTSub] ) but it is an inference measurement nonetheless [bdrmap][IDCong].

The first two categories correspond to the physical distance dependency on Round-Trip Delay (RTD), the next one binds RTD with queuing delay on routers, and the last one helps to identify different ASes using traceroutes. Due to the significant contribution of propagation delay in long-distance hops, RTD will be on the order of 100ms on transatlantic hops, depending on the geolocation of the vantage points. Moreover, RTD is typically higher than 480ms when two hops are connected using geostationary satellite technology (i.e., their orbit is at 36000km). Detecting congestion with latency implies deeper mathematical understanding since network traffic load is not stationary. Nonetheless, as the first approach, a link seems to be congested if observing different/varying statistical results after sending several traceroute probes (e.g., see [IDCong]). Finally, to recognize distinctive ASes in the same traceroute path is challenging, because more data is needed, like AS relationships and RIR delegations among other (for more detail, please consult [bdrmap]).

## 6. RTD Measurements Statistics

Several articles have shown that network traffic presents a self-similar nature [SSNT] [MLRM] which is accountable for filling the queues of the routers. Moreover, router queues are designed to handle traffic bursts, which is one of the most remarkable features of self-similarity. Naturally, while queue length increases, the delay to traverse the queue increases as well and leads to an increase on RTD. Due to traffic bursts generating short-term overflow on buffers (spiky patterns), every RTD only depicts the queueing status on the instant when that packet probe was in transit. For this reason, several RTD measurements during a time window could begin to describe the random behavior of latency. Loss must also be accounted for in the methodology.

To understand the ongoing process, examining the quartiles provides a non-parametric way of analysis. Quartiles are defined by five values: minimum RTD (m), RTD value of the 25% of the Empirical Cumulative Distribution Function (ECDF) (Q1), the median value (Q2),

the RTD value of the 75% of the ECDF (Q3) and the maximum RTD (M). Congestion can be inferred when RTD measurements are spread apart, and consequently, the Inter-Quartile Range (IQR), the distance between Q3 and Q1, increases its value.

This procedure requires the algorithm presented in [P2] to compute quartile values "on the fly".

This procedure allows us to update the quartiles value whenever a new measurement arrives, which is radically different from classic methods of computing quartiles because they need to use the whole dataset to compute the values. This way of calculus provides savings in memory and computing time.

To sum up, the proposed measurement procedure consists of performing traceroutes several times to obtain samples of the RTD in every hop from a path, during a time window (W), and compute the quartiles for every hop. This procedure could be done for a single Member Route flow, a non-exhaustive search with parameter E (defined below) set as False, or for every detected Route Ensemble flow (E=True).

The identification of a specific Hop in traceroute is based on the IP origin address of the returned ICMP Time Exceeded packet, and on the distance identified by the value set in the TTL (or Hop Limit) field inserted by traceroute. As this specific Hop can be reached by different paths, also the IP source and destination addresses of the traceroute packet need to be recorded. Finally, different return paths are distinguished by evaluating the ICMP Time Exceeded TTL (or Hop Limit) of the reply message: if this TTL (or Hop Limit) is constant for different paths containing the same Hop, the return paths have the same distance. Moreover, this distance can be estimated considering that the TTL (or Hop Limit) value is normally initialized with values 64, 128, or 255. The 5-tuple (origin IP, destination IP, reply IP, distance, response TTL or Hop Limit) unequivocally identifies every measurement.

This algorithm below runs in the origin of the traceroute. It returns the Qs quartiles for every Hop and Alt (alternative paths because of balancing). Notice that the "Alt" parameter condenses the parameters of the 5-tuple (origin IP, destination IP, reply IP, distance, response TTL), i.e., one for each possible combination.

```

=====
0  input:  W (window time of the measurement)
1          i_t (time between two measurements, set the i_t time
2              long enough to avoid incomplete results)
3          E (True: exhaustive, False: a single path)
4          Dst (destination IP address)
5  output: Qs (quartiles for every Hop and Alt)
=====
6  T := start_timer(W)
7  while T is not finished do:
8      start_timer(i_t)
9      RTD(Hop,Alt) = advanced-traceroute(Dst,E)
10     for each Hop and Alt in RTD do:
11         |   Qs[Dst,Hop,Alt] := ComputeQs(RTD(Hop,Alt))
12     done
13     wait until i_t timer is expired
14 done
15 return (Qs)
=====

```

During the time *W*, lines 6 and 7 assure that the measurement loop is made. Line 8 and 13 set a timer for each cycle of measurements. A cycle comprises the traceroutes packets, considering every possible Hop and the alternatives paths in the Alt variable (ensured in lines 9-12). In line 9, the advance-traceroute could be either Paris-traceroute or Scamper, which will use the "exhaustive" mode or "tracelb" option if *E* is set True, respectively. The procedure returns a list of tuples (*m*,*Q1*,*Q2*,*Q3*,*M*) for each intermediate hop, or "Alt" in as a function of the 5-tuple, in the path towards the Dst. Finally, lines 10 through 12 stores each measurement into the real-time quartiles computation.

Notice there are cases where the even having a unique hop at distance *h* from the Src to Dst, the returning path could have several possibilities, yielding in different total paths. In this situation, the algorithm will return more "Alt" for this particular hop.

## 7. Security Considerations

The security considerations that apply to any active measurement of live paths are relevant here as well. See [RFC4656] and [RFC5357].

The active measurement process of "changing several fields to keep the checksum of different packets identical" does not require special security considerations because it is part of synthetic traffic generation, and is designed to have minimal to zero impact on network processing (to process the packets for ECMP).

Some of the protocols used (e.g., ICMP) do not provide cryptographic protection for the requested/returned data, and there are risks of processing untrusted data in general, but these are limitations of the existing protocols where we are applying new methods.

For applicable Hybrid methods, the security considerations in[I-D.ietf-ippm-ioam-data] apply.

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.

## 8. IANA Considerations

This memo makes no requests of IANA. We thank the good folks at IANA for having checked this section anyway.

## 9. Acknowledgements

The original 3 authors (Ignacio, Al, Joachim) acknowledge Ruediger Geib, for his penetrating comments on the initial draft, and his initial text for the Appendix on MPLS. Carlos Pignataro challenged the authors to consider a wider scope, and applied his substantial expertise with many technologies and their measurement features in his extensive comments. Frank Brockners also shared useful comments, so did Footer Foote. We thank them all!

## 10. Appendix I MPLS Methods for Route Assessment

A Node assessing an MPLS path must be part of the MPLS domain where the path is implemented. When this condition is met, RFC 8029 provides a powerful set of mechanisms to detect "correct operation of the data plane, as well as a mechanism to verify the data plane against the control plane" [RFC8029].

MPLS routing is based on the presence of a Forwarding Equivalence Class (FEC) Stack in all visited Nodes. Selecting one of several Equal Cost Multi Path (ECMP) is however based on information hidden deeper in the stack. Late deployments may support a so called "Entropy label" for this purpose. State of the art deployments base their choice of an ECMP member interface on the complete MPLS label stack and on IP addresses up to the complete 5 tuple IP header information (see Section 2.4 of [RFC7325]). Load Sharing based on IP

information decouples this function from the actual MPLS routing information. Thus, an MPLS traceroute is able to check how packets with a contiguous number of ECMP relevant IP addresses (and an identical MPLS label stack) are forwarded by a particular router. The minimum number of equivalent MPLS paths traceable at a router should be 32. Implementations supporting more paths are available.

The MPLS echo request and reply messages offering this feature must support the Downstream Detailed Mapping TLV (was Downstream Mapping initially, but the latter has been deprecated). The MPLS echo response includes the incoming interface where a router received the MPLS Echo request. The MPLS Echo reply further informs which of the *n* addresses relevant for the load sharing decision results in a particular next hop interface and contains the next hop's interface address (if available). This ensures that the next hop will receive a properly coded MPLS Echo request in the next step route of assessment.

[RFC8403] explains how a central Path Monitoring System could be used to detect arbitrary MPLS paths between any routers within a single MPLS domain. The combination of MPLS forwarding, Segment Routing and MPLS traceroute offers a simple architecture and a powerful mechanism to detect and validate (segment routed) MPLS paths.

## 11. References

### 11.1. 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.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.

- [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>>.
- [RFC5388] Niccolini, S., Tartarelli, S., Quittek, J., Dietz, T., and M. Swamy, "Information Model and XML Data Model for Traceroute Measurements", RFC 5388, DOI 10.17487/RFC5388, December 2008, <<https://www.rfc-editor.org/info/rfc5388>>.
- [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>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [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>>.
- [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>>.
- [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>>.

## 11.2. Informative References

- [bdrmap] Luckie, M., Dhamdhere, A., Huffaker, B., Clark, D., and KC. Claffy, "bdrmap: Inference of Borders Between IP Networks", In Proceedings of the 2016 ACM on Internet Measurement Conference, pp. 381-396. ACM, 2016.
- [IDCong] Luckie, M., Dhamdhere, A., Clark, D., and B. Huffaker, "Challenges in inferring Internet interdomain congestion", In Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 15-22. ACM, 2014.
- [LOAD\_BALANCE] Sanguanpong, S., Pittayapitak, W., and K. Kasom Koht-Arsa, "COMPARISON OF HASH STRATEGIES FOR FLOW-BASED LOAD BALANCING", International Journal of Electronic Commerce Studies, Vol.6, No.2, pp.259-268. <http://dx.doi.org/10.7903/ijecs.1346>, 2015.
- [MLB] Augustin, B., Friedman, T., and R. Teixeira, "Measuring load-balanced paths in the Internet", Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 149-160. ACM, 2007., 2007.
- [MLRM] Fontugne, R., Mazel, J., and K. Fukuda, "An empirical mixture model for large-scale RTT measurements", 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 2470-2478. IEEE, 2015., 2015.
- [P2] Jain, R. and I. Chlamtac, "The P 2 algorithm for dynamic calculation of quartiles and histograms without storing observations", Communications of the ACM 28.10 (1985): 1076-1085, 2015.
- [PT] Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute", Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 153-158. ACM, 2006., 2006.

- [RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <<https://www.rfc-editor.org/info/rfc2991>>.
- [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>>.
- [RFC5835] Morton, A., Ed. and S. Van den Berghe, Ed., "Framework for Metric Composition", RFC 5835, DOI 10.17487/RFC5835, April 2010, <<https://www.rfc-editor.org/info/rfc5835>>.
- [RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <<https://www.rfc-editor.org/info/rfc5837>>.
- [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>>.
- [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>>.
- [RFC7325] Villamizar, C., Ed., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, DOI 10.17487/RFC7325, August 2014, <<https://www.rfc-editor.org/info/rfc7325>>.
- [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>>.
- [RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <<https://www.rfc-editor.org/info/rfc8403>>.



- [RTTSub] Bischof, Z., Rula, J., and F. Bustamante, "In and out of Cuba: Characterizing Cuba's connectivity", In Proceedings of the 2015 ACM Conference on Internet Measurement Conference, pp. 487-493. ACM, 2015.
- [SCAMPER] Matthew Luckie, M., "Scamper: a scalable and extensible packet prober for active measurement of the Internet", Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pp. 239-245. ACM, 2010., 2010.
- [SSNT] Park, K. and W. Willinger, "Self-Similar Network Traffic and Performance Evaluation (1st ed.)", John Wiley & Sons, Inc., New York, NY, USA, 2000.

## Authors' Addresses

J. Ignacio Alvarez-Hamelin  
Universidad de Buenos Aires  
Av. Paseo Colon 850  
Buenos Aires C1063ACV  
Argentina

Phone: +54 11 5285-0716  
Email: [ihameli@cnet.fi.uba.ar](mailto:ihameli@cnet.fi.uba.ar)  
URI: <http://cnet.fi.uba.ar/ignacio.alvarez-hamelin/>

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](mailto:acm@research.att.com)

Joachim Fabini  
TU Wien  
Gusshausstrasse 25/E389  
Vienna 1040  
Austria

Phone: +43 1 58801 38813  
Fax: +43 1 58801 38898  
Email: [Joachim.Fabini@tuwien.ac.at](mailto:Joachim.Fabini@tuwien.ac.at)  
URI: <http://www.tc.tuwien.ac.at/about-us/staff/joachim-fabini/>

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

Email: cpignata@cisco.com

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

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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2020

G. Mirsky  
ZTE Corp.  
G. Jun  
ZTE Corporation  
H. Nydell  
Accedian Networks  
R. Foote  
Nokia  
October 31, 2019

Simple Two-way Active Measurement Protocol  
draft-ietf-ippm-stamp-10

Abstract

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

Status of This Memo

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 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions used in this document . . . . .	3
2.1. Terminology . . . . .	3
2.2. Requirements Language . . . . .	3
3. Operation and Management of Performance Measurement Based on STAMP . . . . .	3
4. Theory of Operation . . . . .	4
4.1. UDP Port Numbers in STAMP Testing . . . . .	5
4.2. Session-Sender Behavior and Packet Format . . . . .	5
4.2.1. Session-Sender Packet Format in Unauthenticated Mode . . . . .	5
4.2.2. Session-Sender Packet Format in Authenticated Mode . . . . .	7
4.3. Session-Reflector Behavior and Packet Format . . . . .	8
4.3.1. Session-Reflector Packet Format in Unauthenticated Mode . . . . .	9
4.3.2. Session-Reflector Packet Format in Authenticated Mode . . . . .	10
4.4. Integrity Protection in STAMP . . . . .	11
4.5. Confidentiality Protection in STAMP . . . . .	12
4.6. Interoperability with TWAMP Light . . . . .	12
5. Operational Considerations . . . . .	13
6. IANA Considerations . . . . .	13
7. Security Considerations . . . . .	13
8. Acknowledgments . . . . .	14
9. References . . . . .	14
9.1. Normative References . . . . .	14
9.2. Informative References . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

Development and deployment of the Two-Way Active Measurement Protocol (TWAMP) [RFC5357] and its extensions, e.g., [RFC6038] that defined Symmetrical Size for TWAMP, provided invaluable experience. Several independent implementations of both TWAMP and TWAMP Light exist, have been deployed, and provide important operational performance measurements.

At the same time, there has been noticeable interest in using a more straightforward mechanism for active performance monitoring that can provide deterministic behavior and inherent separation of control (vendor-specific configuration or orchestration) and test functions. Recent work on IP Edge to Customer Equipment using TWAMP Light from Broadband Forum [BBF.TR-390] demonstrated that interoperability among

implementations of TWAMP Light is difficult because the composition and operation of TWAMP Light were not sufficiently specified in [RFC5357]. According to [RFC8545], TWAMP Light includes a sub-set of TWAMP-Test functions. Thus, to have a comprehensive tool to measure packet loss and delay requires support by other applications that provide, for example, control and security.

This document defines an active performance measurement test protocol, Simple Two-way Active Measurement Protocol (STAMP), that enables measurement of both one-way and round-trip performance metrics like delay, delay variation, and packet loss. Some TWAMP extensions, e.g., [RFC7750] are supported by the extensions to STAMP base specification in [I-D.ietf-ippm-stamp-option-tlv].

## 2. Conventions used in this document

### 2.1. Terminology

STAMP - Simple Two-way Active Measurement Protocol

NTP - Network Time Protocol

PTP - Precision Time Protocol

HMAC Hashed Message Authentication Code

OWAMP One-Way Active Measurement Protocol

TWAMP Two-Way Active Measurement Protocol

MBZ Must be Zero

### 2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Operation and Management of Performance Measurement Based on STAMP

Figure 1 presents the Simple Two-way Active Measurement Protocol (STAMP) Session-Sender, and Session-Reflector with a measurement session. In this document, a measurement session also referred to as STAMP session, is the bi-directional packet flow between one specific Session-Sender and one particular Session-Reflector for a time duration. The configuration and management of the STAMP Session-

Sender, Session-Reflector, and management of the STAMP sessions are outside the scope of this document and can be achieved through various means. A few examples are: Command Line Interface, telecommunication services' OSS/BSS systems, SNMP, and Netconf/YANG-based SDN controllers.

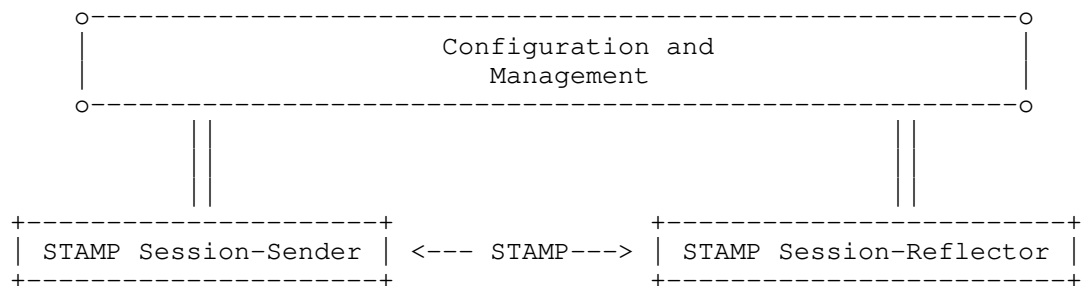


Figure 1: STAMP Reference Model

#### 4. Theory of Operation

STAMP Session-Sender transmits test packets over UDP transport toward STAMP Session-Reflector. STAMP Session-Reflector receives Session-Sender's packet and acts according to the configuration. Two modes of STAMP Session-Reflector characterize the expected behavior and, consequently, performance metrics that can be measured:

- o Stateless - STAMP Session-Reflector does not maintain test state and will use the value in the Sequence Number field in the received packet as the value for the Sequence Number field in the reflected packet. As a result, values in Sequence Number and Session-Sender Sequence Number fields are the same, and only round-trip packet loss can be calculated while the reflector is operating in stateless mode.
- o Stateful - STAMP Session-Reflector maintains test state thus enabling the ability to determine forward loss, gaps recognized in the received sequence number. As a result, both near-end (forward) and far-end (backward) packet loss can be computed. That implies that the STAMP Session-Reflector MUST keep a state for each configured STAMP-test session, uniquely identifying STAMP-test packets to one such session instance, and enabling adding a sequence number in the test reply that is individually incremented on a per-session basis.

STAMP supports two authentication modes: unauthenticated and authenticated. Unauthenticated STAMP test packets, defined in Section 4.2.1 and Section 4.3.1, ensure interworking between STAMP and TWAMP Light as described in Section 4.6 packet formats.

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

#### 4.1. UDP Port Numbers in STAMP Testing

A STAMP Session-Sender MUST use UDP port 862 (TWAMP-Test Receiver Port) as the default destination UDP port number. A STAMP implementation of Session-Sender MUST be able to use as the destination UDP port numbers from User, a.k.a. Registered, Ports and Dynamic, a.k.a. Private or Ephemeral, Ports ranges defined in [RFC6335]. Before using numbers from the User Ports range, the possible impact on the network MUST be carefully studied and agreed by all users of the network domain where the test has been planned.

An implementation of STAMP Session-Reflector by default MUST receive STAMP test packets on UDP port 862. An implementation of Session-Reflector that supports this specification MUST be able to define the port number to receive STAMP test packets from User Ports and Dynamic Ports ranges that are defined in [RFC6335]. STAMP defines two different test packet formats, one for packets transmitted by the STAMP-Session-Sender and one for packets transmitted by the STAMP-Session-Reflector.

#### 4.2. Session-Sender Behavior and Packet Format

A STAMP Session-Reflector supports the symmetrical size of test packets, as defined in Section 3 [RFC6038], as the default behavior. A reflected test packet includes more information and thus is larger. Because of that, the base STAMP Session-Sender packet is padded to match the size of a reflected STAMP test packet. Hence, the base STAMP Session-Sender packet has a minimum size of 44 octets in unauthenticated mode, see Figure 2, and 112 octets in the authenticated mode, see Figure 4. The variable length of a test packet in STAMP is supported by using Extra Padding TLV defined in [I-D.ietf-ippm-stamp-option-tlv].

##### 4.2.1. Session-Sender Packet Format in Unauthenticated Mode

STAMP Session-Sender packet format in unauthenticated mode:

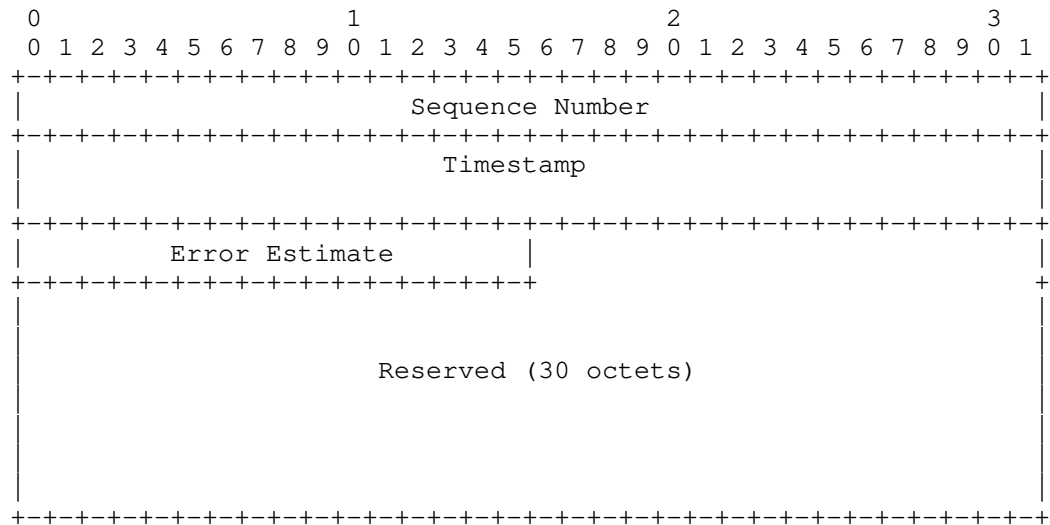


Figure 2: STAMP Session-Sender test packet format in unauthenticated mode

where fields are defined as the following:

- o Sequence Number is four octets long field. For each new session its value starts at zero and is incremented with each transmitted packet.
- o Timestamp is eight octets long field. STAMP node MUST support Network Time Protocol (NTP) version 4 64-bit timestamp format [RFC5905], the format used in [RFC5357]. STAMP node MAY support IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE.1588.2008], the format used in [RFC8186]. The use of the specific format, NTP or PTP, is part of configuration of the Session-Sender or the particular test session.
- o Error Estimate is two octets long field with format displayed in Figure 3

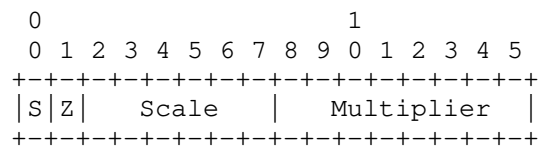


Figure 3: Error Estimate Format



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

- \* 0 - NTP 64 bit format of a timestamp;
- \* 1 - PTPv2 truncated format of a timestamp.

The default behavior of the STAMP Session-Sender and Session-Reflector is to use the NTP 64-bit timestamp format (Z field value of 0). An operator, using configuration/management function, MAY configure STAMP Session-Sender and Session-Reflector to using the PTPv2 truncated format of a timestamp (Z field value of 1). Note, that an implementation of a Session-Sender that supports this specification MAY be configured to use PTPv2 format of a timestamp even though the Session-Reflector is configured to use NTP format.

- o Reserved field in the Session-Sender unauthenticated packet is 30 octets long. It MUST be all zeroed on the transmission and MUST be ignored on receipt.

#### 4.2.2. Session-Sender Packet Format in Authenticated Mode

STAMP Session-Sender packet format in authenticated mode:

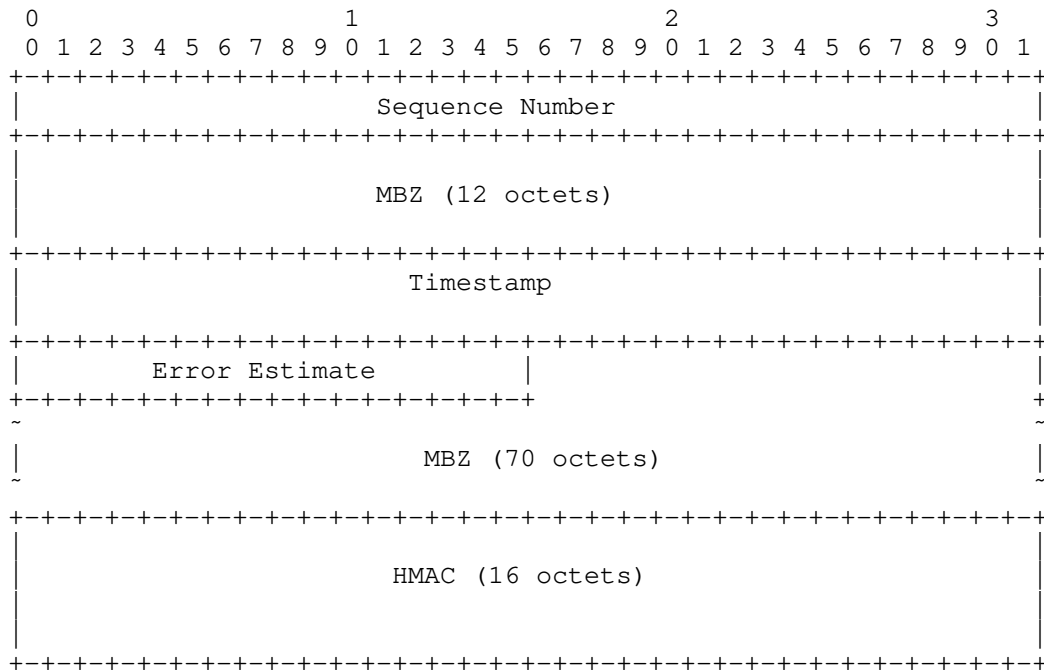


Figure 4: STAMP Session-Sender test packet format in authenticated mode

The field definitions are the same as the unauthenticated mode, listed in Section 4.2.1. Also, Must-Be-Zero (MBZ) fields are used to make the packet length a multiple of 16 octets. The value of the field MUST be zeroed on transmission and MUST be ignored on receipt. Note, that the MBZ field is used to calculate a key-hashed message authentication code (HMAC) ([RFC2104]) hash. Also, the packet includes HMAC hash at the end of the PDU. The detailed use of the HMAC field is described in Section 4.4.

#### 4.3. Session-Reflector Behavior and Packet Format

The Session-Reflector receives the STAMP test packet and verifies it. If the base STAMP test packet validated, the Session-Reflector, that supports this specification, prepares and transmits the reflected test packet symmetric to the packet received from the Session-Sender copying the content beyond the size of the base STAMP packet (see Section 4.2).

## 4.3.1. Session-Reflector Packet Format in Unauthenticated Mode

For unauthenticated mode:

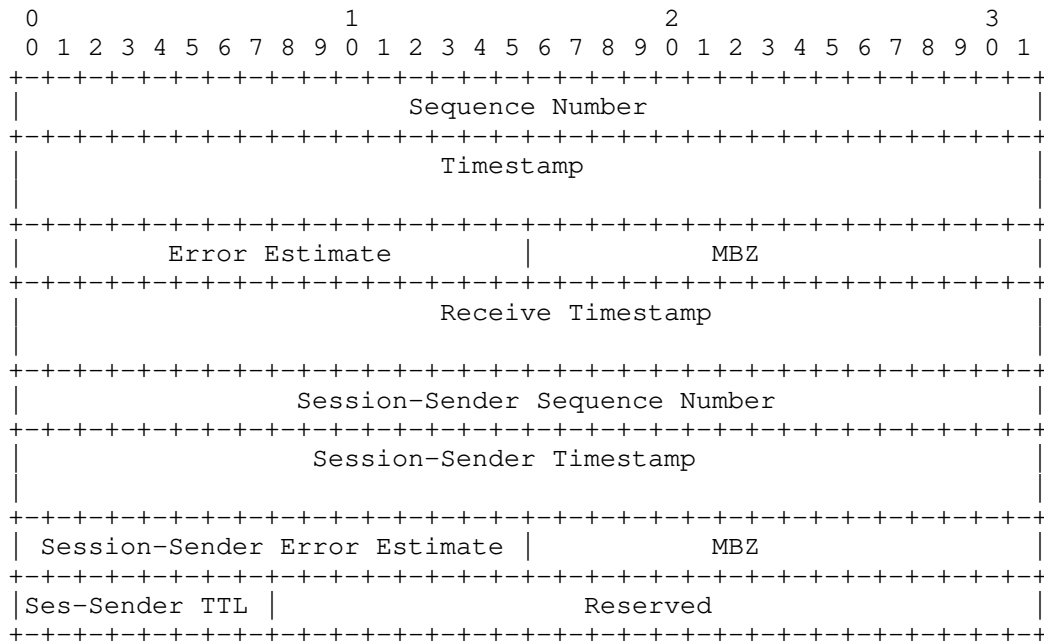


Figure 5: STAMP Session-Reflector test packet format in unauthenticated mode

where fields are defined as the following:

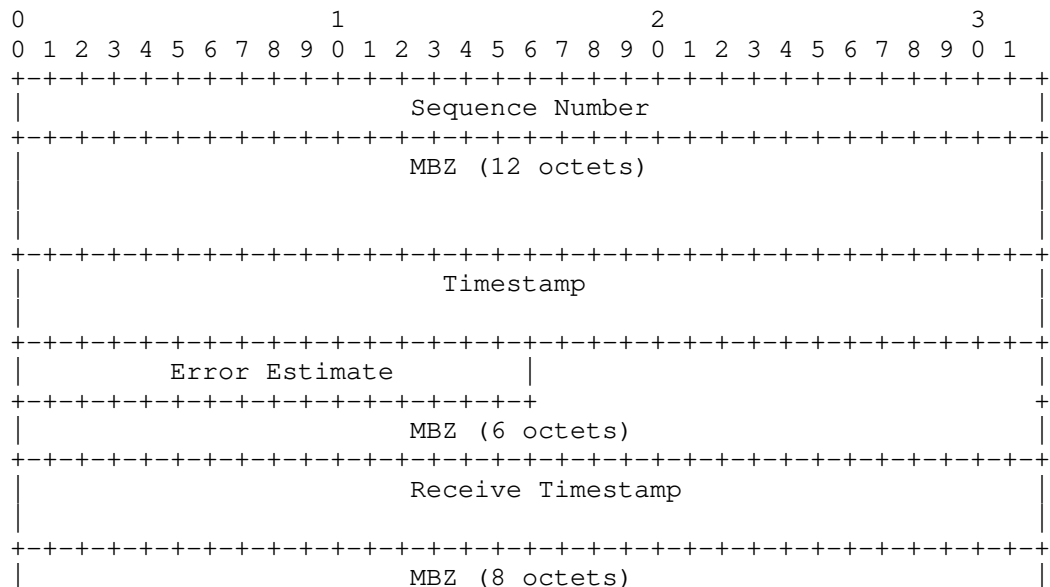
- o Sequence Number is four octets long field. The value of the Sequence Number field is set according to the mode of the STAMP Session-Reflector:
  - \* in the stateless mode, the Session-Reflector copies the value from the received STAMP test packet's Sequence Number field;
  - \* in the stateful mode, the Session-Reflector counts the transmitted STAMP test packets. It starts with zero and is incremented by one for each subsequent packet for each test session. The Session-Reflector uses that counter to set the value of the Sequence Number field.
- o Timestamp and Receive Timestamp fields are each eight octets long. The format of these fields, NTP or PTPv2, indicated by the Z field of the Error Estimate field as described in Section 4.2. Receive

Timestamp is the time the test packet was received by the Session-Reflector. Timestamp – the time taken by the Session-Reflector at the start of transmitting the test packet.

- o Error Estimate has the same size and interpretation as described in Section 4.2. It is applicable to both Timestamp and Receive Timestamp.
- o Session-Sender Sequence Number, Session-Sender Timestamp, and Session-Sender Error Estimate are copies of the corresponding fields in the STAMP test packet sent by the Session-Sender.
- o Session-Sender TTL is one octet long field, and its value is the copy of the TTL field in IPv4 (or Hop Limit in IPv6) from the received STAMP test packet.
- o MBZ is used to achieve alignment of fields within the packet on a four octets boundary. The value of the field MUST be zeroed on transmission and MUST be ignored on receipt.
- o Reserved field in the Session-Reflector unauthenticated packet is three octets long. It MUST be all zeroed on the transmission and MUST be ignored on receipt.

#### 4.3.2. Session-Reflector Packet Format in Authenticated Mode

For the authenticated mode:



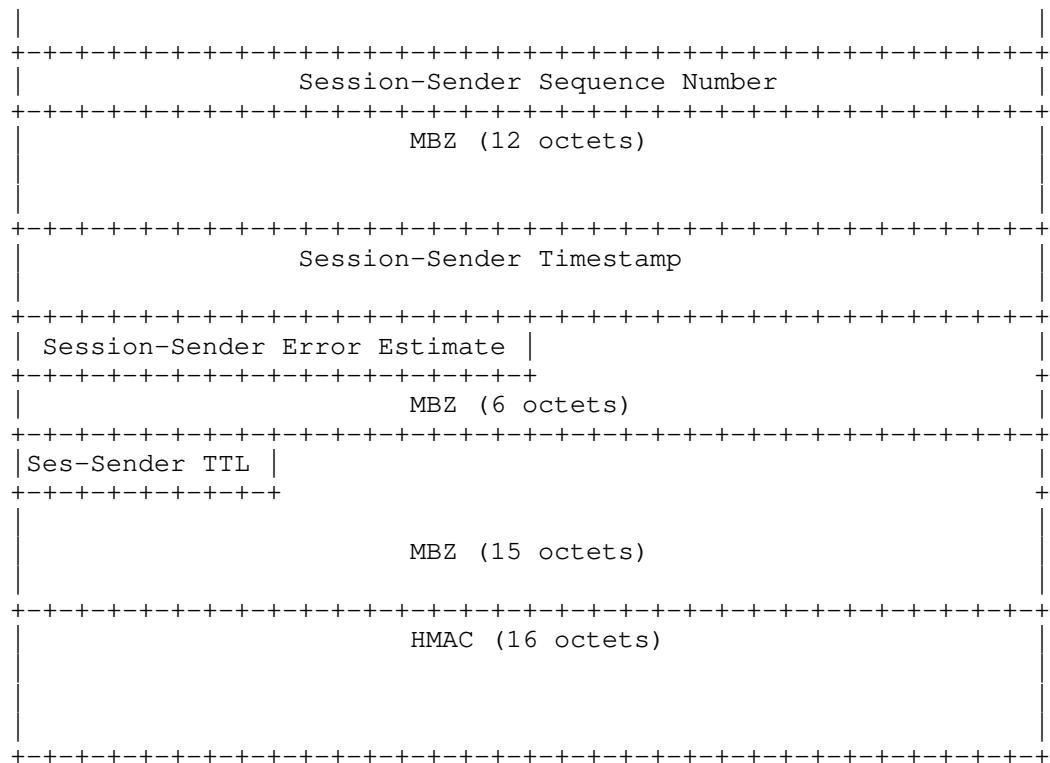


Figure 6: STAMP Session-Reflector test packet format in authenticated mode

The field definitions are the same as the unauthenticated mode, listed in Section 4.3.1. Additionally, the MBZ field is used to make the packet length a multiple of 16 octets. The value of the field MUST be zeroed on transmission and MUST be ignored on receipt. Note, that the MBZ field is used to calculate HMAC hash value. Also, STAMP Session-Reflector test packet format in authenticated mode includes HMAC ([RFC2104]) hash at the end of the PDU. The detailed use of the HMAC field is in Section 4.4.

#### 4.4. Integrity Protection in STAMP

Authenticated mode provides integrity protection to each STAMP message by adding Hashed Message Authentication Code (HMAC). STAMP uses HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPsec defined in [RFC4868]); hence the length of the HMAC field is 16 octets. In the Authenticated mode, HMAC covers the first six blocks (96 octets). HMAC uses its own key that may be unique for

each STAMP test session; key management and the mechanisms to distribute the HMAC key are outside the scope of this specification. One example is to use an orchestrator to configure HMAC key based on STAMP YANG data model [I-D.ietf-ippm-stamp-yang]. HMAC MUST be verified as early as possible to avoid using or propagating corrupted data.

Future specifications may define the use of other, more advanced cryptographic algorithms, possibly providing an update to the STAMP YANG data model [I-D.ietf-ippm-stamp-yang].

#### 4.5. Confidentiality Protection in STAMP

If confidentiality protection for STAMP is required, a STAMP test session MUST use a secured transport. For example, STAMP packets could be transmitted in the dedicated IPsec tunnel or share the IPsec tunnel with the monitored flow. Also, Datagram Transport Layer Security protocol would provide the desired confidentiality protection.

#### 4.6. Interoperability with TWAMP Light

One of the essential requirements to STAMP is the ability to interwork with a TWAMP Light device. Because STAMP and TWAMP use different algorithms in Authenticated mode (HMAC-SHA-256 vs. HMAC-SHA-1), interoperability is only considered for Unauthenticated mode. There are two possible combinations for such use case:

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

In the former case, the Session-Sender might not be aware that its Session-Reflector does not support STAMP. For example, a TWAMP Light Session-Reflector may not support the use of UDP port 862 as specified in [RFC8545]. Thus Section 4. permits a STAMP Session-Sender to use alternative ports. If any of STAMP extensions are used, the TWAMP Light Session-Reflector will view them as Packet Padding field.

In the latter scenario, if a TWAMP Light Session-Sender does not support the use of UDP port 862, the test management system MUST set STAMP Session-Reflector to use UDP port number, as permitted by Section 4. The Session-Reflector MUST be set to use the default format for its timestamps, NTP.

A STAMP Session-Reflector that supports this specification will transmit the base packet (Figure 5) if it receives a packet smaller

than the STAMP base packet. If the packet received from TWAMP Session-Sender is larger than the STAMP base packet, the STAMP Session-Reflector that supports this specification will copy the content of the remainder of the received packet to transmit reflected packet of symmetrical size.

## 5. Operational Considerations

STAMP is intended to be used on production networks to enable the operator to assess service level agreements based on packet delay, delay variation, and loss. When using STAMP over the Internet, especially when STAMP test packets are transmitted with the destination UDP port number from the User Ports range, the possible impact of the STAMP test packets MUST be thoroughly analyzed. The use of STAMP for each case MUST be agreed by users of nodes hosting the Session-Sender and Session-Reflector before starting the STAMP test session.

Also, the use of the well-known port number as the destination UDP port number in STAMP test packets transmitted by a Session-Sender would not impede the ability to measure performance in an Equal Cost Multipath environment and analysis in Section 5.3 [RFC8545] fully applies to STAMP.

## 6. IANA Considerations

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

## 7. Security Considerations

[RFC5357] does not identify security considerations specific to TWAMP-Test but refers to security considerations identified for OWAMP in [RFC4656]. Since both OWAMP and TWAMP include control plane and data plane components, only security considerations related to OWAMP-Test, discussed in Sections 6.2, 6.3 [RFC4656] apply to STAMP.

STAMP uses the well-known UDP port number allocated for the OWAMP-Test/TWAMP-Test Receiver port. Thus the security considerations and measures to mitigate the risk of the attack using the registered port number documented in Section 6 [RFC8545] equally apply to STAMP. Because of the control and management of a STAMP test being outside the scope of this specification only the more general requirement is set:

To mitigate the possible attack vector, the control, and management of a STAMP test session MUST use the secured transport.

The load of the STAMP test packets offered to a network MUST be carefully estimated, and the possible impact on the existing services MUST be thoroughly analyzed before launching the test session. [RFC8085] section 3.1.5 provides guidance on handling network load for UDP-based protocol. While the characteristic of test traffic depends on the test objective, it is highly recommended to stay in the limits as provided in [RFC8085].

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

## 8. Acknowledgments

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

## 9. References

### 9.1. Normative References

- [I-D.ietf-ippm-stamp-option-tlv]  
Mirsky, G., Xiao, M., Jun, G., Nydell, H., Foote, R., and A. Masputra, "Simple Two-way Active Measurement Protocol Optional Extensions", draft-ietf-ippm-stamp-option-tlv-01 (work in progress), September 2019.
- [IEEE.1588.2008]  
"Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588, March 2008.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.



- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <<https://www.rfc-editor.org/info/rfc6038>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <<https://www.rfc-editor.org/info/rfc8186>>.
- [RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019, <<https://www.rfc-editor.org/info/rfc8545>>.

## 9.2. Informative References

- [BBF.TR-390]  
"Performance Measurement from IP Edge to Customer Equipment using TWAMP Light", BBF TR-390, May 2017.
- [I-D.ietf-ippm-stamp-yang]  
Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-05 (work in progress), October 2019.

- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <<https://www.rfc-editor.org/info/rfc4868>>.
- [RFC7750] Hedin, J., Mirsky, G., and S. Baillargeon, "Differentiated Service Code Point and Explicit Congestion Notification Monitoring in the Two-Way Active Measurement Protocol (TWAMP)", RFC 7750, DOI 10.17487/RFC7750, February 2016, <<https://www.rfc-editor.org/info/rfc7750>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.

## Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

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

Phone: +86 18105183663  
Email: [guo.jun2@zte.com.cn](mailto:guo.jun2@zte.com.cn)

Henrik Nydell  
Accedian Networks

Email: [hnydell@accedian.com](mailto:hnydell@accedian.com)

Richard Foote  
Nokia

Email: [footer.foote@nokia.com](mailto:footer.foote@nokia.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 13 January 2022

G. Mirsky  
X. Min  
ZTE Corp.  
W.S. Luo  
Ericsson  
12 July 2021

Simple Two-way Active Measurement Protocol (STAMP) Data Model  
draft-ietf-ippm-stamp-yang-09

Abstract

This document specifies the data model for implementations of Session-Sender and Session-Reflector for Simple Two-way Active Measurement Protocol (STAMP) mode using YANG.

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 13 January 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions used in this document . . . . .	2
1.1.1. Requirements Language . . . . .	3
2. Scope, Model, and Applicability . . . . .	3
2.1. Data Model Parameters . . . . .	3
2.1.1. STAMP-Sender . . . . .	3
2.1.2. STAMP-Reflector . . . . .	4
3. Data Model . . . . .	4
3.1. Tree Diagrams . . . . .	5
3.2. YANG Module . . . . .	10
4. IANA Considerations . . . . .	31
5. Security Considerations . . . . .	32
6. Acknowledgments . . . . .	33
7. References . . . . .	33
7.1. Normative References . . . . .	33
7.2. Informative References . . . . .	34
Appendix A. Example of STAMP Session Configuration . . . . .	35
Authors' Addresses . . . . .	36

## 1. Introduction

The Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] can be used to measure performance parameters of IP networks such as latency, jitter, and packet loss by sending test packets and monitoring their experience in the network. The STAMP protocol [RFC8762] in unauthenticated mode is on-wire compatible with TWAMP Light, discussed in Appendix I [RFC5357]. The TWAMP Light is known to have many implementations though no common management framework being defined, thus leaving some aspects of test packet processing to interpretation. As one of the goals of STAMP is to support these variations, this document presents their analysis; describes the data model of the base STAMP specification. The defined STAMP data model can be augmented to include STAMP extensions, for example, described in [RFC8972]. This document defines the STAMP data model and specifies it formally, using the YANG data modeling language [RFC7950].

This version of the interfaces data model conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

## 1.1. Conventions used in this document

### 1.1.1. 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, Model, and Applicability

The scope of this document includes a model of the STAMP as defined in [RFC8762] and Section 3 [RFC8972].

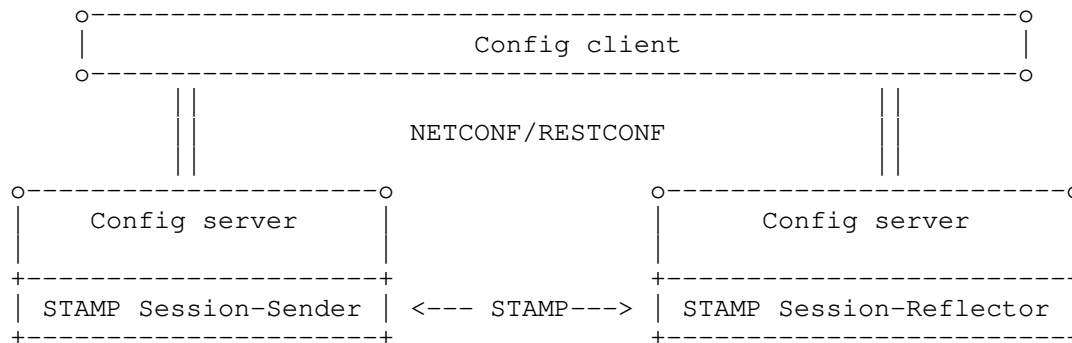


Figure 1: STAMP Reference Model

### 2.1. Data Model Parameters

This section describes containers within the STAMP data model.

#### 2.1.1. STAMP-Sender

The stamp-session-sender container holds items that are related to the configuration of the stamp Session-Sender logical entity.

The stamp-session-sender-state container holds information about the state of the particular STAMP test session.

RPCs stamp-sender-start and stamp-sender-stop respectively start and stop the referenced session by the stamp-session-id of the STAMP.

#### 2.1.1.1. Controls for Test Session and Performance Metric Calculation

The data model supports several scenarios for a STAMP Session-Sender to execute test sessions and calculate performance metrics:

- \* The test mode in which the test packets are sent unbound in time as defined by the parameter 'interval' in the stamp-session-sender container frequency is referred to as continuous mode. Performance metrics in the continuous mode are calculated at a period defined by the parameter 'measurement-interval'.
- \* The test mode that has a specific number of the test packets configured for the test session in the 'number-of-packets' parameter is referred to as a periodic mode. The STAMP-Sender MAY repeat the test session with the same parameters. The 'repeat' parameter defines the number of tests and the 'repeat-interval' - the interval between the consecutive tests. The performance metrics are calculated after each test session when the interval defined by the 'session-timeout' expires.

#### 2.1.2. STAMP-Reflector

The stamp-session-reflector container holds items that are related to the configuration of the STAMP Session-Reflector logical entity.

The stamp-session-refl-state container holds Session-Reflector state data for the particular STAMP test session.

### 3. Data Model

Creating the STAMP data model presents several challenges, and among them is the identification of a test-session at Session-Reflector. A Session-Reflector MAY require only as little as the STAMP Session Identifier (SSID) and the source IP address in received STAMP-Test packet to spawn a new test session. More so, to test processing of Class-of-Service along the same route in Equal Cost Multi-Path environment Session-Sender may perform STAMP test sessions concurrently using the same source IP address, source UDP port number, destination IP address, and destination UDP port number. Thus the only parameter that can be used to differentiate these test sessions would be DSCP value. The DSCP field may get re-marked along the path, and without the use of Class of Service TLV (Section 4.4 [RFC8972]) that will go undetected, but by using SSID and the source IP address as a key, we can ensure that STAMP test packets that are considered as different test sessions follow the same path even in ECMP environments.

### 3.1. Tree Diagrams

This section presents a simplified graphical representation of the STAMP data model using a YANG tree diagram [RFC8340].

```

module: ietf-stamp
+--rw stamp
|   +--rw stamp-session-sender {session-sender}?
|   |   +--rw sender-enable?    boolean
|   |   +--rw sender-test-session* [stamp-session-id]
|   |   |   +--rw test-session-enable?    boolean
|   |   |   +--rw number-of-packets?      union
|   |   |   +--rw interval?              uint32
|   |   |   +--rw session-timeout?        uint32
|   |   |   +--rw measurement-interval?   uint32
|   |   |   +--rw repeat?                union
|   |   |   +--rw repeat-interval?        uint32
|   |   |   +--rw dscp-value?             inet:dscp
|   |   |   +--rw test-session-reflector-mode? session-reflector-mode
|   |   |   +--rw sender-ip              inet:ip-address
|   |   |   +--rw session-sender-udp-port inet:port-number
|   |   |   +--rw stamp-session-id        uint32
|   |   |   +--rw session-reflector-ip    inet:ip-address
|   |   |   +--rw session-reflector-udp-port? inet:port-number
|   |   |   +--rw sender-timestamp-format? timestamp-format
|   |   |   +--rw security! {stamp-security}?
|   |   |   |   +--rw key-chain?    kc:key-chain-ref
|   |   |   +--rw first-percentile? percentile
|   |   |   +--rw second-percentile? percentile
|   |   |   +--rw third-percentile? percentile
|   +--rw stamp-session-reflector {session-reflector}?
|   |   +--rw reflector-enable?    boolean
|   |   +--rw ref-wait?            uint32
|   |   +--rw reflector-mode-state? session-reflector-mode
|   |   +--rw reflector-test-session* [stamp-session-id]
|   |   |   +--rw stamp-session-id        union
|   |   |   +--rw dscp-handling-mode?      session-dscp-mode
|   |   |   +--rw dscp-value?             inet:dscp
|   |   |   +--rw sender-ip?              union
|   |   |   +--rw sender-udp-port?         union
|   |   |   +--rw reflector-ip?           union
|   |   |   +--rw reflector-udp-port?      inet:port-number
|   |   |   +--rw reflector-timestamp-format? timestamp-format
|   |   +--rw security! {stamp-security}?
|   |   |   +--rw key-chain?    kc:key-chain-ref

```

Figure 2: STAMP Configuration Tree Diagram

```

module: ietf-stamp
  +--ro stamp-state
    +--ro stamp-session-sender-state {session-sender}?
      +--ro test-session-state* [stamp-session-id]
        +--ro stamp-session-id          uint32
        +--ro sender-session-state?     enumeration
      +--ro current-stats
        +--ro start-time                yang:date-and-time
        +--ro interval?                 uint32
        +--ro duplicate-packets?        uint32
        +--ro reordered-packets?        uint32
        +--ro sender-timestamp-format?  timestamp-format
        +--ro reflector-timestamp-format? timestamp-format
        +--ro dscp?                     inet:dscp
      +--ro two-way-delay
        +--ro delay
          +--ro min?    yang:gauge64
          +--ro max?    yang:gauge64
          +--ro avg?    yang:gauge64
        +--ro delay-variation
          +--ro min?    yang:gauge32
          +--ro max?    yang:gauge32
          +--ro avg?    yang:gauge32
      +--ro one-way-delay-far-end
        +--ro delay
          +--ro min?    yang:gauge64
          +--ro max?    yang:gauge64
          +--ro avg?    yang:gauge64
        +--ro delay-variation
          +--ro min?    yang:gauge32
          +--ro max?    yang:gauge32
          +--ro avg?    yang:gauge32
      +--ro one-way-delay-near-end
        +--ro delay
          +--ro min?    yang:gauge64
          +--ro max?    yang:gauge64
          +--ro avg?    yang:gauge64
        +--ro delay-variation
          +--ro min?    yang:gauge32
          +--ro max?    yang:gauge32
          +--ro avg?    yang:gauge32
      +--ro low-percentile
        +--ro delay-percentile
          +--ro rtt-delay?    yang:gauge64
          +--ro near-end-delay? yang:gauge64

```



```

| | +--ro far-end-delay? yang:gauge64
| | +--ro delay-variation-percentile
| | | +--ro rtt-delay-variation? yang:gauge32
| | | +--ro near-end-delay-variation? yang:gauge32
| | | +--ro far-end-delay-variation? yang:gauge32
+--ro mid-percentile
| | +--ro delay-percentile
| | | +--ro rtt-delay? yang:gauge64
| | | +--ro near-end-delay? yang:gauge64
| | | +--ro far-end-delay? yang:gauge64
| | +--ro delay-variation-percentile
| | | +--ro rtt-delay-variation? yang:gauge32
| | | +--ro near-end-delay-variation? yang:gauge32
| | | +--ro far-end-delay-variation? yang:gauge32
+--ro high-percentile
| | +--ro delay-percentile
| | | +--ro rtt-delay? yang:gauge64
| | | +--ro near-end-delay? yang:gauge64
| | | +--ro far-end-delay? yang:gauge64
| | +--ro delay-variation-percentile
| | | +--ro rtt-delay-variation? yang:gauge32
| | | +--ro near-end-delay-variation? yang:gauge32
| | | +--ro far-end-delay-variation? yang:gauge32
+--ro two-way-loss
| | +--ro loss-count? int32
| | +--ro loss-ratio? percentage
| | +--ro loss-burst-max? int32
| | +--ro loss-burst-min? int32
| | +--ro loss-burst-count? int32
+--ro one-way-loss-far-end
| | +--ro loss-count? int32
| | +--ro loss-ratio? percentage
| | +--ro loss-burst-max? int32
| | +--ro loss-burst-min? int32
| | +--ro loss-burst-count? int32
+--ro one-way-loss-near-end
| | +--ro loss-count? int32
| | +--ro loss-ratio? percentage
| | +--ro loss-burst-max? int32
| | +--ro loss-burst-min? int32
| | +--ro loss-burst-count? int32
+--ro sender-ip inet:ip-address
+--ro session-sender-udp-port inet:port-number
+--ro session-reflector-ip inet:ip-address
+--ro session-reflector-udp-port? inet:port-number
+--ro sent-packets? uint32
+--ro rcv-packets? uint32
+--ro sent-packets-error? uint32

```

```

|   +--ro rcv-packets-error?          uint32
|   +--ro last-sent-seq?              uint32
|   +--ro last-rcv-seq?              uint32
+--ro history-stats* [stamp-session-id]
|   +--ro stamp-session-id            uint32
|   +--ro end-time                    yang:date-and-time
|   +--ro interval?                  uint32
|   +--ro duplicate-packets?         uint32
|   +--ro reordered-packets?         uint32
|   +--ro sender-timestamp-format?   timestamp-format
|   +--ro reflector-timestamp-format? timestamp-format
|   +--ro dscp?                      inet:dscp
+--ro two-way-delay
|   +--ro delay
|   |   +--ro min?    yang:gauge64
|   |   +--ro max?    yang:gauge64
|   |   +--ro avg?    yang:gauge64
|   +--ro delay-variation
|   |   +--ro min?    yang:gauge32
|   |   +--ro max?    yang:gauge32
|   |   +--ro avg?    yang:gauge32
+--ro one-way-delay-far-end
|   +--ro delay
|   |   +--ro min?    yang:gauge64
|   |   +--ro max?    yang:gauge64
|   |   +--ro avg?    yang:gauge64
|   +--ro delay-variation
|   |   +--ro min?    yang:gauge32
|   |   +--ro max?    yang:gauge32
|   |   +--ro avg?    yang:gauge32
+--ro one-way-delay-near-end
|   +--ro delay
|   |   +--ro min?    yang:gauge64
|   |   +--ro max?    yang:gauge64
|   |   +--ro avg?    yang:gauge64
|   +--ro delay-variation
|   |   +--ro min?    yang:gauge32
|   |   +--ro max?    yang:gauge32
|   |   +--ro avg?    yang:gauge32
+--ro low-percentile
|   +--ro delay-percentile
|   |   +--ro rtt-delay?      yang:gauge64
|   |   +--ro near-end-delay? yang:gauge64
|   |   +--ro far-end-delay?  yang:gauge64
|   +--ro delay-variation-percentile
|   |   +--ro rtt-delay-variation? yang:gauge32
|   |   +--ro near-end-delay-variation? yang:gauge32
|   |   +--ro far-end-delay-variation? yang:gauge32

```

```

+--ro mid-percentile
|   +--ro delay-percentile
|   |   +--ro rtt-delay?          yang:gauge64
|   |   +--ro near-end-delay?    yang:gauge64
|   |   +--ro far-end-delay?     yang:gauge64
|   +--ro delay-variation-percentile
|   |   +--ro rtt-delay-variation? yang:gauge32
|   |   +--ro near-end-delay-variation? yang:gauge32
|   |   +--ro far-end-delay-variation? yang:gauge32
+--ro high-percentile
|   +--ro delay-percentile
|   |   +--ro rtt-delay?          yang:gauge64
|   |   +--ro near-end-delay?    yang:gauge64
|   |   +--ro far-end-delay?     yang:gauge64
|   +--ro delay-variation-percentile
|   |   +--ro rtt-delay-variation? yang:gauge32
|   |   +--ro near-end-delay-variation? yang:gauge32
|   |   +--ro far-end-delay-variation? yang:gauge32
+--ro two-way-loss
|   +--ro loss-count?            int32
|   +--ro loss-ratio?            percentage
|   +--ro loss-burst-max?        int32
|   +--ro loss-burst-min?        int32
|   +--ro loss-burst-count?      int32
+--ro one-way-loss-far-end
|   +--ro loss-count?            int32
|   +--ro loss-ratio?            percentage
|   +--ro loss-burst-max?        int32
|   +--ro loss-burst-min?        int32
|   +--ro loss-burst-count?      int32
+--ro one-way-loss-near-end
|   +--ro loss-count?            int32
|   +--ro loss-ratio?            percentage
|   +--ro loss-burst-max?        int32
|   +--ro loss-burst-min?        int32
|   +--ro loss-burst-count?      int32
+--ro sender-ip                  inet:ip-address
+--ro session-sender-udp-port    inet:port-number
+--ro session-reflector-ip       inet:ip-address
+--ro session-reflector-udp-port? inet:port-number
+--ro sent-packets?              uint32
+--ro rcv-packets?               uint32
+--ro sent-packets-error?        uint32
+--ro rcv-packets-error?         uint32
+--ro last-sent-seq?             uint32
+--ro last-rcv-seq?             uint32
+--ro stamp-session-refl-state {session-reflector}?
|   +--ro reflector-light-admin-status? boolean

```

```

+---ro test-session-state* [stamp-session-id]
  +---ro stamp-session-id          uint32
  +---ro reflector-timestamp-format? timestamp-format
  +---ro sender-ip                  inet:ip-address
  +---ro session-sender-udp-port    inet:port-number
  +---ro session-reflector-ip       inet:ip-address
  +---ro session-reflector-udp-port? inet:port-number
  +---ro sent-packets?              uint32
  +---ro rcv-packets?               uint32
  +---ro sent-packets-error?        uint32
  +---ro rcv-packets-error?         uint32
  +---ro last-sent-seq?             uint32
  +---ro last-rcv-seq?              uint32

```

Figure 3: STAMP State Tree Diagram

```

rpcs:
+---x stamp-sender-start
|   +---w input
|       +---w stamp-session-id    uint32
+---x stamp-sender-stop
    +---w input
        +---w stamp-stamp-session-id    uint32

```

Figure 4: STAMP RPC Tree Diagram

### 3.2. YANG Module

```

<CODE BEGINS> file "ietf-stamp@2021-07-12.yang"
module ietf-stamp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-stamp";
  //namespace need to be assigned by IANA
  prefix "ietf-stamp";

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference "RFC 6991: Common YANG Types.";
  }
  import ietf-key-chain {
    prefix kc;
    reference "RFC 8177: YANG Data Model for Key Chains.";
  }
}

```

## organization

"IETF IPPM (IP Performance Metrics) Working Group";

## contact

"WG Web: <http://tools.ietf.org/wg/ippm/>  
WG List: [ippm@ietf.org](mailto:ippm@ietf.org)

Editor: Greg Mirsky  
[gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Editor: Xiao Min  
[xiao.min2@zte.com.cn](mailto:xiao.min2@zte.com.cn)

Editor: Wei S Luo  
[wei.s.luo@ericsson.com](mailto:wei.s.luo@ericsson.com)";

## description

"This YANG module specifies a vendor-independent model  
for the Simple Two-way Active Measurement Protocol (STAMP).

The data model covers two STAMP logical entities -  
Session-Sender and Session-Reflector; characteristics  
of the STAMP test session, as well as measured and  
calculated performance metrics.

Copyright (c) 2021 IETF Trust and the persons identified as  
the document authors. 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 "2021-07-10" {

description

"Initial Revision. Base STAMP specification is covered";

reference

"RFC XXXX: STAMP YANG Data Model.";

}

/\*

\* Typedefs

\*/

typedef session-reflector-mode {

type enumeration {

enum stateful {

```
    description
        "When the Session-Reflector is stateful,
        i.e. is aware of STAMP-Test session state.";
    }
    enum stateless {
        description
            "When the Session-Reflector is stateless,
            i.e. is not aware of the state of
            STAMP-Test session.";
    }
}
description "State of the Session-Reflector";
reference
    "RFC 8762 Simple Two-way Active
    Measurement Protocol (STAMP) Section 4.";
}

typedef session-dscp-mode {
    type enumeration {
        enum copy-received-value {
            description
                "Use DSCP value copied from received
                STAMP test packet of the test session.";
        }
        enum use-configured-value {
            description
                "Use DSCP value configured for this
                test session on the Session-Reflector.";
        }
    }
}
description
    "DSCP handling mode by Session-Reflector.";
}

typedef timestamp-format {
    type enumeration {
        enum ntp-format {
            description
                "NTP 64 bit format of a timestamp";
        }
        enum ptp-format {
            description
                "PTPv2 truncated format of a timestamp";
        }
    }
}
description
    "Timestamp format used by Session-Sender
    or Session-Reflector.";
```

```
    reference
      "RFC 8762 Simple Two-way Active
      Measurement Protocol (STAMP) Section 4.2.1.";
  }

  typedef percentage {
    type decimal64 {
      fraction-digits 5;
    }
    description "Percentage";
  }

  typedef percentile {
    type decimal64 {
      fraction-digits 5;
    }
    description
      "Percentile is a measure used in statistics
      indicating the value below which a given
      percentage of observations in a group of
      observations fall.";
  }

  /*
   * Feature definitions.
   */
  feature session-sender {
    description
      "This feature relates to the device functions as the
      STAMP Session-Sender";
    reference
      "RFC 8762 Simple Two-way Active
      Measurement Protocol (STAMP) Section 4.2.";
  }

  feature session-reflector {
    description
      "This feature relates to the device functions as the
      STAMP Session-Reflector";
    reference
      "RFC 8762 Simple Two-way Active
      Measurement Protocol (STAMP) Section 4.3.";
  }

  feature stamp-security {
    description "Secure STAMP supported";
    reference
```

```
    "RFC 8762 Simple Two-way Active
      Measurement Protocol (STAMP) Section 4.4.";
  }

  /*
   * Reusable node groups
   */

  grouping maintenance-statistics {
    description "Maintenance statistics grouping";
    leaf sent-packets {
      type uint32;
      description "Packets sent";
    }
    leaf rcv-packets {
      type uint32;
      description "Packets received";
    }
    leaf sent-packets-error {
      type uint32;
      description "Packets sent error";
    }
    leaf rcv-packets-error {
      type uint32;
      description "Packets received error";
    }
    leaf last-sent-seq {
      type uint32;
      description "Last sent sequence number";
    }
    leaf last-rcv-seq {
      type uint32;
      description "Last received sequence number";
    }
  }

  grouping test-session-statistics {
    description
      "Performance metrics calculated for
       a STAMP test session.";

    leaf interval {
      type uint32;
      units microseconds;
      description
        "Time interval between transmission of two
         consecutive packets in the test session";
    }
  }
```



```
leaf duplicate-packets {
  type uint32;
  description "Duplicate packets";
}

leaf reordered-packets {
  type uint32;
  description "Reordered packets";
}

leaf sender-timestamp-format {
  type timestamp-format;
  description "Sender Timestamp format";
}

leaf reflector-timestamp-format {
  type timestamp-format;
  description "Reflector Timestamp format";
}

leaf dscp {
  type inet:dscp;
  description
    "The DSCP value that was placed in the header of
    STAMP UDP test packets by the Session-Sender.";
}

container two-way-delay {
  description
    "two way delay result of the test session";
  uses delay-statistics;
}

container one-way-delay-far-end {
  description
    "one way delay far-end of the test session";
  uses delay-statistics;
}

container one-way-delay-near-end {
  description
    "one way delay near-end of the test session";
  uses delay-statistics;
}

container low-percentile {
  when "/stamp/stamp-session-sender/"
    +"sender-test-session[stamp-session-id]/"
```

```
    +"first-percentile != '0.00'" {
      description
        "Only valid if the
         the first-percentile is not NULL";
    }
  description
    "Low percentile report";
  uses time-percentile-report;
}

container mid-percentile {
  when "/stamp/stamp-session-sender/"
    +"sender-test-session[stamp-session-id]/"
    +"second-percentile != '0.00'" {
    description
      "Only valid if the
       the first-percentile is not NULL";
  }
  description
    "Mid percentile report";
  uses time-percentile-report;
}

container high-percentile {
  when "/stamp/stamp-session-sender/"
    +"sender-test-session[stamp-session-id]/"
    +"third-percentile != '0.00'" {
    description
      "Only valid if the
       the first-percentile is not NULL";
  }
  description
    "High percentile report";
  uses time-percentile-report;
}

container two-way-loss {
  description
    "Two way loss count and ratio result of
     the test session";
  uses packet-loss-statistics;
}

container one-way-loss-far-end {
  when "/stamp/stamp-session-sender/"
    +"sender-test-session[stamp-session-id]/"
    +"test-session-reflector-mode = 'stateful'" {
    description
```

```
        "One-way statistic is only valid if the
        session-reflector is in stateful mode.";
    }
    description
        "One way loss count and ratio far-end of
        the test session";
    uses packet-loss-statistics;
}

container one-way-loss-near-end {
    when "/stamp/stamp-session-sender/"
        +"sender-test-session[stamp-session-id]/"
        +"test-session-reflector-mode = 'stateful'" {
        description
            "One-way statistic is only valid if the
            session-reflector is in stateful mode.";
        }
        description
            "One way loss count and ratio near-end of
            the test session";
        uses packet-loss-statistics;
    }
    uses session-parameters;
    uses maintenance-statistics;
}

grouping stamp-session-percentile {
    description "Percentile grouping";
    leaf first-percentile {
        type percentile;
        default 95.00;
        description
            "First percentile to report";
    }
    leaf second-percentile {
        type percentile;
        default 99.00;
        description
            "Second percentile to report";
    }
    leaf third-percentile {
        type percentile;
        default 99.90;
        description
            "Third percentile to report";
    }
}
}
```

```
grouping delay-statistics {
  description "Delay statistics grouping";
  container delay {
    description "Packets transmitted delay";
    leaf min {
      type yang:gauge64;
      units nanoseconds;
      description
        "Min of Packets transmitted delay";
    }
    leaf max {
      type yang:gauge64;
      units nanoseconds;
      description
        "Max of Packets transmitted delay";
    }
    leaf avg {
      type yang:gauge64;
      units nanoseconds;
      description
        "Avg of Packets transmitted delay";
    }
  }
}
```

```
container delay-variation {
  description
    "Packets transmitted delay variation";
  leaf min {
    type yang:gauge32;
    units nanoseconds;
    description
      "Min of Packets transmitted
        delay variation";
  }
  leaf max {
    type yang:gauge32;
    units nanoseconds;
    description
      "Max of Packets transmitted
        delay variation";
  }
  leaf avg {
    type yang:gauge32;
    units nanoseconds;
    description
      "Avg of Packets transmitted
        delay variation";
  }
}
```

```
    }  
  }  
  
  grouping time-percentile-report {  
    description "Delay percentile report grouping";  
    container delay-percentile {  
      description  
        "Report round-trip, near- and far-end delay";  
      leaf rtt-delay {  
        type yang:gauge64;  
        units nanoseconds;  
        description  
          "Percentile of round-trip delay";  
      }  
      leaf near-end-delay {  
        type yang:gauge64;  
        units nanoseconds;  
        description  
          "Percentile of near-end delay";  
      }  
      leaf far-end-delay {  
        type yang:gauge64;  
        units nanoseconds;  
        description  
          "Percentile of far-end delay";  
      }  
    }  
  }  
  
  container delay-variation-percentile {  
    description  
      "Report round-trip, near- and far-end delay variation";  
    leaf rtt-delay-variation {  
      type yang:gauge32;  
      units nanoseconds;  
      description  
        "Percentile of round-trip delay-variation";  
    }  
    leaf near-end-delay-variation {  
      type yang:gauge32;  
      units nanoseconds;  
      description  
        "Percentile of near-end delay variation";  
    }  
    leaf far-end-delay-variation {  
      type yang:gauge32;  
      units nanoseconds;  
      description  
        "Percentile of far-end delay-variation";  
    }  
  }  
}
```

```
    }  
  }  
}  
  
grouping packet-loss-statistics {  
  description  
    "Grouping for Packet Loss statistics";  
  leaf loss-count {  
    type int32;  
    description  
      "Number of lost packets  
      during the test interval.";  
  }  
  leaf loss-ratio {  
    type percentage;  
    description  
      "Ratio of packets lost to packets  
      sent during the test interval.";  
  }  
  leaf loss-burst-max {  
    type int32;  
    description  
      "Maximum number of consecutively  
      lost packets during the test interval.";  
  }  
  leaf loss-burst-min {  
    type int32;  
    description  
      "Minimum number of consecutively  
      lost packets during the test interval.";  
  }  
  leaf loss-burst-count {  
    type int32;  
    description  
      "Number of occasions with packet  
      loss during the test interval.";  
  }  
}  
  
grouping session-parameters {  
  description  
    "Parameters Session-Sender";  
  leaf sender-ip {  
    type inet:ip-address;  
    mandatory true;  
    description "Sender IP address";  
  }  
  leaf session-sender-udp-port {
```

```
    type inet:port-number {
      range "49152..65535";
    }
    mandatory true;
    description "Sender UDP port number";
    reference
      "RFC 8762 Simple Two-Way Active
      Measurement Protocol Section 4.1.";
  }
  leaf stamp-session-id {
    type uint32;
    description
      "A STAMP test session identifier
      assigned by the Session-Sender.";
    reference
      "RFC 8972 Simple Two-Way Active
      Measurement Protocol Optional
      Extensions Section 3.";
  }
  leaf session-reflector-ip {
    type inet:ip-address;
    mandatory true;
    description "Reflector IP address";
  }
  leaf session-reflector-udp-port {
    type inet:port-number{
      range "862 | 1024..49151 | 49152..65535";
    }
    default 862;
    description
      "Reflector UDP port number";
    reference
      "RFC 8762 Simple Two-Way Active
      Measurement Protocol Section 4.1.";
  }
}

grouping session-security {
  description
    "Grouping for STAMP security and related parameters";
  container security {
    if-feature stamp-security;
    presence "Enables secure STAMP";
    description
      "Parameters for STAMP authentication";
    leaf key-chain {
      type kc:key-chain-ref;
      description "Name of key-chain";
    }
  }
}
```

```
    }
  }
  reference
    "RFC 8762 Simple Two-Way Active
    Measurement Protocol Section 4.4.";
}

/*
 * Configuration Data
 */
container stamp {
  description
    "Top level container for STAMP configuration";

  container stamp-session-sender {
    if-feature session-sender;
    description "STAMP Session-Sender container";

    leaf sender-enable {
      type boolean;
      default "true";
      description
        "Whether this network element is enabled to
        act as STAMP Session-Sender";
      reference
        "RFC 8762 Simple Two-Way Active
        Measurement Protocol Section 4.2.";
    }

    list sender-test-session {
      key "stamp-session-id";
      unique "stamp-session-id";
      description
        "This structure is a container of test session
        managed objects";

      leaf test-session-enable {
        type boolean;
        default "true";
        description
          "Whether this STAMP Test session is enabled";
      }

      leaf number-of-packets {
        type union {
          type uint32 {
            range 1..4294967294 {
              description
```



```
        "The overall number of UDP test packet
        to be transmitted by the sender for this
        test session";
    }
}
type enumeration {
    enum forever {
        description
            "Indicates that the test session SHALL
            be run *forever*.";
    }
}
default 10;
description
    "This value determines if the STAMP-Test session is
    bound by number of test packets or not.";
}

leaf interval {
    type uint32;
    units microseconds;
    description
        "Time interval between transmission of two
        consecutive packets in the test session in
        microseconds";
}

leaf session-timeout {
    when "../number-of-packets != 'forever'" {
        description
            "Test session timeout only valid if the
            test mode is periodic.";
    }
    type uint32;
    units "seconds";
    default 900;
    description
        "The timeout value for the Session-Sender to
        collect outstanding reflected packets.";
}

leaf measurement-interval {
    when "../number-of-packets = 'forever'" {
        description
            "Valid only when the test to run forever,
            i.e. continuously.";
    }
}
```

```
    type uint32;
    units "seconds";
    default 60;
    description
        "Interval to calculate performance metric when
        the test mode is 'continuous'.";
}

leaf repeat {
    type union {
        type uint32 {
            range 0..4294967294;
        }
        type enumeration {
            enum forever {
                description
                    "Indicates that the test session SHALL
                    be repeated *forever* using the
                    information in repeat-interval
                    parameter, and SHALL NOT decrement
                    the value.";
            }
        }
    }
    default 0;
    description
        "This value determines if the STAMP-Test session must
        be repeated. When a test session has completed, the
        repeat parameter is checked. The default value
        of 0 indicates that the session MUST NOT be repeated.
        If the repeat value is 1 through 4,294,967,294
        then the test session SHALL be repeated using the
        information in repeat-interval parameter.
        The implementation MUST decrement the value of repeat
        after determining a repeated session is expected.";
}

leaf repeat-interval {
    when "../repeat != '0'";
    type uint32;
    units seconds;
    default 0;
    description
        "This parameter determines the timing of repeated
        STAMP-Test sessions when repeat is more than 0.";
}

leaf dscp-value {
```

```
        type inet:dscp;
        default 0;
        description
            "DSCP value to be set in the test packet.";
    }

    leaf test-session-reflector-mode {
        type session-reflector-mode;
        default "stateless";
        description
            "The mode of STAMP-Reflector for the test session.";
    }

    uses session-parameters;
    leaf sender-timestamp-format {
        type timestamp-format;
        default ntp-format;
        description "Sender Timestamp format";
    }
    uses session-security;
    uses stamp-session-percentile;
}

container stamp-session-reflector {
    if-feature session-reflector;
    description
        "STAMP Session-Reflector container";
    leaf reflector-enable {
        type boolean;
        default "true";
        description
            "Whether this network element is enabled to
            act as STAMP Session-Reflector";
    }

    leaf ref-wait {
        type uint32 {
            range 1..604800;
        }
        units seconds;
        default 900;
        description
            "REFWAIT(STAMP test session timeout in seconds),
            the default value is 900";
    }

    leaf reflector-mode-state {
```

```
type session-reflector-mode;
    default stateless;
description
    "The state of the mode of the STAMP
    Session-Reflector";
}

list reflector-test-session {
    key "session-index";
    unique "sender-ip stamp-session-id";
    description
        "This structure is a container of test session
        managed objects";

    leaf session-index {
        type uint32;
        description "Session index";
    }

    leaf stamp-session-id {
        type union {
            type uint32;
            type enumeration {
                enum any {
                    description
                        "Indicates that the Session-Reflector
                        accepts STAMP test packets from
                        a Session-Sender with any SSID
                        value";
                }
            }
        }
        description
            "This value determines whether specific
            SSID of the Session-Sender
            or the wildcard, i.e. any SSID accepted";
        reference
            "RFC 8972 Simple Two-Way Active
            Measurement Protocol Optional
            Extensions Section 3.";
    }

    leaf dscp-handling-mode {
        type session-dscp-mode;
        default copy-received-value;
        description
            "Session-Reflector handling of DSCP:
            - use value copied from received STAMP-Test packet;
    }
}
```

```
        - use value explicitly configured";
    }

    leaf dscp-value {
        when "../dscp-handling-mode = 'use-configured-value'";
        type inet:dscp;
        default 0;
        description
            "DSCP value to be set in the reflected packet
            if dscp-handling-mode is set to use-configured-value.";
    }

    leaf sender-ip {
        type union {
            type inet:ip-address;
            type enumeration {
                enum any {
                    description
                        "Indicates that the Session-Reflector
                        accepts STAMP test packets from
                        any Session-Sender";
                }
            }
        }
        default any;
        description
            "This value determines whether specific
            IPv4/IPv6 address of the Session-Sender
            or the wildcard, i.e. any address";
    }

    leaf sender-udp-port {
        type union {
            type inet:port-number {
                range "49152..65535";
            }
            type enumeration {
                enum any {
                    description
                        "Indicates that the Session-Reflector
                        accepts STAMP test packets from
                        any Session-Sender";
                }
            }
        }
        default any;
        description
            "This value determines whether specific
```

```
        port number of the Session-Sender
        or the wildcard, i.e. any";
    }

    leaf reflector-ip {
        type union {
            type inet:ip-address;
            type enumeration {
                enum any {
                    description
                        "Indicates that the Session-Reflector
                        accepts STAMP test packets on
                        any of its interfaces";
                }
            }
        }
        default any;
        description
            "This value determines whether specific
            IPv4/IPv6 address of the Session-Reflector
            or the wildcard, i.e. any address";
    }

    leaf reflector-udp-port {
        type inet:port-number{
            range "862 | 1024..49151 | 49152..65535";
        }
        default 862;
        description
            "Reflector UDP port number";
        reference
            "RFC 8762 Simple Two-Way Active
            Measurement Protocol Section 4.1.";
    }

    leaf reflector-timestamp-format {
        type timestamp-format;
        default ntp-format;
        description "Reflector Timestamp format";
    }
    uses session-security;
}

}

/*
 * Operational state data nodes
 */
```

```
container stamp-state {
  config false;
  description
    "Top level container for STAMP state data";

  container stamp-session-sender-state {
    if-feature session-sender;
    description
      "Session-Sender container for state data";
    list test-session-state{
      key "session-index";
      description
        "This structure is a container of test session
        managed objects";

      leaf session-index {
        type uint32;
        description "Session index";
      }

      leaf sender-session-state {
        type enumeration {
          enum active {
            description "Test session is active";
          }
          enum ready {
            description "Test session is idle";
          }
        }
        description
          "State of the particular STAMP test
          session at the sender";
      }
    }

    container current-stats {
      description
        "This container contains the results for the current
        Measurement Interval in a Measurement session ";
      leaf start-time {
        type yang:date-and-time;
        mandatory true;
        description
          "The time that the current Measurement Interval started";
      }

      uses test-session-statistics;
    }
  }
}
```

```
list history-stats {
  key session-index;
  description
    "This container contains the results for the history
    Measurement Interval in a Measurement session ";
  leaf session-index {
    type uint32;
    description
      "The identifier for the Measurement Interval
      within this session";
  }

  leaf end-time {
    type yang:date-and-time;
    mandatory true;
    description
      "The time that the Measurement Interval ended";
  }

  uses test-session-statistics;
}

}

container stamp-session-refl-state {
  if-feature session-reflector;
  description
    "STAMP Session-Reflector container for
    state data";
  leaf reflector-light-admin-status {
    type boolean;
    description
      "Whether this network element is enabled to
      act as STAMP Session-Reflector";
  }
}

list test-session-state {
  key "session-index";
  description
    "This structure is a container of test session
    managed objects";

  leaf session-index {
    type uint32;
    description "Session index";
  }

  leaf reflector-timestamp-format {
```



```
        type timestamp-format;
        description "Reflector Timestamp format";
    }
    uses session-parameters;
    uses maintenance-statistics;
}
}
}

rpc stamp-sender-start {
    description
        "start the configured sender session";
    input {
        leaf stamp-session-id {
            type uint32;
            mandatory true;
            description
                "The STAMP session to be started";
        }
    }
}

rpc stamp-sender-stop {
    description
        "stop the configured sender session";
    input {
        leaf stamp-session-id {
            type uint32;
            mandatory true;
            description
                "The session to be stopped";
        }
    }
}
}
}
<CODE ENDS>
```

#### 4. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-stamp

Registrant Contact: The IPPM WG of the IETF.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC7950].

name: ietf-stamp

namespace: urn:ietf:params:xml:ns:yang:ietf-stamp

prefix: stamp

reference: RFC XXXX

## 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 [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured 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 an adverse effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can adversely affect the routing subsystem of both the local device and the network. This may lead to corruption of the measurement that may result in false corrective action, e.g., false negative or false positive. That could be, for example, prolonged and undetected deterioration of the quality of service or actions to improve the quality unwarranted by the real network conditions.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can disclose the operational state information of VRRP on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

TBD

## 6. Acknowledgments

Authors recognize and appreciate valuable comments provided by Adrian Pan and Henrik Nydell.

## 7. References

### 7.1. 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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [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>>.
- [RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-Way Active Measurement Protocol Optional Extensions", RFC 8972, DOI 10.17487/RFC8972, January 2021, <<https://www.rfc-editor.org/info/rfc8972>>.

## 7.2. Informative References

- [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>>.

## Appendix A. Example of STAMP Session Configuration

Figure 5 shows a configuration example of a STAMP-Sender.

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <stamp xmlns="urn:ietf:params:xml:ns:yang:ietf-stamp">
    <stamp-session-sender>
      <session-enable>enable</session-enable>
      <stamp-session-id>10</stamp-session-id>
      <test-session-enable>enable<test-session-enable>
      <number-of-packets>forever</number-of-packets>
      <interval>10</interval> <!-- 10 microseconds -->
      <measurement-interval/> <!-- use default 60 seconds -->
      <!-- use default 0 repetitions,
            i.e. do not repeat this session -->
      <repeat/>
      <dscp-value/> <!-- use default 0 (CS0) -->
      <!-- use default 'stateless' -->
      <test-session-reflector-mode/>
      <sender-ip></sender-ip>
      <session-sender-udp-port></session-sender-udp-port>
      <session-reflector-ip></session-reflector-ip>
      <session-reflector-udp-port/> <!-- use default 862 -->
      <sender-timestamp-format/>
      <!-- No authentication -->
      <first-percentile/> <!-- use default 95 -->
      <second-percentile/> <!-- use default 99 -->
      <third-percentile/> <!-- use default 99.9 -->
    </stamp-session-sender>
  </stamp>
</data>
```

Figure 5: XML instance of STAMP Session-Sender configuration

```
<?xml version="1.0" encoding="utf-8"?>
<data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <stamp xmlns="urn:ietf:params:xml:ns:yang:ietf-stamp">
    <stamp-session-reflector>
      <session-enable>enable</session-enable>
      <ref-wait/> <!-- use default 900 seconds -->
      <!-- use default 'stateless' -->
      <reflector-mode-state/>
      <stamp-session-id/> <!-- use default 'any' -->
      <!-- use default 'copy-received-value' -->
      <dscp-handling-mode/>
      <!-- not used because of dscp-hanling-mode
            being 'copy-received-value' -->
      <dscp-value/>
      <sender-ip/> <!-- use default 'any' -->
      <sender-udp-port/> <!-- use default 'any' -->
      <reflector-ip/> <!-- use default 'any' -->
      <reflector-udp-port/> <!-- use default 862 -->
      <reflector-timestamp-format/>
      <!-- No authentication -->
    </stamp-session-reflector>
  </stamp>
</data>
```

Figure 6: XML instance of STAMP Session-Reflector configuration

#### Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.com

Xiao Min  
ZTE Corp.

Email: xiao.min2@zte.com.cn

Wei S Luo  
Ericsson

Email: wei.s.luo@ericsson.com

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: July 20, 2022

J. Kumar  
S. Anubolu  
J. Lemon  
R. Manur  
Broadcom Inc.  
H. Holbrook  
Arista Networks  
A. Ghanwani  
Dell EMC  
D. Cai  
H. Ou  
AliBaba Inc.  
Y. Li  
Huawei  
X. Wang  
Fujian Ruijie Networks co., ltd.  
January 20, 2022

Inband Flow Analyzer  
draft-kumar-ippm-ifa-04

Abstract

Inband Flow Analyzer (IFA) records flow specific information from an end station and/or switches across a network. This document discusses the method to collect data on a per hop basis across a network and perform localized or end to end analytics operations on the data. This document also describes a transport-agnostic header definition that may be used for tunneled and non-tunneled flows alike.

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 20, 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Terminology . . . . .	3
1.2. Scope . . . . .	3
1.3. Applicability . . . . .	4
1.4. Motivation . . . . .	4
2. Requirements . . . . .	4
2.1. Encapsulation Requirements . . . . .	4
2.2. Operational Requirements . . . . .	5
2.3. Cost and Performance Requirements . . . . .	6
3. IFA Operations . . . . .	6
3.1. IFA Zones . . . . .	8
3.2. IFA Function Nodes . . . . .	8
3.2.1. Initiating Function Node . . . . .	9
3.2.2. Transit Function Node . . . . .	9
3.2.3. Terminating Function Node . . . . .	9
3.2.4. Metadata Fragmentation Function . . . . .	9
3.3. IFA Cloning, Truncation, and Drop . . . . .	10
3.4. IFA Header . . . . .	10
3.4.1. IFA Metadata Header . . . . .	13
3.4.2. IFA Checksum Header . . . . .	13
3.4.3. IFA Metadata Fragmentation (MF) Header . . . . .	14
3.5. IFA Metadata . . . . .	15
3.5.1. Global Name Space (GNS) Identifier . . . . .	15
3.5.2. Local Name Space (LNS) Identifier . . . . .	16
3.5.3. Device ID . . . . .	16
3.6. IFA Network Overhead . . . . .	16
3.7. IFA Analytics . . . . .	17
3.8. IFA Packet Format . . . . .	17
3.8.1. IFA Packet Format with TS Flag Set . . . . .	18
3.8.2. VxLAN Packet . . . . .	20
3.8.3. GRE Packet . . . . .	22



3.8.4. Geneve Packet . . . . .	23
3.8.5. IPinIP Packet . . . . .	25
3.8.6. IPv6 Extension Headers with IFA . . . . .	26
3.8.7. IP AH/ESP/WESP Packet . . . . .	28
3.9. IFA Load Balancing . . . . .	30
4. Interoperability Considerations . . . . .	30
5. Security Considerations . . . . .	31
6. References . . . . .	31
6.1. Normative References . . . . .	31
6.2. Informative References . . . . .	31
Appendix A. . . . .	32
A.1. Probe Marker . . . . .	32
A.2. DSCP . . . . .	32
A.3. IP Options . . . . .	32
A.4. IPv4 Identification or Reserved Flag . . . . .	33
Authors' Addresses . . . . .	33

## 1. Introduction

This document describes Inband Flow Analyzer (IFA) which is a mechanism to mark packets in a flow to enable the collection of metadata regarding the analyzed flow. IFA defines an IFA header to mark the flow and direct the collection of analyzed metadata per marked packet per hop across a network. The ability to mark a packet using an IFA OAM header can also be leveraged to create synthetic flows meant for network data collection. This document describes a mechanism that may be used to monitor live traffic and/or create synthetic flows. This document also describes IFA zones, IFA reports, and IFA metadata. IFA does not require changes to protocol headers in order to collect metadata or analyze flows. IFA puts minimal requirements on switching silicon.

### 1.1. Terminology

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].

IFA: Inband Flow Analyzer

MTU: Maximum Transmit Unit

### 1.2. Scope

This document describes IFA deployment, the type of traffic that is supported, header definitions, analytics, and data path functions.

IFA deployment involves defining an IFA zone and understanding the requirements in terms of traffic overhead and points of data collection. Given that IFA provides the ability to perform local analytics on the collected data, this document describes the scope of the analytics function as well. The scope of IFA is from an end station and/or ToR, through any/all nodes in the network, and terminating in a network switch and/or an end station.

IFA can create a synthetic stream of traffic and use it to collect metadata along the path. This sampled stream is later discarded. IFA can also insert metadata on a per packet basis in live traffic. Inband insertion of metadata can be done within the payload or via tail stamping.

This draft defines an identification mechanism using a dedicated protocol type in the IP header for identifying IFA.

### 1.3. Applicability

IFA is capable of providing traffic analysis in an encapsulation-agnostic manner. Simple TCP and UDP flows, as well as tunneled flows, can be monitored. IFA can be enabled on an end station, or it can be enabled just on network switches. Enabling IFA on an end station provides better scalability and visibility by monitoring intra end station or inter end station traffic. IFA performs best when there is hardware assistance for deriving the flow metadata in the data path. This document describes data path functions for IFA.

### 1.4. Motivation

The main motivation for IFA is to collect analyzed metadata from packets within a flow for a given application. The definition of the IFA header ensures that it works for any IP packet, and with minimal impact on hardware performance.

## 2. Requirements

IFA requirements are defined with operational efficiency, performance of the network, and cost of hardware in mind.

### 2.1. Encapsulation Requirements

IFA packets MUST be clearly marked and identifiable so that a networking element in the flow path can insert metadata or perform other IFA operations.

IFA packets need to be easily identified for performance reasons. IFA packet identification MUST be the same for all the IP packet

types. This means that expensive hardware modifications are not needed for supporting new protocol types.

Since IFA packet processing is a data path function, the IFA header MUST keep the processing overhead minimal. Simple parsing in the switch hardware with localized read/write fields in IFA header will optimize the switch performance and cost.

A single IFA encapsulation MUST support IPv4 and IPv6 protocol types for tunneled and non-tunneled packets, preserving the fields used for load balancing hash computation.

IFA MAY support a checksum for the entire IFA metadata stack instead of a checksum per metadata element.

## 2.2. Operational Requirements

IFA MUST preserve the flow path across the network.

IFA MUST incur minimal traffic overhead.

IFA MUST provide an option to clone and truncate a packet to avoid disrupting the PMTU discovery of a network.

Cloning SHOULD be supported. Sampling of cloned traffic MUST be at a sampled ratio to keep the network overhead to a minimum.

IFA MUST provide the ability to insert metadata on cloned traffic.

IFA MUST provide the ability to insert metadata on live traffic.

IFA MAY provide the ability to specify checksum validation on the IFA header and metadata.

IFA MUST provide the ability to define a zone using hop count.

IFA MUST provide the ability for a networking element to perform metadata insertion in the payload.

IFA MAY provide the ability for networking element to insert metadata as tail stamping.

IFA MUST be able to support an IFA zone name space, also referred to as a global name space.

IFA MUST be able to support a per hop name space, also referred to as a local name space.

IFA MAY be able to support fragmentation of metadata. Fragmentation is needed to support a large number of hops in the network path.

### 2.3. Cost and Performance Requirements

The IFA header and metadata MUST be treated as foreign data present in the application data. IFA SHOULD be able to insert or strip the IFA header and metadata without modifying the layer 4 headers. This will help keep the cost of hardware down with no degradation in performance.

IFA MUST support the ability to clone and/or truncate, live traffic for IFA metadata insertion. This is needed for PMTU protocols to work within the IFA zone.

The IFA header MUST provide the ability to differentiate between a cloned packet and an original packet. This is needed for hardware to be able to identify and filter the cloned traffic at the edge of an IFA zone.

IFA encapsulation MUST provide mechanism to avoid impacting the parse depth of hardware for packet processing.

IFA MUST NOT require pre-allocation for reserving the space in a packet. The overhead of managing reserved space in a packet can result in performance degradation.

### 3. IFA Operations

IFA performs flow analysis, and possible actions on the flow data, inband. Once a flow is enabled for analysis, a node with the role of "Initiator" makes a copy of the flow or samples the live traffic flow, or tags a live traffic flow for analysis and data collection. Copying of a flow is done by sampling or cloning the flow. These new packets are representative packets of the original flow and possess the exact same characteristics as the original flow. This means that IFA packets traverse the same path in the network and same queues in the networking element as the original packet would. Figure 1 shows the IFA based Telemetry Framework. The terminating node is responsible for terminating the IFA flow by summarizing the metadata of the entire path and sending it to a Collector.

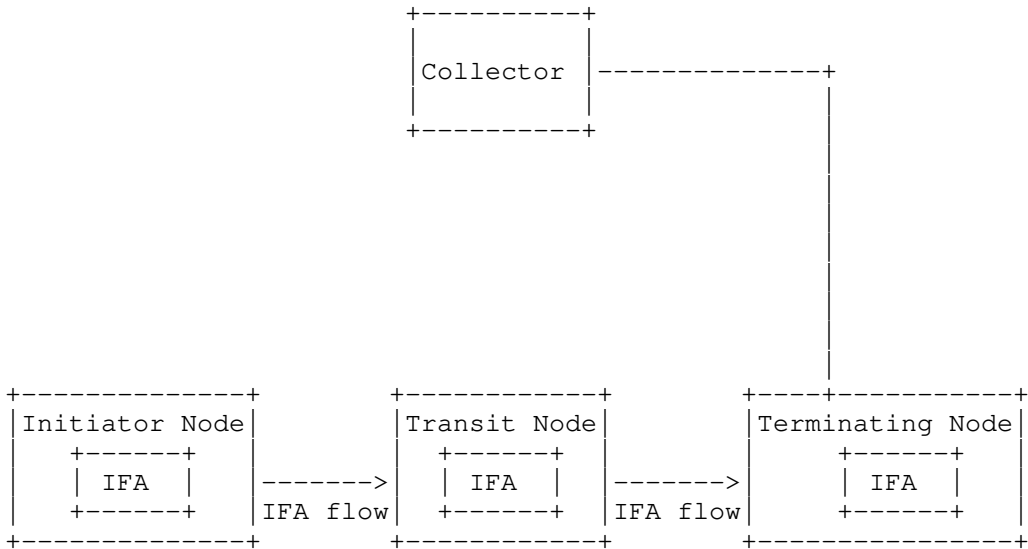


Figure 1: IFA Zone Framework without fragmentation

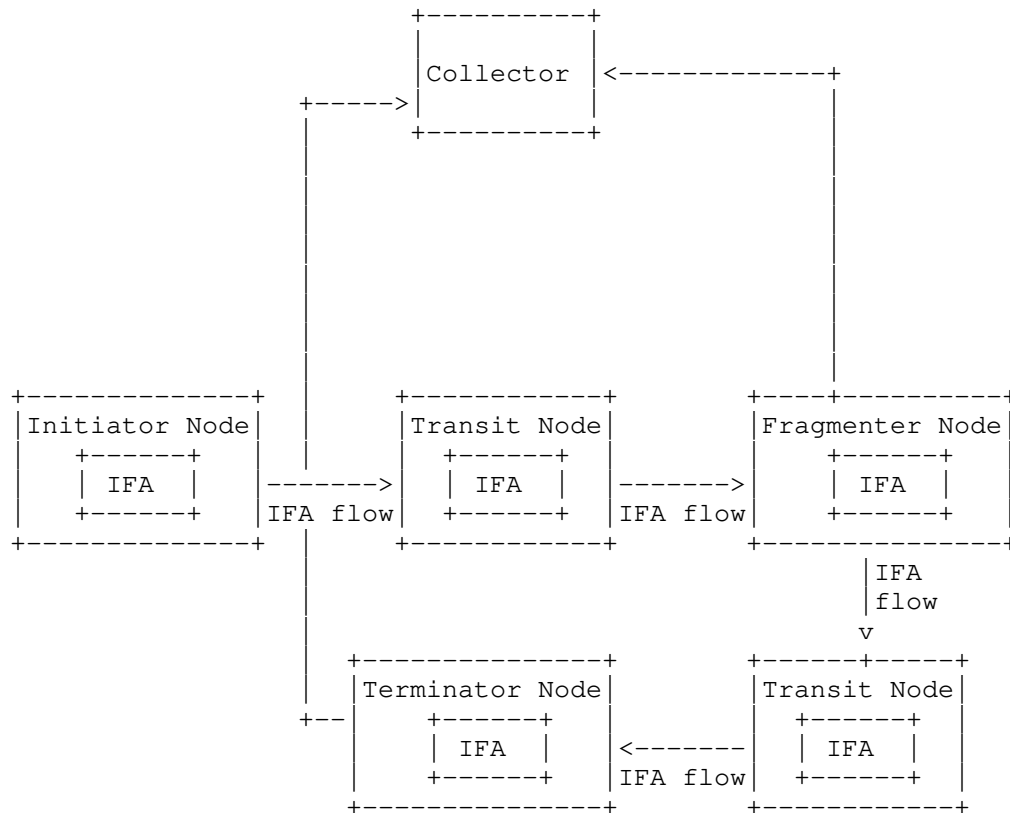


Figure 2 IFA Zone Framework with metadata fragmentation

### 3.1. IFA Zones

An IFA zone is the domain of interest where IFA monitoring is enabled. An IFA zone MUST have designated IFA function nodes. An IFA zone can be controlled by setting an appropriate TTL value in the L3 header. Initiating and Terminating function nodes are always at the edge of the IFA zone. Internal nodes in the IFA zone are always Transit function nodes.

### 3.2. IFA Function Nodes

There are three types of IFA functional nodes with respect to a specific or set of flows. Each node MAY perform metadata fragmentation function as well.

### 3.2.1. Initiating Function Node

An end station, a switch, or any other middleware can perform the IFA initiating function. It is advantageous to keep this role closest to the application to maximize flow visibility. An IFA initiating function node performs the following functions for a flow:

- Samples the flow traffic of interest based on a configuration.
- Converts the traffic into an IFA flow by adding an IFA header to each sample.
- Updates the packet with initiating function node metadata.
- MAY mandate a specific template ID metadata by all networking elements.
- MAY mandate tail stamping of metadata by all networking elements.

### 3.2.2. Transit Function Node

An IFA transit node is responsible for inserting transit node metadata in the IFA packets in the specified flow.

### 3.2.3. Terminating Function Node

An IFA terminating node is responsible for the following for a flow:

- Inserts terminating node metadata in an IFA packet.
- Performs a local analytics function on one or more segments of metadata, e.g., threshold breach for residence time, congestion notifications, and so on.
- Filters an IFA flow in case of cloned traffic.
- Sends a copy or report of the packet to collector.
- Removes the IFA headers and forwards the packet in case of live traffic.

### 3.2.4. Metadata Fragmentation Function

There are cases where the size of metadata may grow too big for link MTU or path MTU, or where it imposes excessive overhead for the terminating function node to remove it. This is specially true in networks with a large number of hops between initiator function node and terminating function node. This is also true where the size of per hop metadata itself is large. For such cases, IFA defines a metadata fragmentation function. Metadata fragmentation function allows, removal of metadata from the packet and send a copy/report of the packet to collector. Correlation of metadata fragments and recreation of metadata stack for the entire flow path is done by the collector.

There is no dedicated node performing the metadata fragmentation function. As an IFA packet traverses the hops in an IFA zone, any

node MAY detect the need to fragment the packet's metadata stack and perform metadata fragmentation.

Metadata fragmentation is done if the IFA header in the packet has "MDF" bit set and the current length of the metadata would exceed the maximum length after the addition of metadata by the current node. A node MAY create a copy of the packet or create an IFA report, remove the existing metadata stack from the packet, insert its own metadata, and finally forward the packet. A node MAY also update the IFA MDF (Meta Data Fragment) header fragment identifier, current length, IP length, and IP header checksum.

The maximum length in an IFA header, if set to "0", MAY trigger the metadata fragmentation special function. This mechanism can be used to generate IFA reports at each hop and never insert metadata in the packet. If maximum length is set to "0", a node MAY ONLY create an IFA report or copy of the packet including its own metadata. A node MUST NOT update the IFA MD header current length, IP length, or insert metadata in the IFA packet. The node MUST increment the IFA MDF header fragment identifier field.

### 3.3. IFA Cloning, Truncation, and Drop

IFA allows cloning of live traffic. It is expected that cloned traffic will have the same network path characterization as the original traffic i.e. follow the same network path, use the same queues etc.

Cloned traffic can be truncated to accommodate the PMTU of the IFA zone.

Cloned traffic MUST be dropped by the terminating function node of the IFA zone.

### 3.4. IFA Header

The IFA header is described below. An experimental IP protocol number is used in the IP header to identify an IFA packet. The IP header protocol type field is copied into the IFA header NextHdr field for hardware to correctly interpret the layer 4 header.



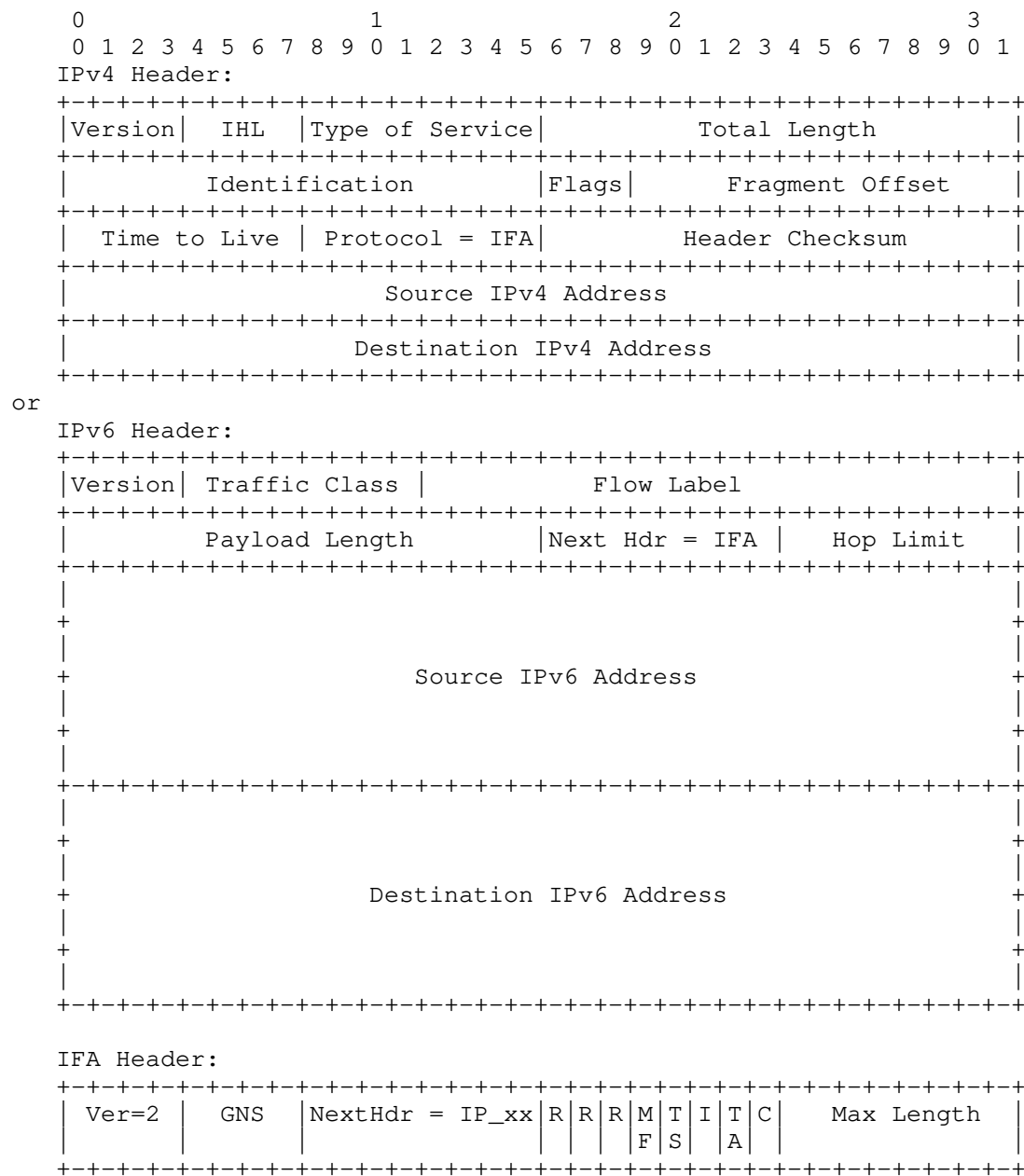


Figure 2: IFA 2 Header Format

- (1) Version (4 bits) - Specifies the version of IFA header.

(2) GNS (4 bits) - Global Name Space. Specifies the IFA zone scoped name space for IFA metadata.

(3) Protocol Type (8 bits) - IP Header protocol type. This is copied from the IP header.

(4) Flags (8 bits)

0: R - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

1: R - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

2: R - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

3: MF - Metadata Fragment. Indicates the presence of the optional metadata fragment header. This header is inserted and initialized by the initiator node. If the MF bit is set, nodes in the path MAY perform fragmentation of metadata stack if the current length exceeds the maximum length.

4: TS - Tail Stamp. Indicates the IFA zone is requiring tail stamping of metadata.

5: I - Inband. Indicates this is live traffic. Strip and forward MUST be performed by the terminator node if this bit is set.

6: TA - Turn Around. Indicates that the IFA packet needs to be turned around at the terminating node of the IFA zone and sent back to source IP address. This bit MAY be used for probe packets where probes are collection bidirectional information in the network. This is same as echo request and echo reply. A packet MAY be generated with TA bit set and collects metadata in one direction and after it is turned around by the terminating function node, collects metadata in the reverse direction.

7: C - Checksum - Indicates the presence of the optional checksum header. The checksum MUST be computed and updated for the IFA header and metadata at each node that modified the header and/or metadata. A node MAY perform checksum validation before updating the checksum.

(5) Max Length (8 bits) - Specifies the maximum allowed length of the metadata stack in multiples of 4 octets. This field is initialized by the initiator node. Each node in the path MUST compare the current length with the max length, and if the current length equals

or exceeds the max length, the transit nodes MUST stop inserting metadata.

### 3.4.1. IFA Metadata Header

The IFA metadata header is always present.

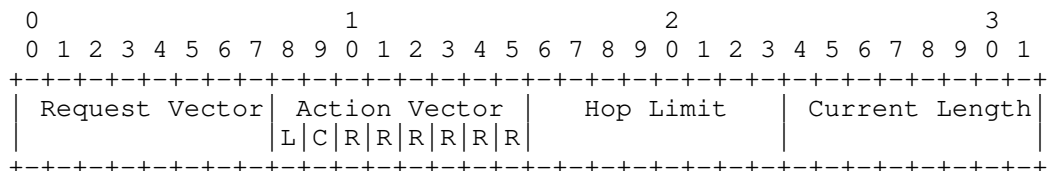


Figure 3: IFA Metadata Header Format

Request Vector (8 bits) - This vector specifies the presence of fields as specified by GNS. Fields are always 4-octet aligned. This field can be made extensible by defining a new GNS for an IFA zone.

Action Vector (8 bits) - This vector specifies node-local or end-to-end action on the IFA packets.

- ```

0: L - Loss.   Loss bit to measure packet loss.

1: C - Color.  Color bit to mark the packet.

2: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

3: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

4: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

5: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

6: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

7: R - Reserved.  MUST be initialized to 0 on transmission and
ignored on receipt.

```

Hop Limit (8 bits) - Specifies the maximum allowed hops in an IFA zone. This field is initialized by the initiator node. The hop limit MUST be decremented at each hop. If the incoming hop limit is 0, current nodes MUST NOT insert metadata. A value of 0xFF means that the Hop limit check MUST be ignored.

Current Length (8 bits) - Specifies the current length of the metadata in multiples of 4 octets.

### 3.4.2. IFA Checksum Header

The IFA checksum header is optional. Presence of the checksum header is indicated by the C bit in the flags field of the IFA header.

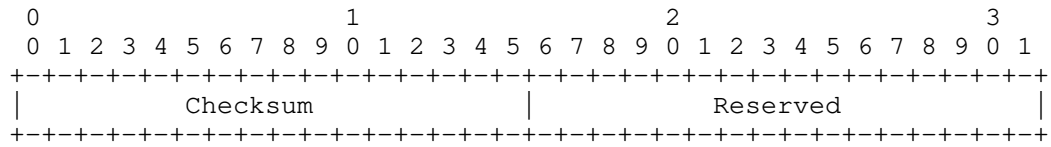


Figure 4: IFA Checksum Header Format

Checksum (16 bits) - The checksum covers the IFA header and metadata stack. Initiator function node MAY compute the full checksum including IFA header and metadata. Other nodes MAY compute delta checksum for the inserted/deleted metadata.

Reserved (16 bits) - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

### 3.4.3. IFA Metadata Fragmentation (MF) Header

The IFA metadata fragmentation (MF) header is optional. Presence of the fragmentation header is indicated by the MF bit in the flags field of the IFA header.

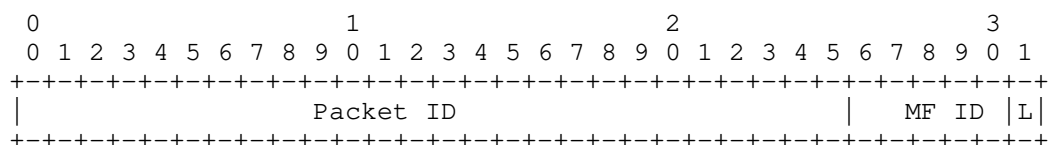


Figure 5: IFA MF Header Format

Packet ID (26 bits) - Packet identification value generated by the initiator node. This value is node scoped.

Metadata Fragment ID (5 bits) - The initiator MUST initialize this value to 0. A node performing metadata fragmentation function MUST increment the value by 1.

L (1 bit) - This bit is set by the node creating the last metadata fragment. This will ALWAYS be the terminating function node. If incoming hop limit is "0", terminating function node will still generate copy/report of the packet and MUST set L bit. Collector MUST implement mechanism to recover from lost packets/reports with L bit set.

The MF header is a fixed overhead of 4 octets per packet. A network operator MUST identify the need for using IFA metadata fragmentation. The following network conditions can be considered:

- If an IFA packet may exceed the link or path MTU of the flow path
- If there are large number of hops in a flow path and MAY trigger link or path MTU breach
- If the length of metadata creates excessive overhead for terminating function node to delete the metadata.
- If each hop needs to generate its own IFA report (postcard mechanism)

With 26 bits of packet id, a maximum datagram lifetime (MDL) of 3 seconds, and an average Internet mix (IMIX) packet size of 512 bytes,

we get 183.25 Gbps of IFA traffic bit rate per node before the packet identifier wraps around. The collector can use [device id, packet id, MF id, L] to rebuild the fragmented packet.

5 bits of MF id will support 32 metadata fragments.

### 3.5. IFA Metadata

The IFA metadata is the information inserted by each hop after the IFA header. The IFA metadata can be inserted at the following offsets:

- Payload Stamping: Immediately after the layer 4 header. This is the default setting.
- Tail Stamping: After the end of the packet. This is controlled by the TS bit in the flags field of the IFA header.

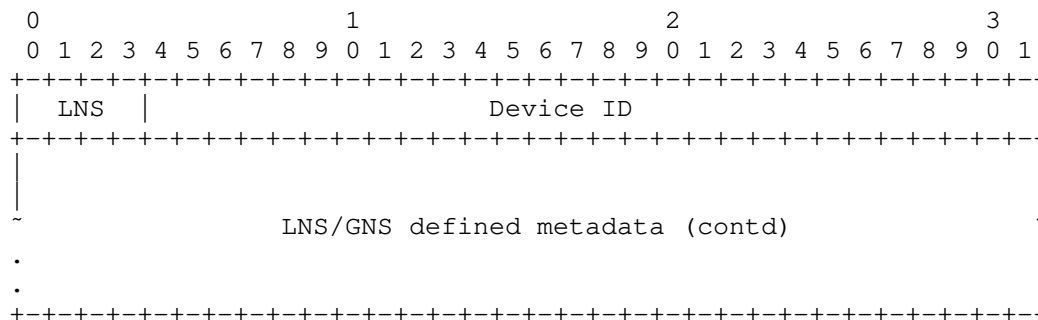


Figure 6: IFA Metadata Format

The IFA metadata header contains a set fields as defined by the name space identifier. Two types of name space identifiers are proposed.

### 3.5.1. Global Name Space (GNS) Identifier

A Global Name Space (GNS) is specified in the IFA header by the initiator node. The scope of a GNS is an IFA zone. All networking elements in an IFA zone MUST insert metadata as per the GNS ID specified in the IFA header. This is defined as the "Uniform Mode" of deployment.

A GNS value of 0xF indicates that metadata in an IFA zone is defined by the LNS of each hop.

The advantage of using the uniform mode is having a simple and uniform metadata stack. This means less load on a collector for parsing.

The disadvantage is that metadata fields are supported based on the least capable networking element in the IFA zone.

### 3.5.2. Local Name Space (LNS) Identifier

A Local Name Space (LNS) is specified in the metadata header. A GNS value of 0xF in the IFA header indicates the presence of an LNS. This is defined as the "Non-uniform Mode" of deployment.

A switch pipeline MUST parse the GNS field in the IFA header. The parsing result will dictate the name space ID that the hop needs to comply with.

The advantage of using the non-uniform mode is having a flexible metadata stack. This allows each hop to include the most relevant data for that hop.

The disadvantage is more complex parsing by a collector.

### 3.5.3. Device ID

A 28-bit unique identifier for the device inserting the metadata. If a GNS other than 0xF is present, then the device ID can be expanded to a 32 bit value. This is to support including an IPv4 loopback address as a Device ID.

### 3.6. IFA Network Overhead

A common problem associated with inserting metadata on a per packet per flow basis is the amount of traffic overhead on the network. IFA 2 is defined to minimize the overhead on the network.

|                          |            |
|--------------------------|------------|
| IFA Base Header          | : 4 octets |
| IFA Metadata Header      | : 4 octets |
| IFA Checksum Header      | : 4 octets |
| IFA Fragmentation Header | : 4 octets |

Minimum Overhead:

IFA header : 4 octets

IFA Metadata Header : 4 octets

Total Min Overhead : 8 octets per packet

### 3.7. IFA Analytics

There are two kinds of actions considered in this proposal.

- (1) Action Bit MAP in IFA Header - This is encoded in the IFA header. Each node in the path MAY use the action bitmap to insert or not insert the metadata based on exceeding a locally-specified threshold. Not inserting the metadata is indicated by setting the field value to -1 (all 1s).
- (2) Terminating Node Actions - A terminating node may decide to perform threshold or other actions on the set of metadata in the packet. This information is not encoded in the IFA header.

### 3.8. IFA Packet Format

The IFA header is treated as a layer 3 extension header. IFA header and metadata stack length is reflected in IP total length field. IPv6 extension headers are ordered. The IFA header MUST be the last extension header in the IPv6 extension header chain. Similarly in case of IPv4 AH/ESP/WESP extension headers, IFA header MUST be the last extension header.

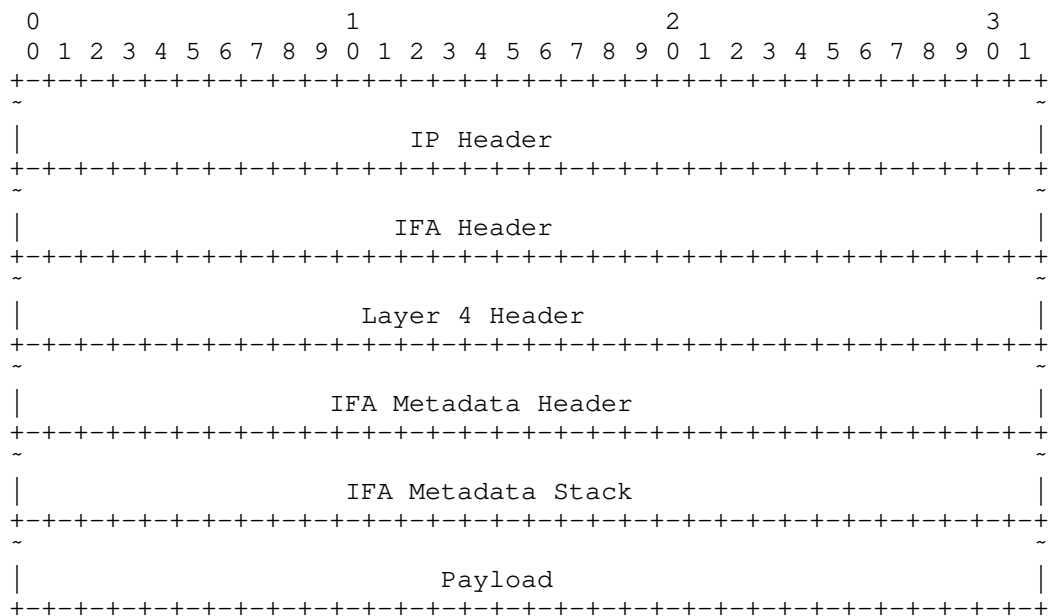


Figure 8 IFA Packet Format



### 3.8.1. IFA Packet Format with TS Flag Set

In case the Tail Stamp flag is set in the IFA header, the IFA metadata header and metadata stack are inserted at the end of the packet just before the FCS. Each node inserts metadata at the bottom of IFA metadata stack.

One of the key advantages of using TS is to support legacy devices and/or appliances that need to look at the layer 5 data. The IP length and IP header checksum are updated at each hop inserting metadata. This is the same as without the TS flag.

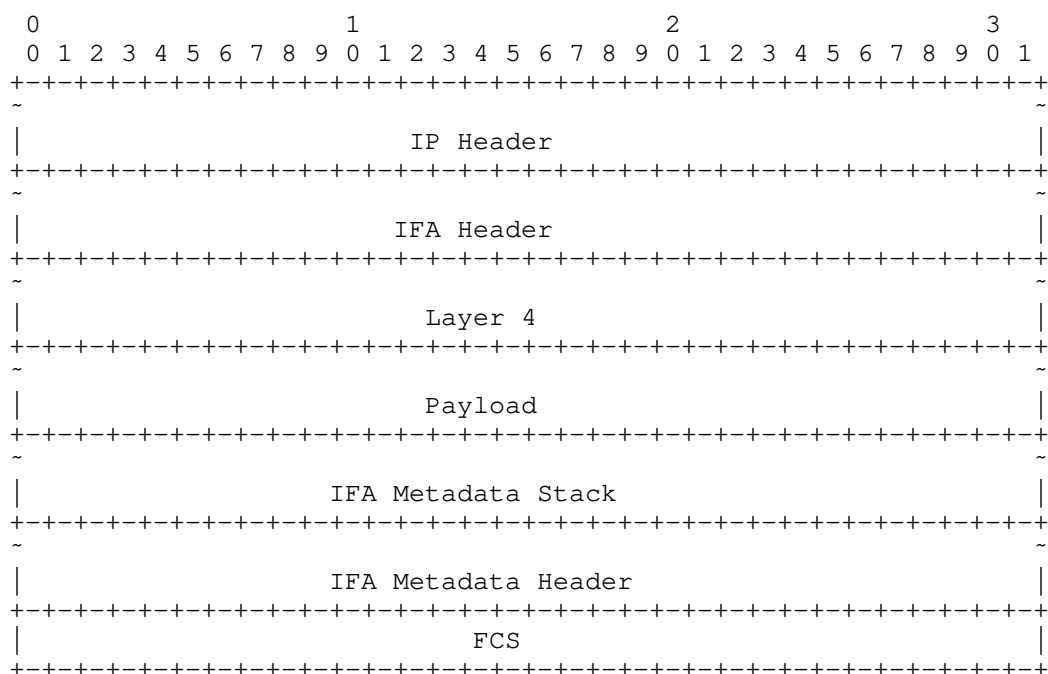
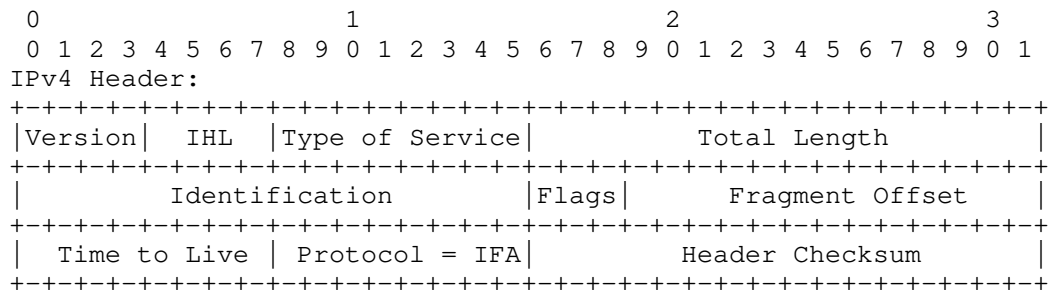
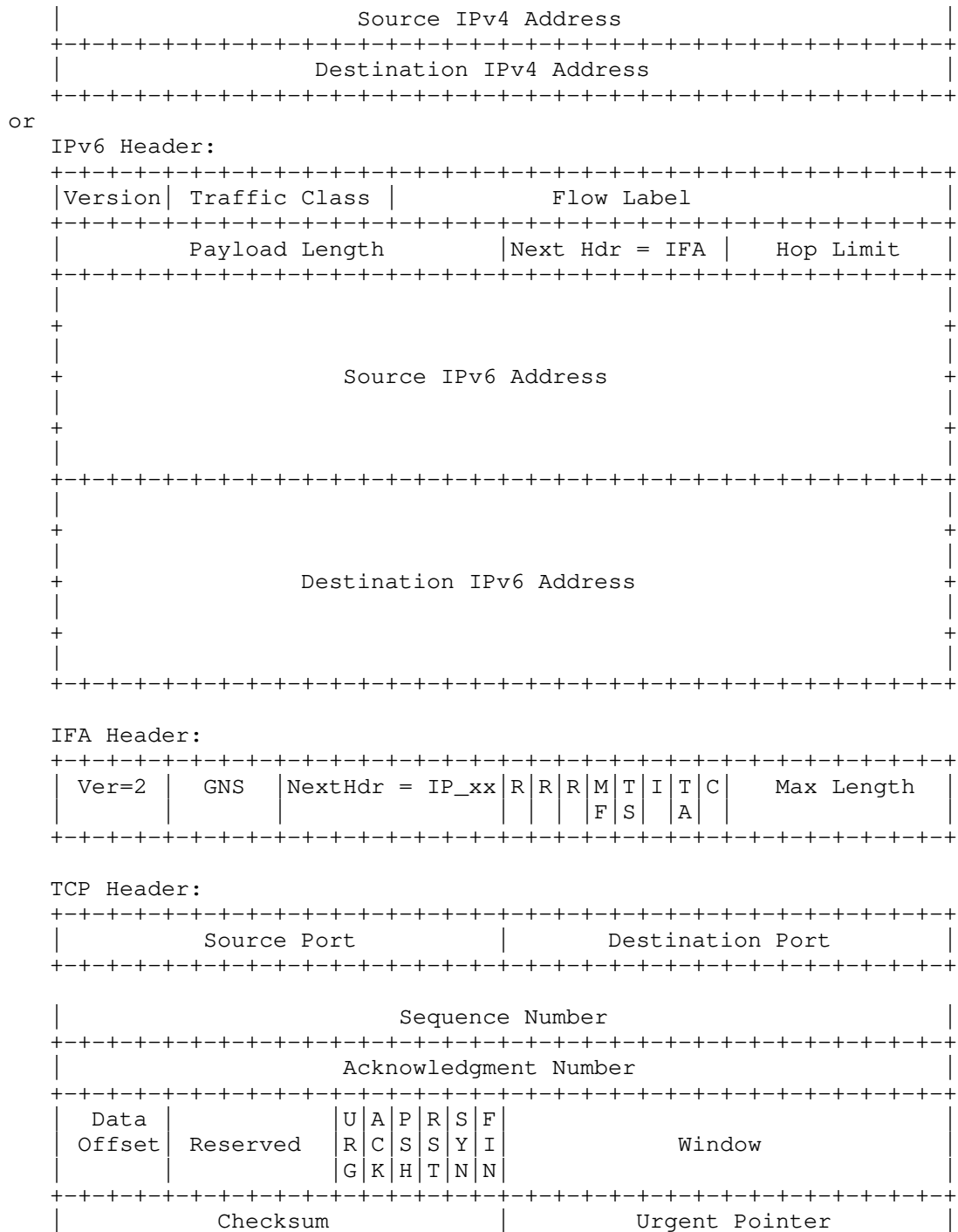


Figure 7: IFA Packet Format with TS 3.8.1 TCP/UDP Packet





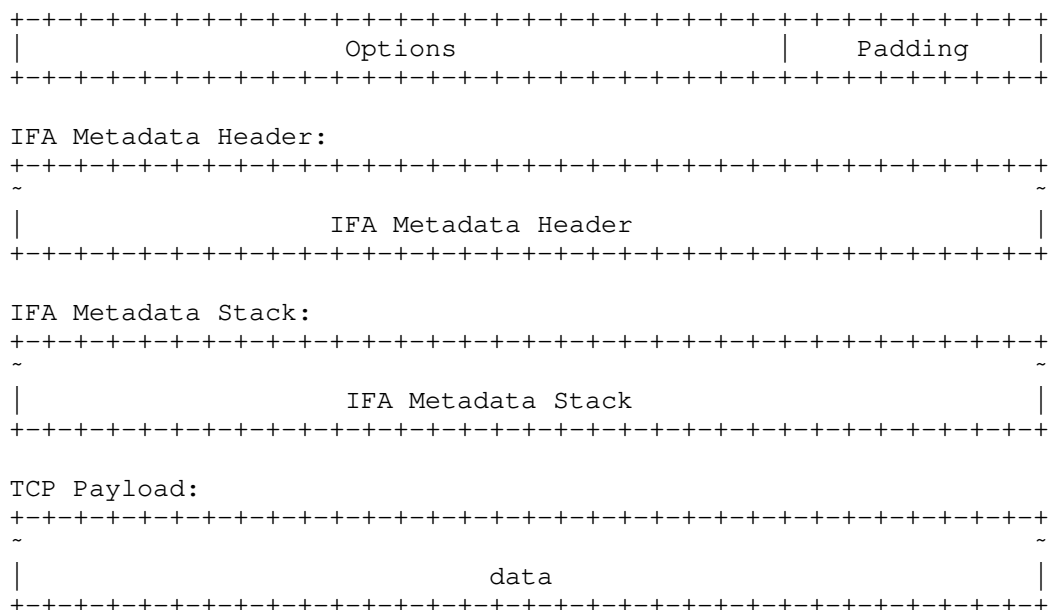
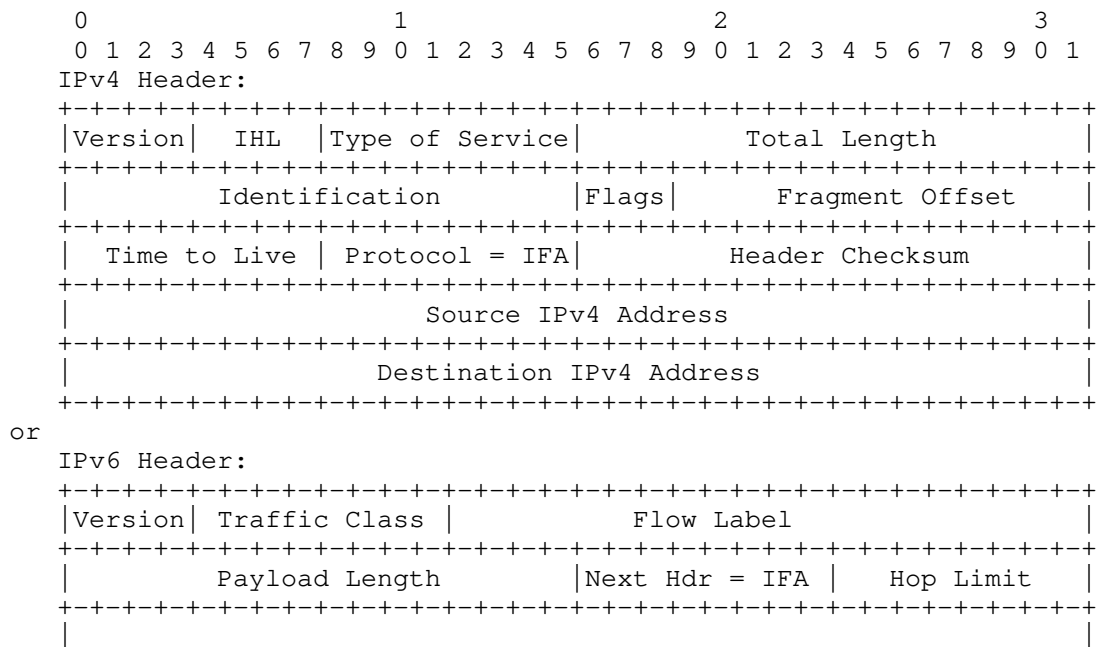


Figure 8: TCP/UDP IFA Packet Format

## 3.8.2. VxLAN Packet



```

+
|
+
|
+
|
+-----+
|
+
|
+
|
+-----+
|
+
|
+
|
+-----+

```

Source IPv4 Address

Destination IPv6 Address

## IFA Header:

```

+-----+
| Ver=2 | GNS | NextHdr = IP_xx | R | R | R | M | T | I | T | C | Max Length |
|       |   |   |             | F | S |   | A |   |   |       |
+-----+

```

## Outer UDP Header:

```

+-----+
| Source Port | Dest Port = VXLAN Port |
+-----+
|
+-----+
| UDP Length | UDP Checksum |
+-----+

```

## IFA Metadata Header:

```

+-----+
~
| IFA Metadata Header |
+-----+

```

## IFA Metadata Stack:

```

+-----+
~
| IFA Metadata Stack |
+-----+

```

## VXLAN Header:

```

+-----+
| R | R | R | R | I | R | R | R | Reserved |
+-----+
| VXLAN Network Identifier (VNI) | Reserved |
+-----+

```

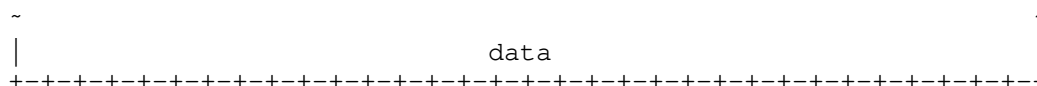
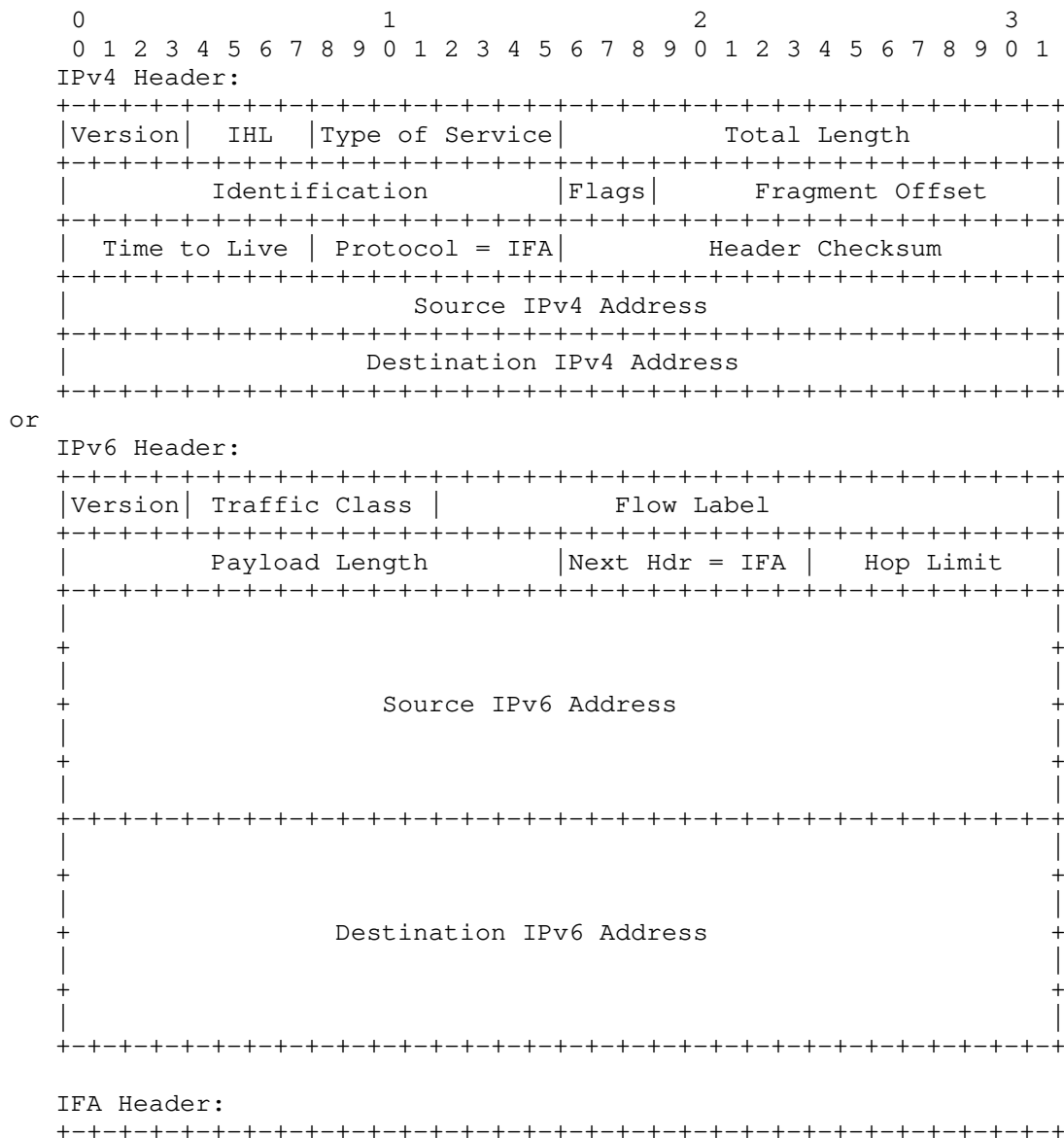


Figure 9: VxLAN IFA Packet Format

## 3.8.3. GRE Packet



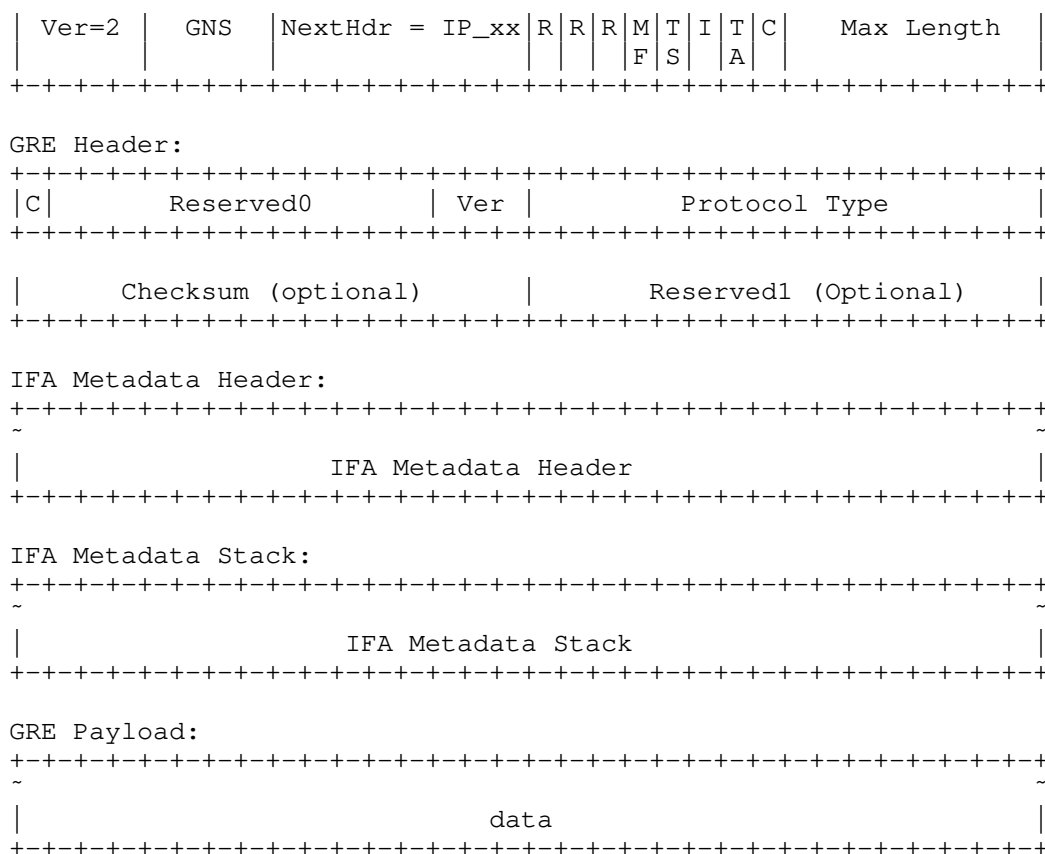
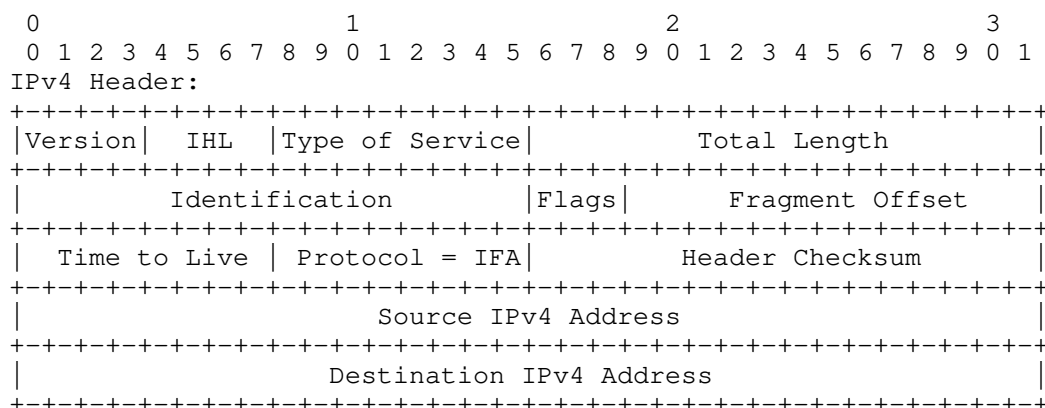


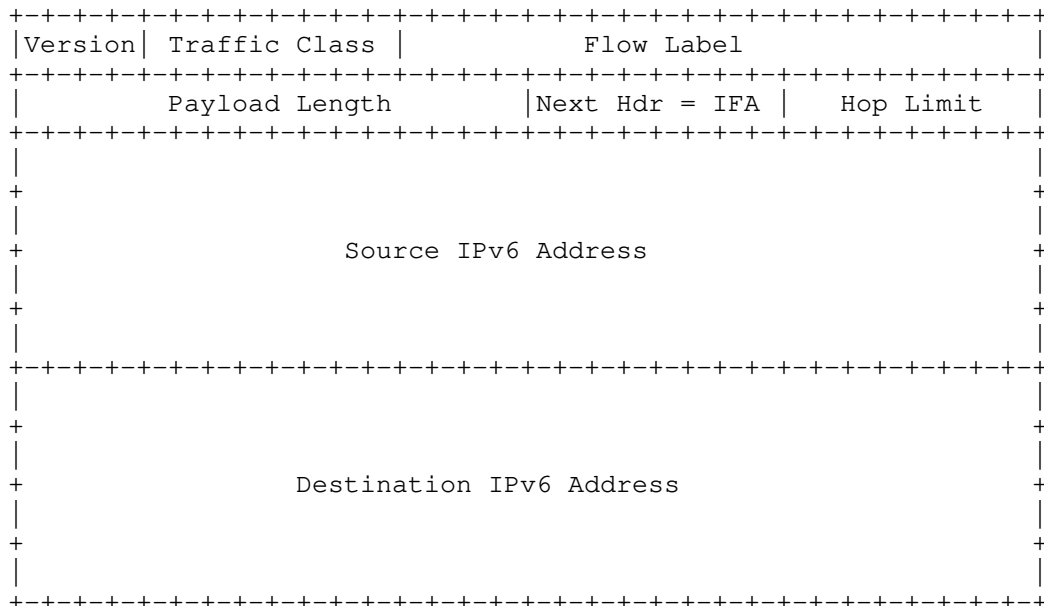
Figure 12 GRE IFA Packet Format

## 3.8.4. Geneve Packet

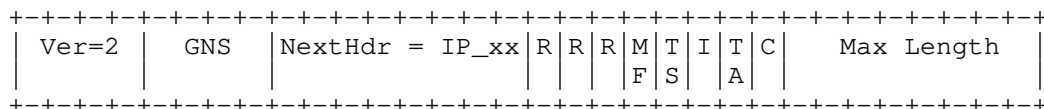


or

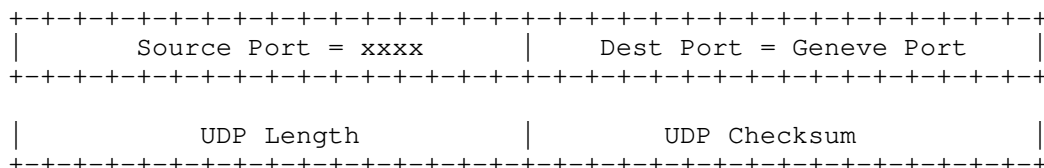
## IPv6 Header:



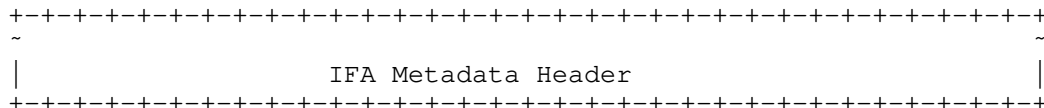
## IFA Header:



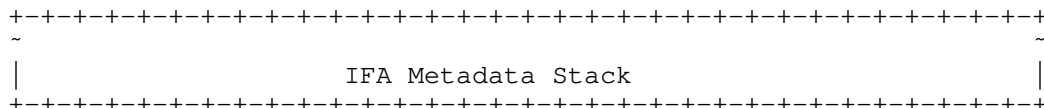
## Outer UDP Header:



## IFA Metadata Header:



## IFA Metadata Stack:



Geneve Header:

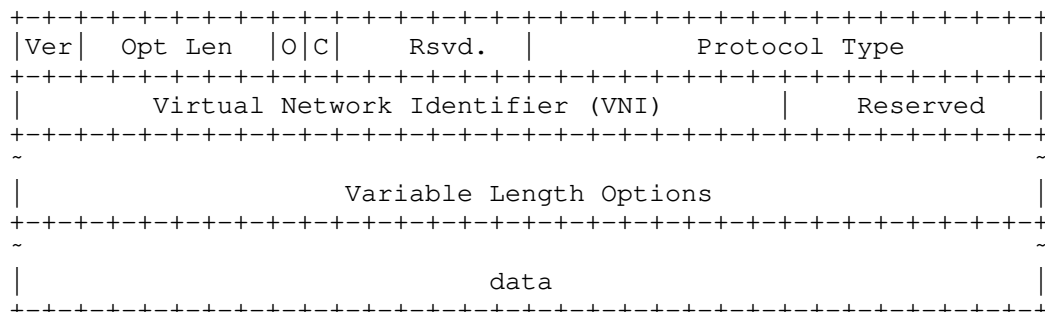
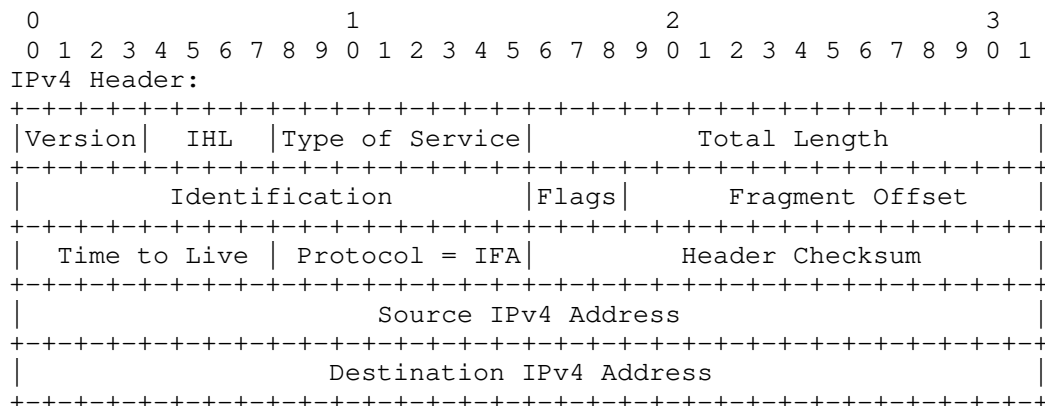


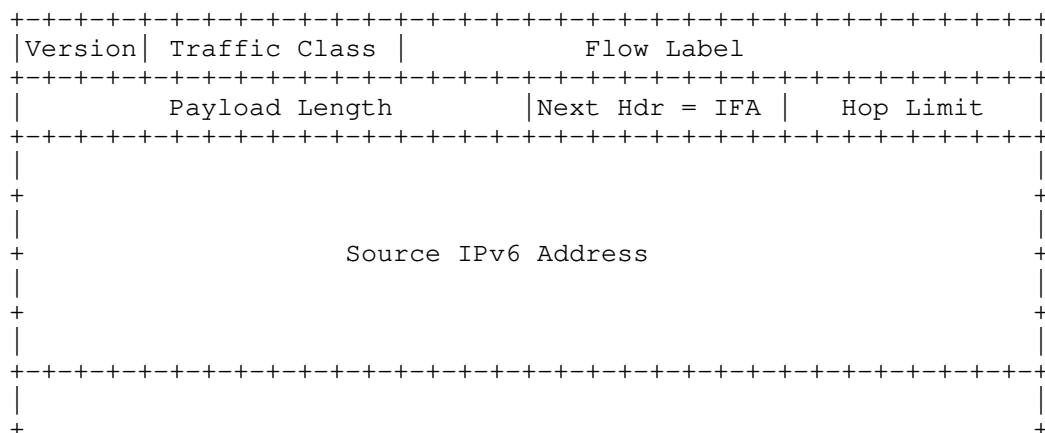
Figure 10: Geneve IFA Packet Format

### 3.8.5. IPinIP Packet



or

IPv6 Header:





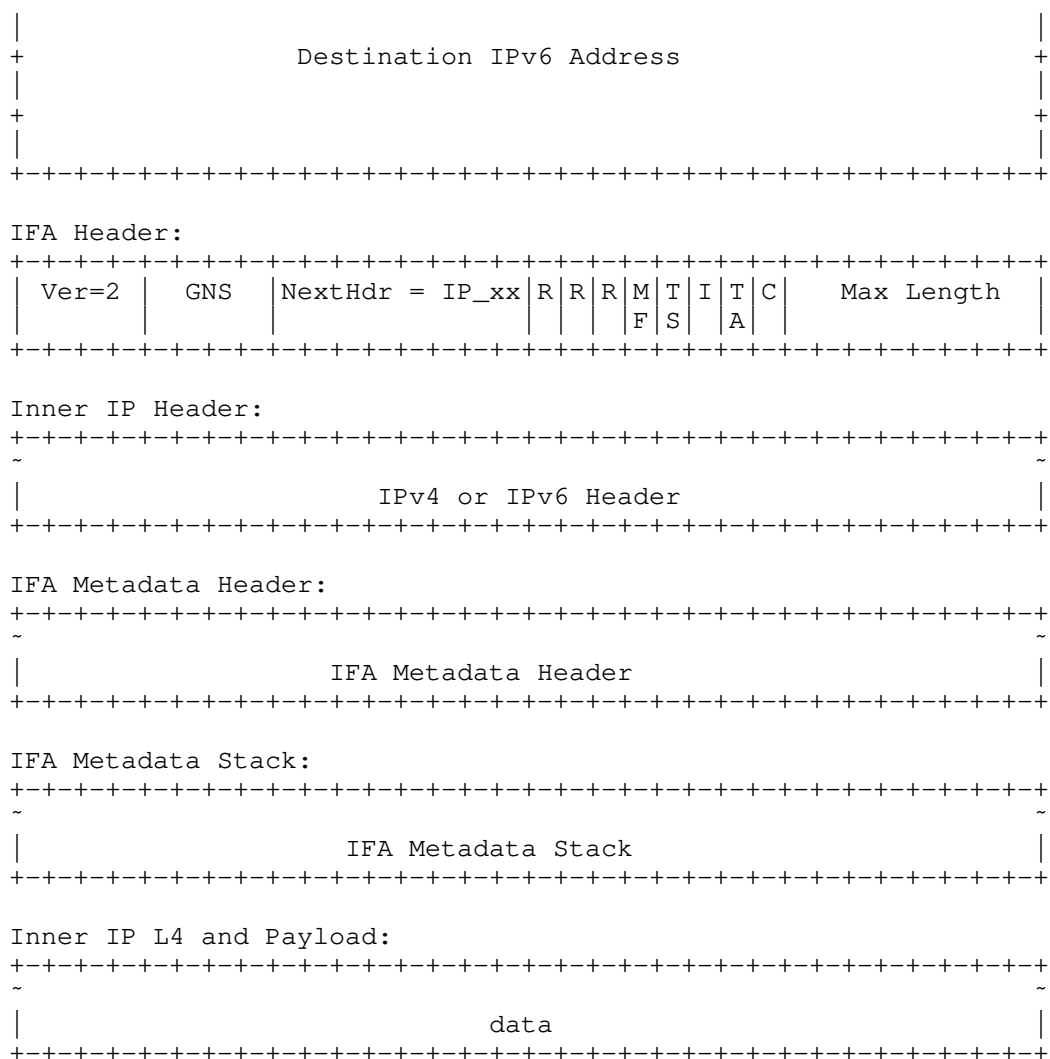
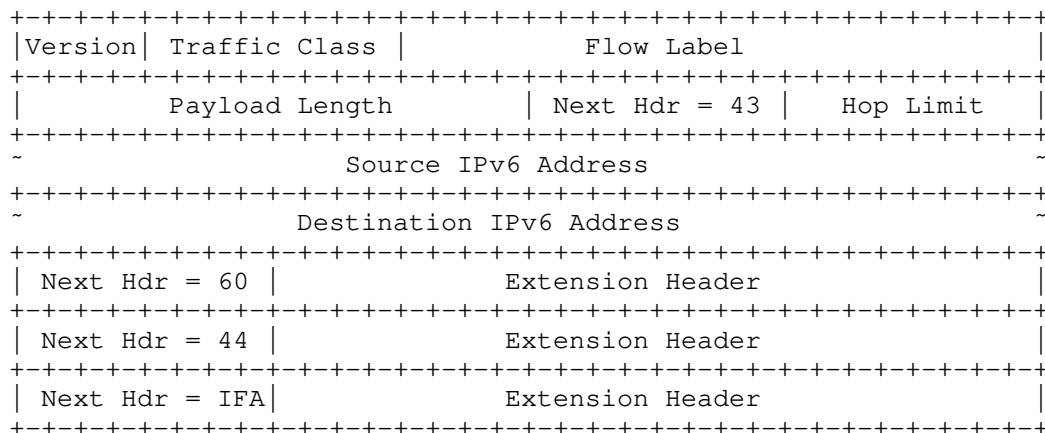


Figure 11: IPinIP IFA Packet Format

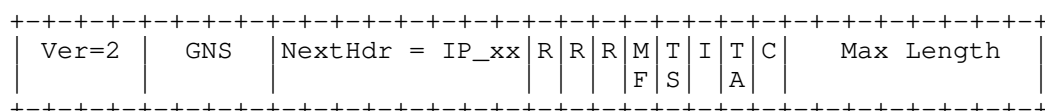
### 3.8.6. IPv6 Extension Headers with IFA

The IFA header is always the last extension header in the IPv6 extension header chain. The last extension header's next header field is stored in the IFA next header field and is replaced by the IFA protocol value.

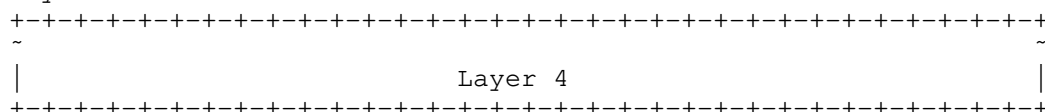
## IPv6 Header:



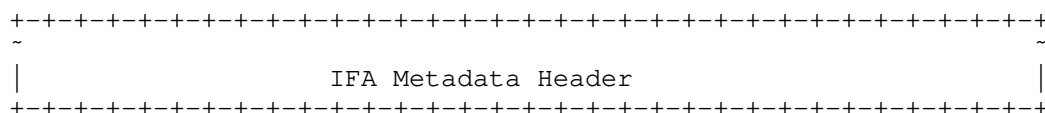
## IFA Header:



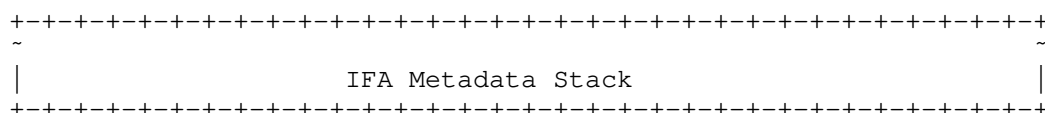
## Layer 4 Header:



## IFA Metadata Header:



## IFA Metadata Stack:



## UDP/TCP Payload:

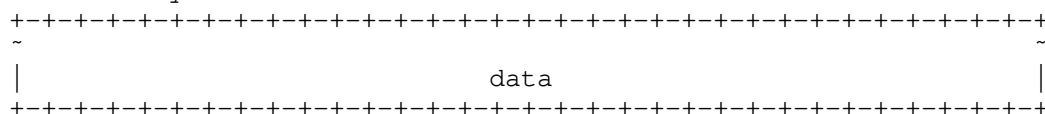


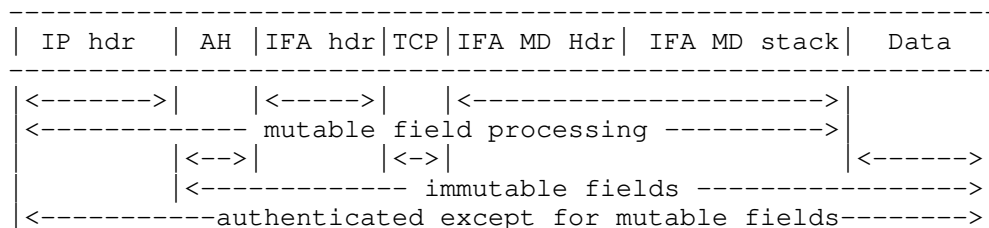
Figure 12: IPv6 Extension Header with IFA Packet Format

## 3.8.7. IP AH/ESP/WESP Packet

An AH, ESP, or WESP header is treated as a chained header in IPv4. The IPv4 protocol field is replaced by the AH/ESP/WESP protocol value and the IPv4 protocol field value is stored in the AH/ESP/WESP next header field.

The IFA header is ALWAYS placed as the last header in a header chain. In case of ESP/WESP where layer 4 and payload is encrypted, IFA metadata stack is placed immediately after IFA header.

## IPv4: AH Transport Mode



## IPv6: AH Transport Mode

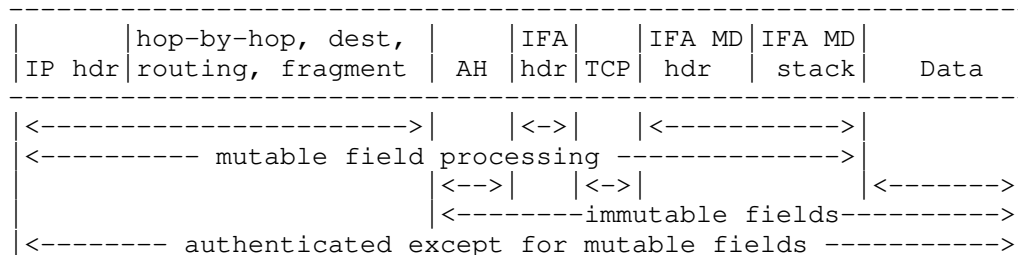
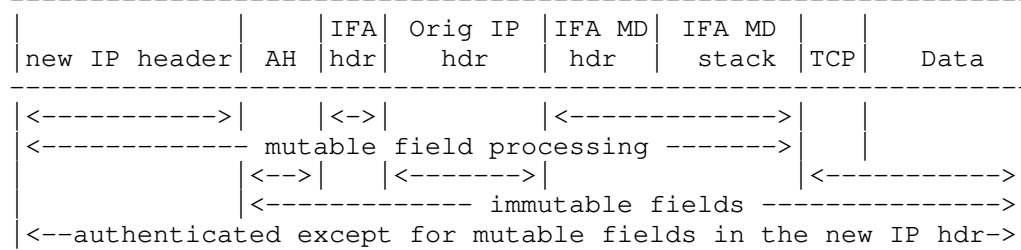


Figure 13: IP AH Transport Mode IFA Packet Format

## IPv4: AH Tunnel Mode



## IPv6: AH Tunnel Mode

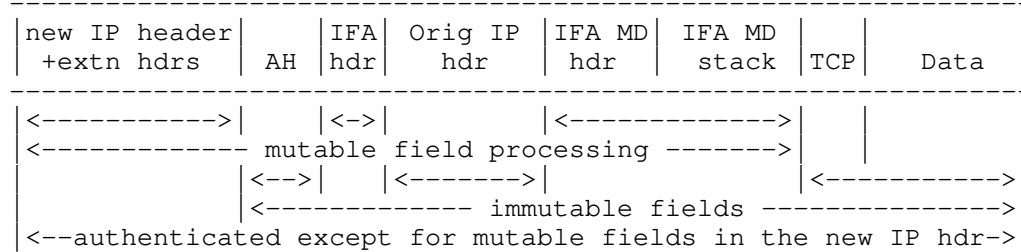
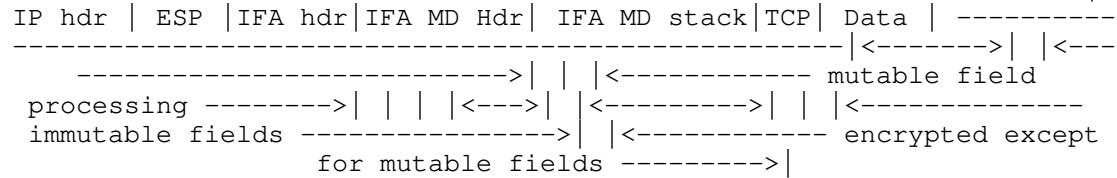


Figure 14: IP AH Tunnel Mode IFA Packet Format IPv4: ESP Transport Mode



## IPv6: ESP Transport Mode

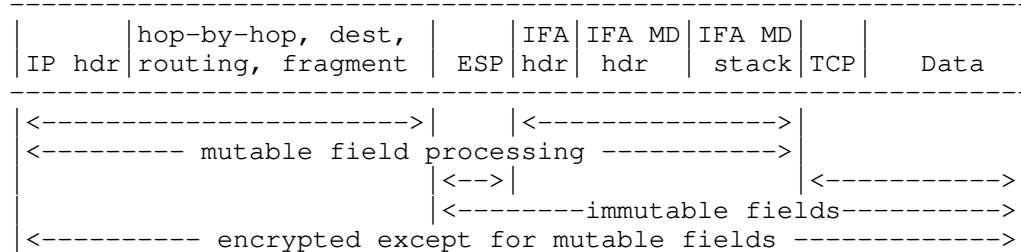
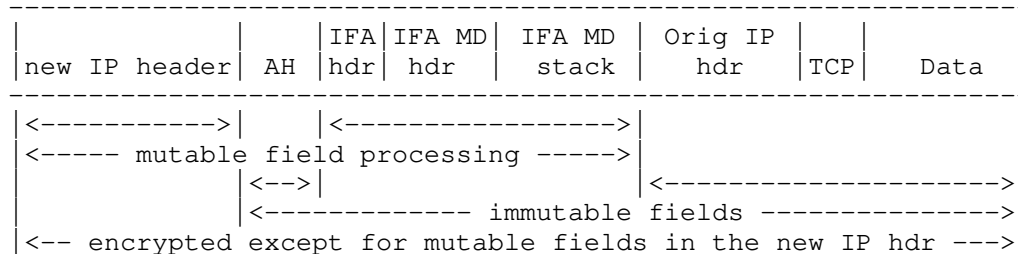


Figure 15: IP ESP Transport Mode IFA Packet Format

## IPv4: ESP Tunnel Mode



## IPv4: ESP Tunnel Mode

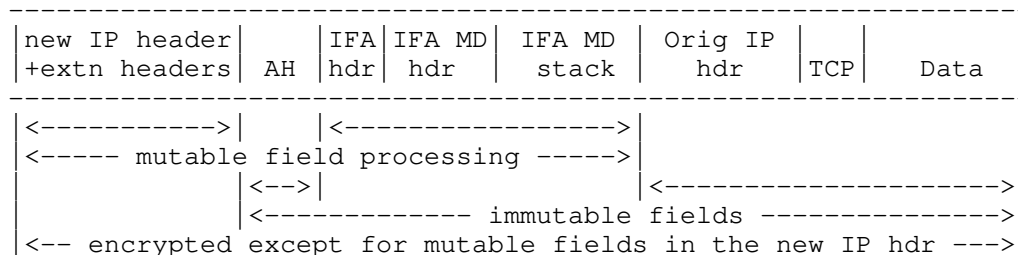


Figure 16: IP AH Tunnel Mode IFA Packet Format

## 3.9. IFA Load Balancing

IFA changes the IP protocol field value to the IFA protocol number. The IP protocol field value is included in the hash computation. This will impact load balancing of flows.

The forwarding plane MUST support reading the IP protocol field value stored in the IFA NextHDR field for hash computation.

The layer 4 header is available at a fixed offset from the IFA header and is available for hash computation.

Hash computation based on the layer 4 payload will depend on the length of the IFA metadata stack present.

## 4. Interoperability Considerations

Version 2 of this protocol specification is not backward compatible with version 1.

## 5. Security Considerations

A successful attack on an OAM protocol can prevent the detection of failures or anomalies, or create a false illusion of nonexistent ones.

The metadata elements of IFA can be used by attackers to collect information about the network hops.

Adding IFA headers or adding to IFA metadata can be used to consume resources within the path being monitored or by a collector.

Adding IFA headers or adding to IFA metadata can be used to force exceeding the MTU for the path being monitored resulting in fragmentation and/or packet drops.

IFA is expected to be deployed within controlled network domains, containing attacks to that controlled domain. Limiting or preventing monitoring or attacks using IFA requires limiting or preventing unauthorized access to the domain in which IFA is to be used, and preventing leaking IFA metadata beyond the controlled domain.

## 6. References

### 6.1. 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>>.

### 6.2. Informative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <<https://www.rfc-editor.org/info/rfc6864>>.
- [RFC3514] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, DOI 10.17487/RFC3514, April 2003, <<https://www.rfc-editor.org/info/rfc3514>>.

[I-D.kumar-ifa]

Kumar, J., Anubolu, S., He, Z., Manur, R., Cai, D., Ou, H., Yizhou, L., and S. Suwei, "Inband Flow Analyzer", draft-kumar-ifa-00 (work in progress), March 2018.

## Appendix A.

Appendix A is for informational purposes only. The following options were considered for the IFA protocol.

### A.1. Probe Marker

One of the challenges of using probe signatures in an IFA header is a false positive.

The IFA version 2 header takes care of large header sizes and identification based on probe markers. Probe markers can cause false positives if there is a match on the first 64 bits of the layer 4 payload.

This approach is not a preferred approach, but is supported by this draft as a version 1.0 header.

### A.2. DSCP

[RFC791] EXP/LSB Pool 3 can be used for identifying IFA packets. CU bits can be used for identifying IFA packets.

The problem with using TOS bits is that they are pervasively used in the network deployment and are responsible for affecting the forwarding decision.

This approach is not supported or recommended by this draft.

### A.3. IP Options

[RFC791] The IP options provide for control functions that are needed or useful in some situations but unnecessary for the most common communications. The IP options include provisions for timestamps, security, and special routing.

There are various problems with this approach.

(1) The IPv4 header size can become arbitrarily large with the presence of options.

(2) A switch pipeline typically handles IP option packets as exception path processing and punts them to a host CPU. (3) IP options make the construction of firewalls cumbersome, and are

typically disallowed or stripped at the perimeter of enterprise networks by firewalls.

This approach is not supported or recommended by this draft.

#### A.4. IPv4 Identification or Reserved Flag

[RFC6864] [RFC3514] Another suggestion is to use the IPv4 identification field or reserved flag. This suggestion is also discarded and not supported for the following reasons:

[RFC6864] prohibits usage of id field for any other purposes.

[RFC3514] prohibits using flags bit 0 for security reasons.

#### Authors' Addresses

Jai Kumar  
Broadcom Inc.  
Email: jai.kumar@broadcom.com

Surendra Anubolu  
Broadcom Inc.  
Email: surendra.anubolu@broadcom.com

John Lemon  
Broadcom Inc.  
Email: john.lemon@broadcom.com

Rajeev Manur  
Broadcom Inc.  
Email: rajeev.manur@broadcom.com

Hugh Holbrook  
Arista Networks  
Email: holbrook@arista.com

Anoop Ghanwani  
Dell EMC  
Email: anoop.ghanwani@dell.com

Dezhong Cai  
AliBaba Inc.  
Email: d.cai@alibaba-inc.com

Heidi Ou  
AliBaba Inc.  
Email: heidi.ou@alibaba-inc.com

Yizhou Li  
Huawei Technologies  
Email: liyizhou@huawei.com

Xiaojun Wang  
Fujian Ruijie Networks co.,ltd.  
Email: wxj@ruijie.com.cn



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 7, 2020

G. Mirsky  
X. Min  
ZTE Corp.  
G. Jun  
ZTE Corporation  
H. Nydell  
Accedian Networks  
R. Foote  
Nokia  
July 6, 2019

Simple Two-way Active Measurement Protocol Optional Extensions  
draft-mirsky-ippm-stamp-option-tlv-05

Abstract

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

Status of This Memo

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 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                    |    |
|----------------------------------------------------|----|
| 1. Introduction . . . . .                          | 2  |
| 2. Conventions used in this document . . . . .     | 2  |
| 2.1. Terminology . . . . .                         | 2  |
| 2.2. Requirements Language . . . . .               | 3  |
| 3. Theory of Operation . . . . .                   | 3  |
| 4. TLV Extensions to STAMP . . . . .               | 4  |
| 4.1. Extra Padding TLV . . . . .                   | 6  |
| 4.2. Location TLV . . . . .                        | 6  |
| 4.3. Timestamp Information TLV . . . . .           | 8  |
| 4.4. Class of Service TLV . . . . .                | 9  |
| 4.5. Direct Measurement TLV . . . . .              | 10 |
| 5. IANA Considerations . . . . .                   | 11 |
| 5.1. STAMP TLV Registry . . . . .                  | 11 |
| 5.2. Synchronization Source Sub-registry . . . . . | 12 |
| 5.3. Timestamping Method Sub-registry . . . . .    | 13 |
| 6. Security Considerations . . . . .               | 14 |
| 7. Acknowledgments . . . . .                       | 14 |
| 8. References . . . . .                            | 14 |
| 8.1. Normative References . . . . .                | 14 |
| 8.2. Informative References . . . . .              | 15 |
| Authors' Addresses . . . . .                       | 15 |

## 1. Introduction

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

## 2. Conventions used in this document

### 2.1. Terminology

STAMP - Simple Two-way Active Measurement Protocol

DSCP - Differentiated Services Code Point

ECN - Explicit Congestion Notification

NTP - Network Time Protocol

PTP - Precision Time Protocol

HMAC Hashed Message Authentication Code

TLV Type-Length-Value

BITS Building Integrated Timing Supply

SSU Synchronization Supply Unit

GPS Global Positioning System

GLONASS Global Orbiting Navigation Satellite System

LORAN-C Long Range Navigation System Version C

MBZ Must Be Zeroed

CoS Class of Service

## 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. Theory of Operation

STAMP Session-Sender transmits test packets to STAMP Session-Reflector. STAMP Session-Reflector receives Session-Sender's packet and acts according to the configuration and optional control information communicated in the Session-Sender's test packet. STAMP defines two different test packet formats, one for packets transmitted by the STAMP-Session-Sender and one for packets transmitted by the STAMP-Session-Reflector. STAMP supports two modes: unauthenticated and authenticated. Unauthenticated STAMP test packets are compatible on the wire with unauthenticated TWAMP-Test [RFC5357] packet formats.

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

## 4. TLV Extensions to STAMP

Figure 1 displays the format of STAMP Session-Sender test packet in unauthenticated mode that includes a TLV.

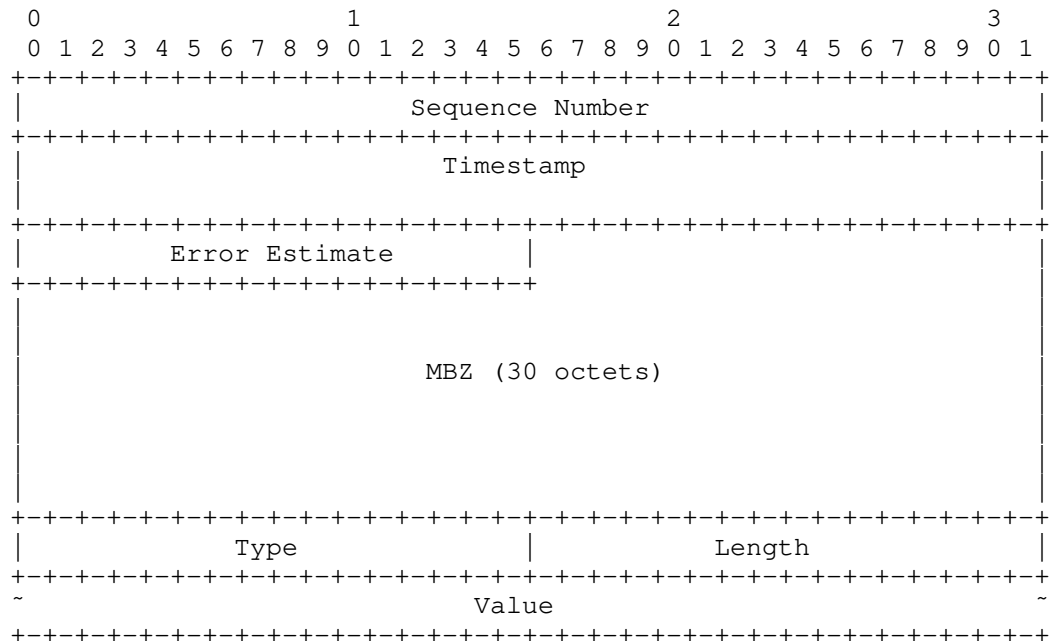


Figure 1: STAMP Session-Sender test packet format with TLV in unauthenticated mode

The MBZ (Must Be Zeroed) field of a test packet transmitted by a STAMP Session-Sender MUST be 30 octets long. A STAMP Session-Sender test packet MUST NOT use the Reflect Octets capability defined in [RFC6038].

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

describe the use of TLVs for STAMP that extend STAMP capability beyond its base specification.

Figure 2 displays the format of STAMP Session-Reflector test packet in unauthenticated mode that includes a TLV.

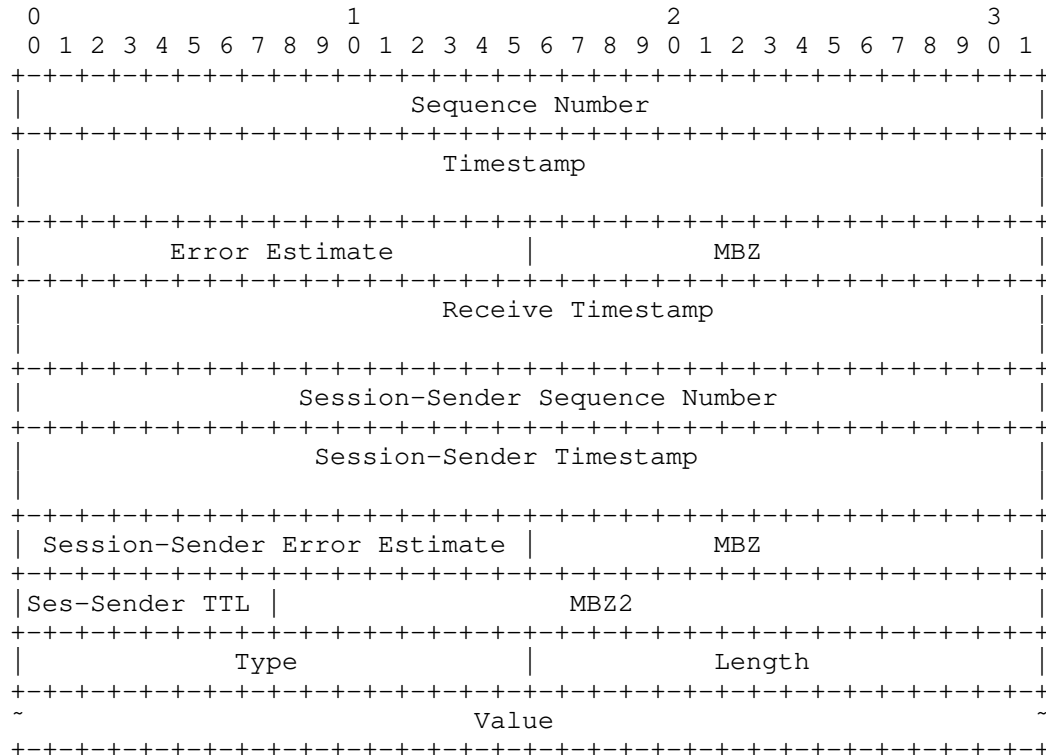


Figure 2: STAMP Session-Reflector test packet format with TLV in unauthenticated mode

The MBZ2 field of a test packet transmitted by a STAMP Session-Reflector MUST be 3 octets long.

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

## 4.1. Extra Padding TLV

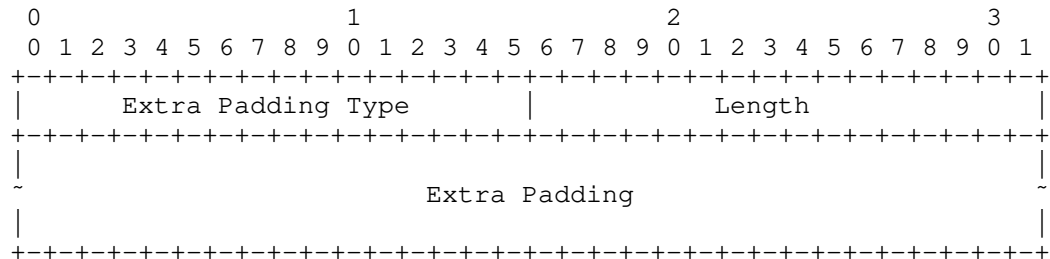


Figure 3: Extra Padding TLV

where fields are defined as the following:

- o Extra Padding Type - TBA1 allocated by IANA Section 5.1
- o Length - 2 octets long field equals length on the Extra Padding field in octets.
- o Extra Padding - a pseudo-random sequence of numbers. The field MAY be filled with all zeroes.

The Extra Padding TLV is similar to the Packet Padding field in TWAMP-Test packet [RFC5357]. The in STAMP the Packet Padding field is used to ensure symmetrical size between Session-Sender and Session-Reflector test packets. Extra Padding TLV MUST be used to create STAMP test packets of larger size.

## 4.2. Location TLV

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

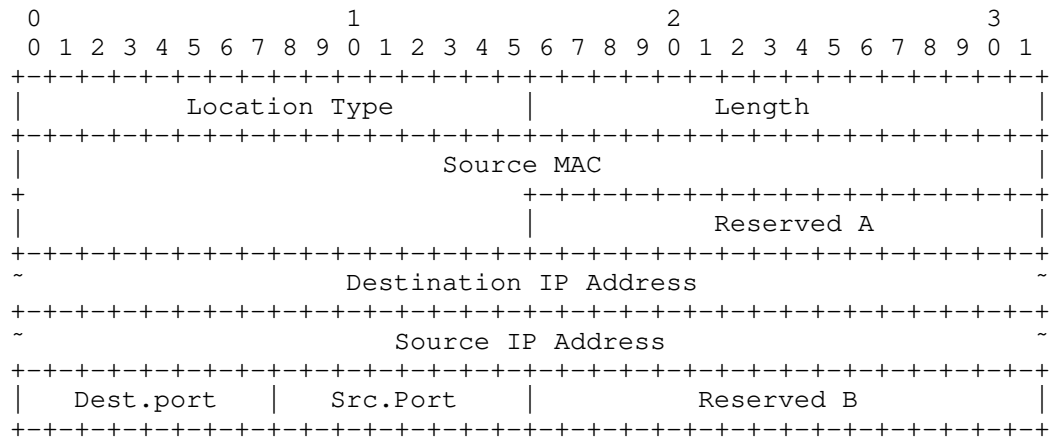


Figure 4: Session-Reflector Location TLV

where fields are defined as the following:

- o Location Type - TBA2 allocated by IANA Section 5.1
- o Length - 2 octets long field equals length on the Value field in octets. Length field value MUST be 20 octets for the IPv4 address family. For the IPv6 address family value of the Length field MUST be 44 octets. All other values are invalid.
- o Source MAC - 6 octets 48 bits long field. The session-reflector MUST copy Source MAC of received STAMP packet into this field.
- o Reserved A - two octets long field. MUST be zeroed on transmission and ignored on reception.
- o Destination IP Address - IPv4 or IPv6 destination address of the received by the session-reflector STAMP packet.
- o Source IP Address - IPv4 or IPv6 source address of the received by the session-reflector STAMP packet.
- o Dest.port - one octet long UDP destination port number of the received STAMP packet.
- o Src.port - one octet long UDP source port number of the received STAMP packet.
- o Reserved B - two octets long field. MUST be zeroed on transmission and ignored on reception.

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

#### 4.3. Timestamp Information TLV

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

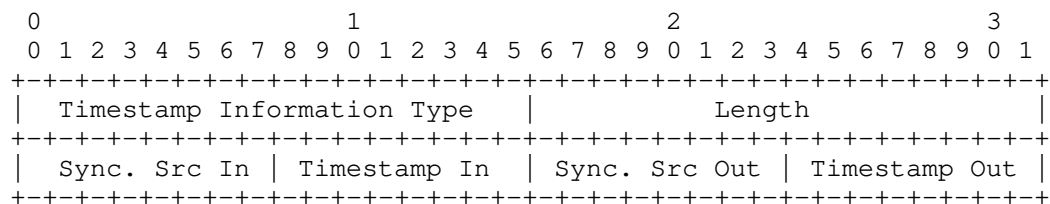


Figure 5: Timestamp Information TLV

where fields are defined as the following:

- o Timestamp Information Type - TBA3 allocated by IANA Section 5.1
- o Length - 2 octets long field, equals four octets.
- o Sync Src In - one octet long field that characterizes the source of clock synchronization at the ingress of Session-Reflector. There are several of methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905], Precision Time Protocol (PTP) [IEEE.1588.2008], Synchronization Supply Unit (SSU) or Building Integrated Timing Supply (BITS), or Global Positioning System (GPS), Global Orbiting Navigation Satellite System (GLONASS) and Long Range Navigation System Version C (LORAN-C). The value is one of Section 5.2.



- o Timestamp In - one octet long field that characterizes the method by which the ingress of Session-Reflector obtained the timestamp T2. A timestamp may be obtained with hardware assist, via software API from a local wall clock, or from a remote clock (the latter referred to as "control plane"). The value is one of Section 5.3.
- o Sync Src Out - one octet long field that characterizes the source of clock synchronization at the egress of Session-Reflector. The value is one of Section 5.2.
- o Timestamp Out - one octet long field that characterizes the method by which the egress of Session-Reflector obtained the timestamp T3. The value is one of Section 5.3.

#### 4.4. Class of Service TLV

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

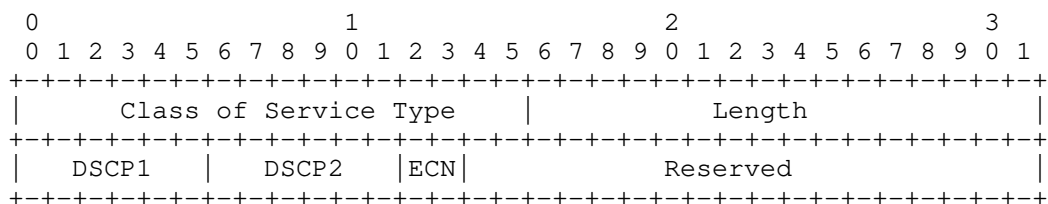


Figure 6: Class of Service TLV

where fields are defined as the following:

- o Class of Service Type - TBA4 allocated by IANA Section 5.1
- o Length - 2 octets long field, equals four octets.

- o DSCP1 - The Differentiated Services Code Point (DSCP) intended by the Session-Sender. To be used as the return DSCP from the Session-Reflector.
- o DSCP2 - The received value in the DSCP field at the Session-Reflector in the forward direction.
- o ECN - The received value in the ECN field at the Session-Reflector in the forward direction.
- o Reserved - 18 bits long field, must be zeroed in transmission and ignored on receipt.

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

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

#### 4.5. Direct Measurement TLV

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

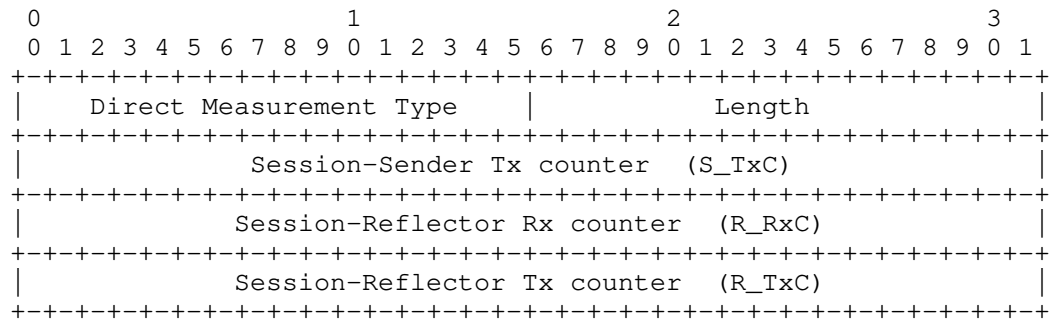


Figure 7: Direct Measurement TLV

where fields are defined as the following:

- o Direct Measurement Type - TBA5 allocated by IANA Section 5.1
- o Length - 2 octets long field equals length on the Value field in octets. Length field value MUST be 12 octets.
- o Session-Sender Tx counter (S\_TxC) is four octets long field.
- o Session-Reflector Rx counter (R\_RxC) is four octets long field. MUST be zeroed by the Session-Sender and filled by the Session-Reflector.
- o Session-Reflector Tx counter (R\_TxC) is four octets long field. MUST be zeroed by the Session-Sender and filled by the Session-Reflector.

## 5. IANA Considerations

### 5.1. STAMP TLV Registry

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

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

Table 1: STAMP TLV Type Registry

This document defines the following new values in 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 |

Table 2: STAMP Types

## 5.2. Synchronization Source Sub-registry

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

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

Table 3: Synchronization Source Sub-registry

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

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

Table 4: Synchronization Sources

### 5.3. Timestamping Method Sub-registry

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

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

Table 5: Timestamping Method Sub-registry

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

| Value | Description   | Reference     |
|-------|---------------|---------------|
| 1     | HW assist     | This document |
| 2     | SW local      | This document |
| 3     | Control plane | This document |

Table 6: Timestamping Methods

## 6. Security Considerations

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

## 7. Acknowledgments

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

## 8. References

### 8.1. Normative References

[I-D.ietf-ippm-stamp]

Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-way Active Measurement Protocol", draft-ietf-ippm-stamp-06 (work in progress), April 2019.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [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>>.
- [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>>.

## 8.2. Informative References

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

## Authors' Addresses

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Xiao Min  
ZTE Corp.

Email: [xiao.min2@zte.com.cn](mailto:xiao.min2@zte.com.cn)

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

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

Henrik Nydell  
Accedian Networks

Email: hnydell@accedian.com

Richard Foote  
Nokia

Email: footer.foote@nokia.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 7, 2020

T. Mizrahi  
Huawei Network.IO Innovation Lab  
C. Arad

G. Fioccola  
Huawei Technologies  
M. Cociglio  
Telecom Italia  
M. Chen  
L. Zheng  
Huawei Technologies  
G. Mirsky  
ZTE Corp.  
July 6, 2019

Compact Alternate Marking Methods for Passive and Hybrid Performance  
Monitoring  
draft-mizrahi-ippm-compact-alternate-marking-05

Abstract

This memo introduces new alternate marking methods that require a compact overhead of either a single bit per packet, or zero bits per packet. This memo also presents a summary of alternate marking methods, and discusses the tradeoffs among them. The target audience of this document is network protocol designers; this document is intended to help protocol designers choose the best alternate marking method(s) based on the protocol's constraints and requirements.

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 7, 2020.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                               |    |
|---------------------------------------------------------------|----|
| 1. Introduction . . . . .                                     | 3  |
| 1.1. Background . . . . .                                     | 3  |
| 1.2. The Scope of This Document . . . . .                     | 4  |
| 2. Terminology . . . . .                                      | 5  |
| 2.1. Requirements Language . . . . .                          | 5  |
| 2.2. Abbreviations . . . . .                                  | 5  |
| 3. Marking Abstractions . . . . .                             | 5  |
| 4. Double Marking . . . . .                                   | 7  |
| 5. Single-bit Marking . . . . .                               | 8  |
| 5.1. Single Marking Using the First Packet . . . . .          | 8  |
| 5.2. Single Marking using the Mean Delay . . . . .            | 8  |
| 5.3. Single Marking using a Multiplexed Marking Bit . . . . . | 8  |
| 5.3.1. Overview . . . . .                                     | 8  |
| 5.4. Pulse Marking . . . . .                                  | 9  |
| 6. Zero Marking Hashed . . . . .                              | 10 |
| 6.1. Hash-based Sampling . . . . .                            | 10 |
| 6.1.1. Hashed Pulse Marking . . . . .                         | 11 |
| 6.1.2. Hashed Step Marking . . . . .                          | 11 |
| 7. Single Marking Hashed . . . . .                            | 11 |
| 8. Timing and Synchronization Aspects . . . . .               | 12 |
| 8.1. Synchronization Aspects in Multiplexed Marking . . . . . | 13 |
| 9. Multipoint Marking Methods . . . . .                       | 14 |
| 10. Summary of Marking Methods . . . . .                      | 15 |
| 11. Alternate Marking using Reserved Values . . . . .         | 19 |
| 12. IANA Considerations . . . . .                             | 20 |
| 13. Security Considerations . . . . .                         | 20 |
| 14. References . . . . .                                      | 20 |
| 14.1. Normative References . . . . .                          | 20 |
| 14.2. Informative References . . . . .                        | 20 |
| Authors' Addresses . . . . .                                  | 21 |

## 1. Introduction

### 1.1. Background

Alternate marking, defined in [RFC8321], is a method for measuring packet loss, packet delay, and packet delay variation. Typical delay measurement protocols require the two measurement points (MPs) to exchange timestamped test packets. In contrast, the alternate marking method does not require control packets to be exchanged. Instead, every data packet carries a marking bit, which is used for triggering measurement events. Note that the frequency of these measurement events is dependent on the users' application(s) and the node characteristics.

The marking bit can be used as a color indication, as defined in [RFC8321], which is toggled periodically. This approach is illustrated in Figure 1.

A: packet with color 0  
B: packet with color 1

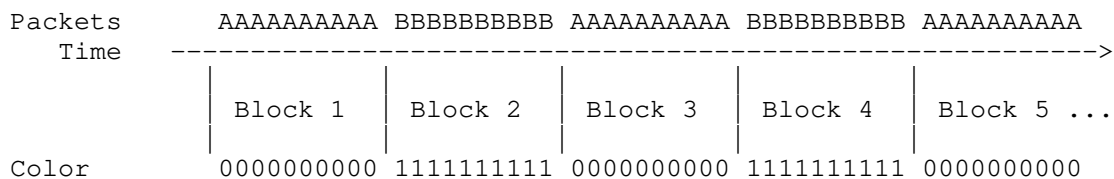


Figure 1: Alternate marking: packets are monitored on a per-color basis.

Alternate marking is used between two MPs, the initiating MP, and the monitoring MP. The initiating MP incorporates the marking field into en-route packets, allowing the monitoring MP to use the marking field in order to bind each packet to the corresponding block.

Each of the MPs maintains two counters, one per color. At the end of each block the counter values can be collected by a central management system, and analyzed; the packet loss can be computed by comparing the counter values of the two MPs.

When using alternate marking delay measurement can be performed in one of three ways (as per [RFC8321]):

- o Single marking using the first packet: in this method each packet uses a single marking bit, used as a color indicator. The first packet of each block is used by both MPs as a reference for delay

measurement. The timestamp of this packet is measured by the two measurement points, and can be collected by the management system from each of the measurement points, which can compute the path delay by comparing the two timestamps. The drawback of this approach is that it is not accurate when packets arrive out-of-order, as the two MPs may have a different view of which packet was the first in the block.

- o Single marking using the mean delay: as in the previous method, each packet uses a single marking method, indicating the color. Each of the MPs computes the average packet timestamp of each block. The management system can then compute the delay by comparing the average times of the two MPs. The drawback of this approach is that it may be computationally heavy, or difficult to implement at the data plane.
- o Double marking: each packet uses two marking bits. One bit is used as a color indicator, and one is used as a timestamping indicator. This method resolves the drawbacks raised for the two previous methods, at the expense of an extra bit in the packet header.

The double marking method is the most straightforward approach. It allows for accurate measurement without incurring expensive computational load. However, in some cases allocating two bits for passive measurement is not possible. For example, if alternate marking is implemented over IPv4, allocating 2 marking bits in the IPv4 header is challenging, as every bit in the 20-octet header is costly; one of the possible approaches discussed in [RFC8321] is to reserve one or two bits from the DSCP field for remarking. In this case every marking bit comes at the expense of reducing the DSCP range by a factor of two.

## 1.2. The Scope of This Document

This memo extends the marking methods of [RFC8321], and introduces methods that require a single marking bit, or zero marking bits.

Two single-bit marking methods are proposed, multiplexed marking and pulse marking. In multiplexed marking the color indicator and the timestamp indicator are multiplexed into a single bit, providing the advantages of the double marking method while using a single bit in the packet header. In pulse marking both delay and loss measurement are triggered by a 'pulse' value in a single marking field.

This document also discusses zero-bit marking methods that leverage well-known hash-based selection approaches ([RFC5474], [RFC5475]).

Alternate marking is discussed in this memo as a single-bit or a two-bit marking method. However, these methods can similarly be applied to larger fields, such as an IPv6 Flow Label or an MPLS Label; single-bit marking can be applied using two reserved values, and two-bit marking can be applied using four reserved values. Marking based on reserved values is further discussed in this document, including its application to MPLS and IPv6.

Finally, this memo summarizes the alternate marking methods, and discusses the tradeoffs among them. It is expected that different network protocols will have different constraints, and therefore may choose to use different alternate marking methods. In some cases it may be preferable to support more than one marking method; in this case the particular marking method may be signaled through the control plane.

## 2. Terminology

### 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 RFC 2119 [RFC2119].

### 2.2. Abbreviations

The following abbreviations are used in this document:

|      |                                                     |
|------|-----------------------------------------------------|
| DSCP | Differentiated Services Code Point                  |
| DM   | Delay Measurement                                   |
| LM   | Loss Measurement                                    |
| LSP  | Label Switched Path                                 |
| MP   | Measurement Point                                   |
| MPLS | Multiprotocol Label Switching                       |
| SFL  | Synonymous Flow Label [I-D.ietf-mpls-sfl-framework] |

## 3. Marking Abstractions

The marking methods that were discussed in Section 1, as well as the methods introduced in this document, use two basic abstractions, pulse detection, and step detection.





## 5. Single-bit Marking

### 5.1. Single Marking Using the First Packet

This method uses a single marking bit that indicates the color, as described in [RFC8321]. Both LM and DM are implemented using a step-based approach; LM is implemented using two color-based counters per flow. The first packet of every period is used by the two MPs as the reference for measuring the delay. As denoted above, the delay computed in this method may be erroneous when packets are delivered out-of-order.

A: packet with color 0  
B: packet with color 1

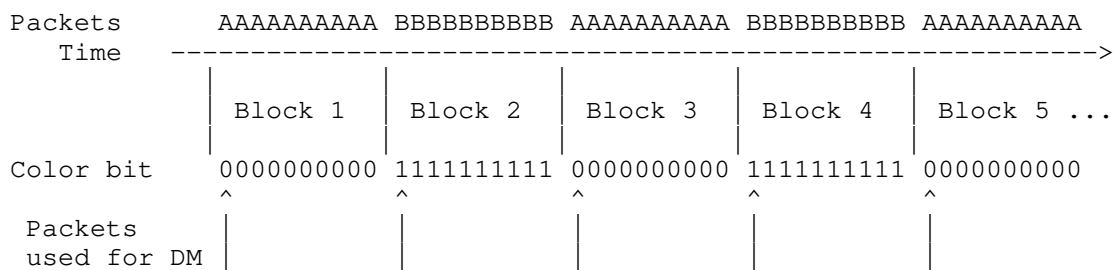


Figure 5: Single marking using the first packet of the block.

### 5.2. Single Marking using the Mean Delay

As in the first-packet approach, in the mean delay approach ([RFC8321]) a single marking bit is used to indicate the color, enabling step-based loss measurement. Delay is measured in each period by averaging the measured delay over all the packets in the period. As discussed above, this approach is not sensitive to out-of-order delivery, but may be heavy from a computational perspective.

### 5.3. Single Marking using a Multiplexed Marking Bit

#### 5.3.1. Overview

This section introduces a method that uses a single marking bit that serves two purposes: a color indicator, and a timestamp indicator. The double marking method that was discussed in the previous section uses two 1-bit values: a color indicator C, and a timestamp indicator T. The multiplexed marking bit, denoted by M, is an exclusive or between these two values:  $M = C \text{ XOR } T$ .



An example of the use of the multiplexed marking bit is depicted in Figure 6. The example considers two routers, R1 and R2, that use the multiplexed bit method to measure traffic from R1 to R2. In each block R1 designates one of the packets for delay measurement. In each of these designated packets the value of the multiplexed bit is reversed compared to the other packets in the same block, allowing R2 to distinguish the designated packets from the other packets.

A: packet with color 0  
B: packet with color 1

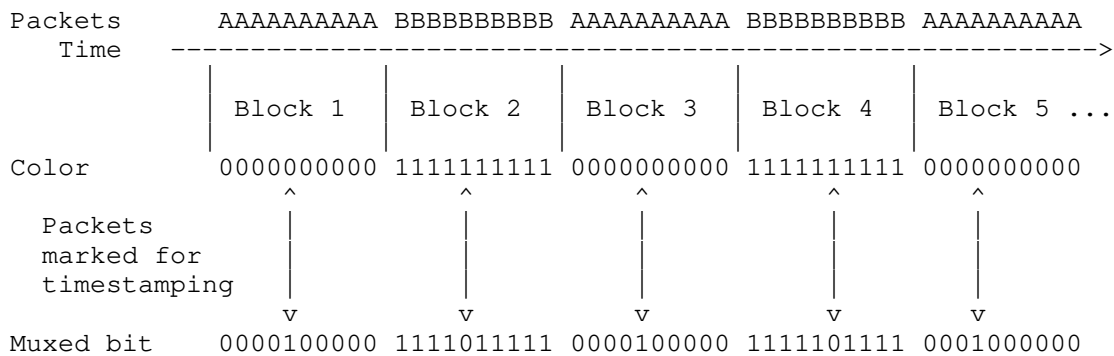


Figure 6: Alternate marking with multiplexed bit.

#### 5.4. Pulse Marking

Pulse marking uses a single marking bit that is used as a trigger for both LM and DM. In this method the two MPs maintain a single per-flow counter for LM, in contrast to the color-based methods which require two counters per flow. In each block one of the packets is marked. The marked packet triggers two actions in each of MPs:

- o The timestamp is captured for DM.
- o The value of the counter is captured for LM.

In each period, each of the MPs exports the timestamp and counter-stamp to the management system, which can then compute the loss and delay in that period. It should be noted that as in [RFC8321], if the length of the measurement period is  $L$  time units, then all network devices must be synchronized to the same clock reference with an accuracy of  $\pm L/2$  time units.

The pulse marking approach is illustrated in Figure 7. Since both LM and DM use a pulse-based trigger, if the marked packet is lost then no measurement is available in this period. Moreover, the LM accuracy may be affected by out-of-order delivery.

P: packet - all packets have the same color

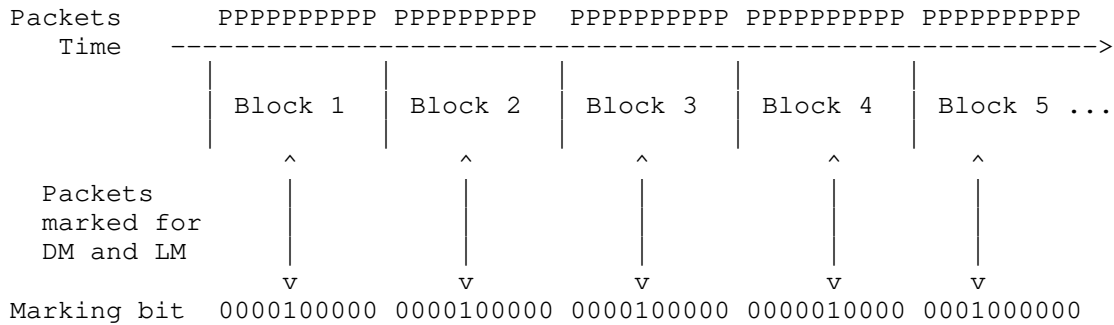


Figure 7: Pulse marking method.

## 6. Zero Marking Hashed

### 6.1. Hash-based Sampling

Hash based selection [RFC5475] is a well-known method for sampling a subset of packets. As defined in [RFC5475]:

A Hash Function  $h$  maps the Packet Content  $c$ , or some portion of it, onto a Hash Range  $R$ . The packet is selected if  $h(c)$  is an element of  $S$ , which is a subset of  $R$  called the Hash Selection Range.

Hash-based selection can be leveraged as a marking method, allowing a zero-bit marking approach. Specifically, the pulse and step abstractions can be implemented using hashed selection:

- o Hashed pulse-based trigger: in this approach, a packet is selected if  $h(c)$  is an element of  $S$ , which is a strict subset of the hash range  $R$ . When  $|S| \ll |R|$ , the average sampling period is long, reducing the probability of ambiguity between consecutive packets.  $|S|$  and  $|R|$  denote the number of elements in  $S$  and  $R$ , respectively.
- o Hashed step-based trigger: the hash values of a given traffic flow are said to be monotonically increasing if for two packets  $p_1$  and

p2, if p1 is sent before p2 then  $h(p1) \leq h(p2)$ . If it is guaranteed that the hash values of a flow are monotonically increasing, then a step-based approach can be used on the range R. For example, in an IPv4 flow the Identification field can be used as the hash value of each packet. Since the Identification field is monotonically increasing, the step-based trigger can be implemented using consecutive ranges of the Identification value. For example, the fourth bit of the Identification field is toggled every 8 packets. Thus, a possible hash function simply takes the fourth bit of the Identification field as the hash value. This hash value is toggled every 8 packets, simulating the alternate marking behavior of Section 4.

Note that as opposed to the double marking and single marking methods, hashed sampling is not based on fixed time intervals, as the duration between sampled packets depends only on the hash value.

It is also important to note that all methods that use hash-based marking require the hash function and the set S to be configured consistently across the MPs.

#### 6.1.1. Hashed Pulse Marking

In this approach a hash is computed over the packet content, and both LM and DM are triggered based on the pulse-based trigger (Section 6.1). A pulse is detected when the hash value  $h(c)$  is equal to one of the values in S. The hash function h and the set S determine the probability (or frequency) of the pulse event.

#### 6.1.2. Hashed Step Marking

As in the previous approach, hashed step marking also uses a hash that is computed over the packet content. In this approach DM is performed using a pulse-based trigger, whereas the LM trigger is step-based (Section 6.1). The main drawback of this method is that the step-based trigger is possible only under the assumption that the hash function is monotonically increasing, which is not necessarily possible in all cases. Specifically, a measured flow is not necessarily an IPv4 5-tuple. For example, a measured flow may include multiple IPv4 5-tuple flows, and in this case the Identification field is not monotonically increasing.

### 7. Single Marking Hashed

Mixed hashed marking combines the single marking approach with hash-based sampling. A single marking bit is used in the packet header as a color indicator, while a hash-based pulse is used to trigger DM. Although this method requires a single bit, it is described in this

section as it is closely related to the other hash-based methods that require zero marking bits.

The hash-based selection for DM can be applied in one of two possible approaches: the basic approach, and the dynamic approach. In the basic approach, packets forwarded between two MPs, MP1 and MP2, are selected using a hash function, as described above. One of the challenges is that the frequency of the sampled packets may vary considerably, making it difficult for the management system to correlate samples from the two MPs. Thus, the dynamic approach can be used.

In the dynamic hash-based sampling, alternate marking is used to create divide time into periods, so that hash-based samples are divided into batches, allowing to anchor the selected samples to their period. Moreover, by dynamically adapting the length of the hash value, the number of samples is bounded in each marking period. This can be realized by choosing first the maximum number of samples (NMAX) to be used with the initial hash length. The algorithm starts with only few hash bits, that permit to select a greater percentage of packets (e.g. with 1 bit of hash half of the packets are sampled). When the number of selected packets reaches NMAX, a hashing bit is added. As a consequence, the sampling proceeds at half of the original rate and the packets already selected that do not match the new hash are discarded. This step can be repeated iteratively. It is assumed that each sample includes the timestamp (used for DM) and the hash value, allowing the management system to match the samples received from the two MPs.

The dynamic process statistically converges at the end of a marking period and the number of selected samples beyond the initial NMAX samples mentioned above is between  $NMAX/2$  and NMAX. Therefore, the dynamic approach paces the sampling rate, allowing to bound the number of sampled packets per sampling period.

## 8. Timing and Synchronization Aspects

As pointed out in [RFC8321], it is assumed that all MPs are synchronized to a common reference time with an accuracy of  $\pm L/2$ , where L is the periodic measurement interval. Thus, the difference between the clock values of any two MPs is bounded by L. Note that this is a relatively relaxed synchronization requirement that does not require complex means of synchronization. Clocks can be synchronized for example using NTP [RFC5905], PTP [IEEE1588], or by other means.

In the step-based approaches the common reference time is used for dividing the time domain into equal-sized measurement periods, such

that all packets forwarded during a measurement period have the same color, and consecutive periods have alternating colors. In the pulse-based approaches the synchronization helps the management system to correlate measurements from multiple measurement points without ambiguity.

### 8.1. Synchronization Aspects in Multiplexed Marking

The single marking bit incorporates two multiplexed values. From the monitoring MP's perspective, the two values are Time-Division Multiplexed (TDM), as depicted in Figure 8. It is assumed that the start time of every measurement period is known to both the initiating MP and the monitoring MP. If the measurement period is  $L$ , then during the first and the last  $L/4$  time units of each block the marking bit is interpreted by the monitoring MP as a color indicator. During the middle part of the block, the marking bit is interpreted as a timestamp indicator; if the value of this bit is different than the color value, the corresponding packet is used as a reference for delay measurement.

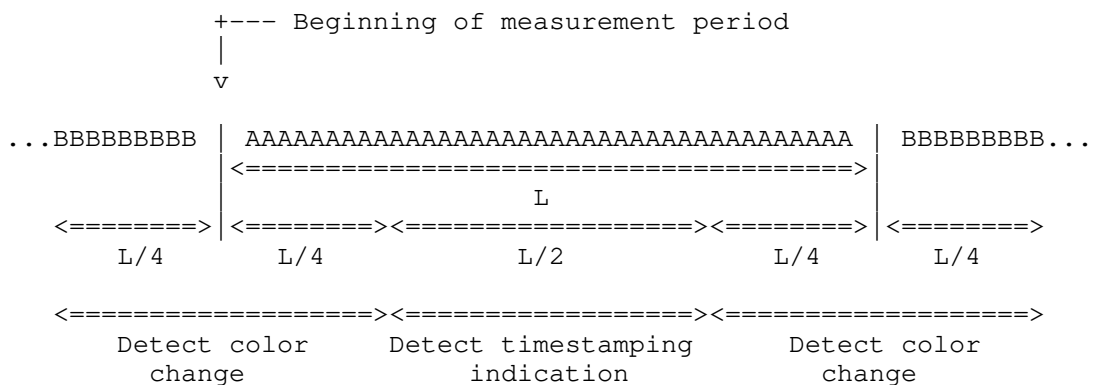


Figure 8: Multiplexed marking field interpretation at the receiving measurement point.

In order to prevent ambiguity in the receiver's interpretation of the marking field, the initiating MP is permitted to set the timestamp indication only during a specific interval, as depicted in Figure 9. Since the receiver is willing to receive the timestamp indication during the middle  $L/2$  time units of the block, the sender refrains from sending the timestamp indication during a guardband interval of  $d$  time units at the beginning and end of the  $L/2$ -period.



performance, for example from MP3 to MP5. Alternate marking in multipoint scenarios is discussed in detail in [I-D.ietf-ippm-multipoint-alt-mark].

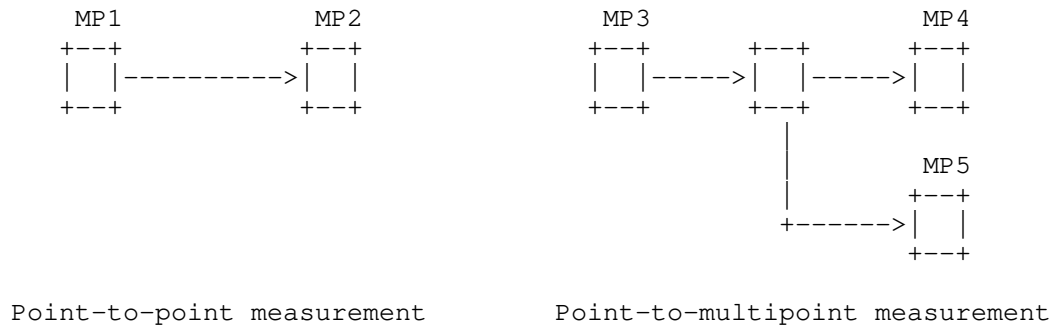


Figure 10: Point-to-point and point-to-multipoint measurements.

## 10. Summary of Marking Methods

This section summarizes the marking methods described in this memo. Each row in the table of Figure 11 represents a marking method. For each method the table specifies the number of bits required in the header, the number of counters per flow for LM, the methods used for LM and DM (pulse or step), and also the resilience to disturbances.

| Method                         | # of bits | # of counters | LM Method                 | DM Method       | Resilience to Reordering |    | Resilience to Packet drops |    |
|--------------------------------|-----------|---------------|---------------------------|-----------------|--------------------------|----|----------------------------|----|
|                                |           |               |                           |                 | LM                       | DM | LM                         | DM |
| Single marking<br>- 1st packet | 1         | 2             | Step                      | Step            | +                        | -- | +                          | -- |
| Single marking<br>- mean delay | 1         | 2             | Step                      | Mean            | +                        | +  | +                          | -  |
| Double marking                 | 2         | 2             | Step                      | Pulse           | +                        | +  | +                          | =  |
| Single marking<br>multiplexed  | 1         | 2             | Step                      | Pulse           | +                        | +  | +                          | =  |
| Pulse marking                  | 1         | 1             | Pulse                     | Pulse           | --                       | +  | -                          | =  |
| Zero marking<br>hashed         | 0         | 1<br>(2)      | Hashed<br>pulse<br>(step) | Hashed<br>pulse | --<br>(-)                | +  | -                          | +  |
| Single marking<br>hashed       | 1         | 2             | Step                      | Hashed<br>pulse | +                        | +  | +                          | +  |

+ Accurate measurement.  
 = Invalidate only if a measured packet is lost (detectable)  
 - No measurement in case of disturbance (detectable).  
 -- False measurement in case of disturbance (not detectable).

Figure 11: Detailed Summary of Marking Methods

In the context of this comparison two possible disturbances are considered: out-of-order delivery, and packet drops. Generally speaking, pulse based methods are sensitive to packet drops, since if the marked packet is dropped no measurement is recorded in the current period. Notably, a missing measurement is detectable by the management system, and is not as severe as a false measurement. Step-based triggers are generally resilient to out-of-order delivery for LM, but are not resilient to out-of-order delivery for DM. Notably, a step-based trigger may yield a false delay measurement when packets are delivered out-of-order, and this inaccuracy is not detectable.

As mentioned above, the double marking method is the most straightforward approach, and is resilient to most of the



disturbances that were analyzed. Its obvious drawback is that it requires two marking bits.

Several single marking methods are discussed in this memo. In this case there is no clear verdict which method is the optimal one. The first packet method may be simple to implement, but may present erroneous delay measurements in case of dropped or reordered packets. Arguably, the mean delay approach and the multiplexed approach may be more difficult to implement (depending on the underlying platform), but are more resilient to the disturbances that were considered here. Note that the computational complexity of the mean delay approach can be reduced by combining it with a hashed approach, i.e., by computing the mean delay over a hash-based subset of the packets. The pulse marking method requires only a single counter per flow, while the other methods require two counters per flow.

The hash-based sampling approaches reduce the overhead to zero bits, which is a significant advantage. However, the sampling period in these approaches is not associated with a fixed time interval. Therefore, in some cases adjacent packets may be selected for the sampling, potentially causing measurement errors. Furthermore, when the traffic rate is low, measurements may become significantly infrequent.

It is clear from the previous table that packet loss measurement can be considered resilient to both reordering and packet drops if at least one bit is used with a step-based approach. Thus, since the packet loss can be considered obvious, the previous table can be simplified into Figure 12, where only the characteristics of delay measurements are highlighted. This more compact table allows room for an additional column referring to multipoint-to-multipoint (Section 9) delay measurement compatibility.

| Marking Method              | # of bits | LM on All Packets | DM Resilience to Reordering | DM Resilience to Packet drops | DM Multipoint compatible |
|-----------------------------|-----------|-------------------|-----------------------------|-------------------------------|--------------------------|
| Single marking - 1st packet | 1         | Yes               | --                          | -                             | No                       |
| Single marking - mean delay | 1         | Yes               | +                           | -                             | Yes                      |
| Double marking              | 2         | Yes               | +                           | =                             | No                       |
| Single marking multiplexed  | 1         | Yes               | +                           | =                             | No                       |
| Pulse marking               | 1         | No                | +                           | =                             | No                       |
| Zero marking hashed         | 0         | No                | +                           | +                             | Yes                      |
| Single marking hashed       | 1         | Yes               | +                           | +                             | Yes                      |

- + Accurate measurement.
- = Invalidate only if a measured packet is lost (detectable)
- No measurement in case of disturbance (detectable).
- False measurement in case of disturbance (not detectable).

Figure 12: Summary of Marking Methods: focus on Delay Measurement

In the context of delay measurement, both zero marking hashed and single marking hashed are resilient to packet drops. Using double marking it could also be possible to perform an accurate measurement in case of packet drops, as long as the packet that is marked for DM is not dropped.

The single marking hashed method seems the most complete approach, especially because it is also compatible with multipoint-to-multipoint measurements.

## 11. Alternate Marking using Reserved Values

As mentioned in Section 1, a marking bit is not necessarily a single bit, but may be implemented by using two well-known values in one of the header fields. Similarly, two-bit marking can be implemented using four reserved values.

A notable example is MPLS Synonymous Flow Labels (SFL), as defined in [I-D.ietf-mpls-rfc6374-sfl]. Two MPLS Label values can be used to indicate the two colors of a given LSP: the original Label value, and an SFL value. A similar approach can be applied to IPv6 using the Flow Label field.

The following example illustrates how alternate marking can be implemented using reserved values. The bit multiplexing approach of Section 5.3 is applicable not only to single-bit color indicators, but also to two-value indicators; instead of using a single bit that is toggled between '0' and '1', two values of the indicator field, U and W, can be used in the same manner, allowing both loss and delay measurement to be performed using only two reserved values. Thus, the multiplexing approach of Figure 6 can be illustrated more generally with two values, U and W, as depicted in Figure 13.

A: packet with color 0

B: packet with color 1

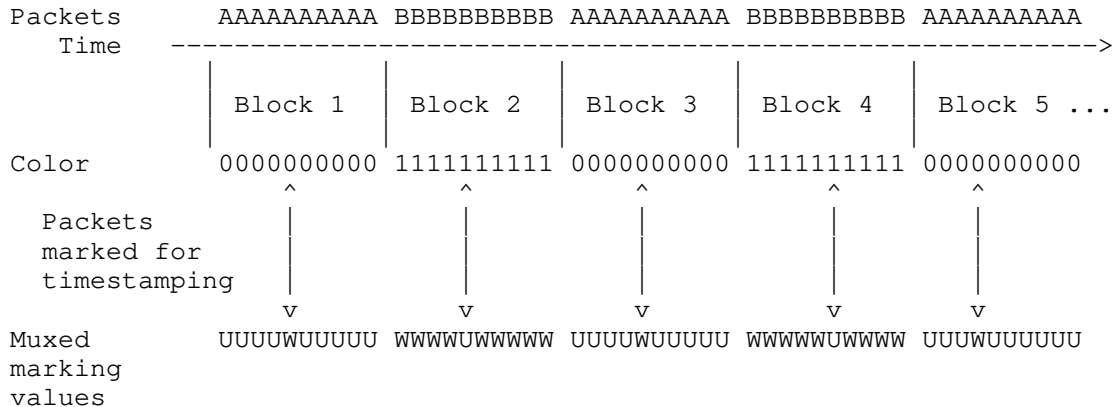


Figure 13: Alternate marking with two multiplexed marking values, U and W.

## 12. IANA Considerations

This memo includes no requests from IANA.

## 13. Security Considerations

The security considerations of the alternate marking method are discussed in [RFC8321]. The analysis of Section 10 emphasizes the sensitivity of some of the alternate marking methods to packet drops and to packet reordering. Thus, a malicious attacker may attempt to tamper with the measurements by either selectively dropping packets, or by selectively reordering specific packets. The multiplexed marking method Section 5.3 that is defined in this document requires slightly more stringent synchronization than the conventional marking method, potentially making the method more vulnerable to attacks on the time synchronization protocol. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

## 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>>.
- [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>>.

### 14.2. Informative 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-02 (work in progress), July 2019.
- [I-D.ietf-mpls-rfc6374-sf1]  
Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S.,  
Mirsky, G., and G. Fioccola, "RFC6374 Synonymous Flow  
Labels", draft-ietf-mpls-rfc6374-sf1-03 (work in  
progress), December 2018.

[I-D.ietf-mpls-sfl-framework]

Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S., and G. Mirsky, "Synonymous Flow Label Framework", draft-ietf-mpls-sfl-framework-04 (work in progress), December 2018.

[IEEE1588]

IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.

[RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, DOI 10.17487/RFC5475, March 2009, <<https://www.rfc-editor.org/info/rfc5475>>.

[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>>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.

#### Authors' Addresses

Tal Mizrahi  
Huawei Network.IO Innovation Lab  
Israel

Email: [tal.mizrahi.phd@gmail.com](mailto:tal.mizrahi.phd@gmail.com)

Carmi Arad

Email: [carmi.arad@gmail.com](mailto:carmi.arad@gmail.com)

Giuseppe Fioccola  
Huawei Technologies

Email: [giuseppe.fioccola@huawei.com](mailto:giuseppe.fioccola@huawei.com)

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

Email: mauro.cociglio@telecomitalia.it

Mach(Guoyi) Chen  
Huawei Technologies

Email: mach.chen@huawei.com

Lianshu Zheng  
Huawei Technologies

Email: vero.zheng@huawei.com

Greg Mirsky  
ZTE Corp.

Email: gregimirsky@gmail.com

OPSAWG  
Internet-Draft  
Intended status: Informational  
Expires: 26 August 2022

H. Song  
Futurewei  
F. Qin  
China Mobile  
H. Chen  
China Telecom  
J. Jin  
LG U+  
J. Shin  
SK Telecom  
22 February 2022

A Framework for In-situ Flow Information Telemetry  
draft-song-opsawg-ifit-framework-17

Abstract

As network scale increases and network operation becomes more sophisticated, existing Operation, Administration, and Maintenance (OAM) methods are no longer sufficient to meet the monitoring and measurement requirements. Emerging data-plane on-path telemetry techniques which provide high-precision flow insight and which issue notifications in real time can supplement existing proactive and reactive methods that run in active and passive modes. These new approaches are collectively known as in-situ flow information telemetry (IFIT). They enable quality of experience for users and applications, and identification of network faults and deficiencies.

This document outlines a high-level framework for IFIT to collect and correlate performance measurement information from the network. It identifies the components that coordinate existing protocol tools and telemetry mechanisms, and addresses deployment challenges for flow-oriented on-path telemetry techniques, especially in carrier networks.

The document is a guide for system designers applying the referenced techniques. It is also intended to motivate further work to enhance the OAM ecosystem.

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 26 August 2022.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

|                                                                    |    |
|--------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                          | 3  |
| 1.1. Classification and Modes of On-path Telemetry . . . . .       | 4  |
| 1.2. Requirements and Challenges . . . . .                         | 6  |
| 1.3. Scope . . . . .                                               | 8  |
| 1.4. Relationship with Network Telemetry Framework (NTF) . . . . . | 8  |
| 1.5. Glossary . . . . .                                            | 9  |
| 2. Architectural Concepts and Key Components . . . . .             | 9  |
| 2.1. Reference Deployment . . . . .                                | 9  |
| 2.2. Key Components . . . . .                                      | 11 |
| 2.2.1. Flexible Flow, Packet, and Data Selection . . . . .         | 11 |
| 2.2.2. Flexible Data Export . . . . .                              | 13 |
| 2.2.3. Dynamic Network Probe . . . . .                             | 15 |
| 2.2.4. On-demand Technique Selection and Integration . . . . .     | 17 |
| 2.3. IFIT for Reflective Telemetry . . . . .                       | 18 |
| 2.3.1. Intelligent Multipoint Performance Monitoring . . . . .     | 19 |
| 2.3.2. Intent-based Network Monitoring . . . . .                   | 19 |
| 3. Guidance for Solution Developers . . . . .                      | 20 |
| 3.1. Encapsulation in Transport Protocols . . . . .                | 20 |
| 3.2. Tunneling Support . . . . .                                   | 21 |
| 3.3. Deployment Automation . . . . .                               | 21 |



|                                       |    |
|---------------------------------------|----|
| 4. Security Considerations . . . . .  | 22 |
| 5. IANA Considerations . . . . .      | 22 |
| 6. Contributors . . . . .             | 22 |
| 7. Acknowledgments . . . . .          | 22 |
| 8. References . . . . .               | 22 |
| 8.1. Normative References . . . . .   | 22 |
| 8.2. Informative References . . . . . | 23 |
| Authors' Addresses . . . . .          | 27 |

## 1. Introduction

Efficient network operation increasingly relies on high-quality data-plane telemetry to provide the necessary visibility into the behavior of traffic flows and network resources. Existing Operation, Administration, and Maintenance (OAM) methods, which include proactive and reactive techniques, running both active and passive modes, are no longer sufficient to meet the monitoring and measurement requirements when networks becomes more autonomous [RFC8993] and application-aware [I-D.li-apn-framework]. The complexity of today's networks and service quality requirements demand new high-precision and real-time OAM techniques.

The ability to expedite network failure detection, fault localization, and recovery mechanisms, particularly in the case of soft failures or path degradation is expected, and it must not cause service disruption. Emerging on-path telemetry techniques can provide high-precision flow insight and real-time network issue notification (e.g., jitter, latency, packet loss, significant bit error variations, and unequal load-balancing). On-Path Telemetry (OPT) refers to data-plane telemetry techniques that directly tap and measure network traffic by embedding instructions or metadata into user packets. The data provided by on-path telemetry are especially useful for verifying Service Level Agreement (SLA) compliance, user experience enhancement, service path enforcement, fault diagnosis, and network resource optimization. It is essential to recognize that existing work on this topic includes a variety of on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX) [I-D.ietf-ippm-ioam-direct-export], Marking-based Postcard-based Telemetry (PBT-M) [I-D.song-ippm-postcard-based-telemetry], Enhanced Alternate Marking (EAM) [I-D.zhou-ippm-enhanced-alternate-marking], and Hybrid Two-Step (HTS) [I-D.mirsky-ippm-hybrid-two-step], have been developed or proposed. These techniques can provide flow information on the entire forwarding path on a per-packet basis in real-time. The aforementioned on-path telemetry techniques differ from the active and passive OAM schemes in that they directly modify and monitor the user packets in networks so as to achieve high measurement accuracy. Formally, these on-path telemetry techniques can be classified as the

OAM hybrid type I, since they involve "augmentation or modification of the stream of interest, or employment of methods that modify the treatment of the streams", according to [RFC7799]. We name these techniques as "In-situ Flow Information Telemetry" (IFIT).

On-path telemetry is useful for application-aware networking operations, not only in data center and enterprise networks, but also in carrier networks which may cross multiple domains. The techniques can provide benefits for carrier network operators in various scenarios. For example, it is critical for the operators who offer high-bandwidth, latency and loss-sensitive services such as video streaming and online gaming to closely monitor the relevant flows in real-time as the basis for any further optimizations.

This framework document is intended to guide system designers attempting to use the referenced techniques as well as to motivate further work to enhance the telemetry ecosystem. It highlights requirements and challenges, outlines important techniques that are applicable, and provides examples of how these might be applied for critical use cases.

The document scope is discussed in Section 1.3.

#### 1.1. Classification and Modes of On-path Telemetry

The operation of IFIT differs from both active OAM and passive OAM as defined in [RFC7799]. It does not generate any active probe packets or passively observe unmodified user packets. Instead, it modifies selected user packets in order to collect useful information about them. Therefore, the operation is categorized as the hybrid OAM type I method per [RFC7799].

This hybrid OAM type I method can be further partitioned into two modes [passport-postcard]. In the passport mode, each node on the path can add telemetry data to the user packets (i.e., stamps the passport). The accumulated data trace is exported at a configured end node. In the postcard mode, each node directly exports the telemetry data using an independent packet (i.e., sends a postcard) while the user packets are unmodified. It is possible to combine the two modes together in one solution. We call this the hybrid mode.

Figure 1 shows the classification of the on-path telemetry techniques.

| Mode      | Passport               | Postcard                 | Hybrid                     |
|-----------|------------------------|--------------------------|----------------------------|
| Technique | IOAM Trace<br>IOAM E2E | IOAM DEX<br>PBT-M<br>EAM | Multicast Telemetry<br>HTS |

Figure 1: On-path Telemetry Technique Classification

IOAM Trace and E2E options are described in [I-D.ietf-ippm-ioam-data].

EAM is described in [I-D.zhou-ippm-enhanced-alternate-marking].

IOAM DEX option is described in [I-D.ietf-ippm-ioam-direct-export].

PBT-M is described in [I-D.song-ippm-postcard-based-telemetry].

Multicast Telemetry is described in [I-D.ietf-mboned-multicast-telemetry].

HTS is described in [I-D.mirsky-ippm-hybrid-two-step].

The advantages of the passport mode include:

- \* It automatically retains the telemetry data correlation along the entire path. The self-describing feature simplifies the data consumption.
- \* The on-path data for a packet is only exported once so the data export overhead is low.
- \* Only the head and tail nodes of the paths need to be configured for header insertion and removal, so the configuration overhead is low.

The disadvantages of the passport mode include:

- \* The telemetry data carried by user packets inflate the packet size, which may be undesirable or prohibitive.
- \* Approaches for encapsulating the instruction header and data in transport protocols need to be standardized.
- \* Carrying sensitive data along the path is vulnerable to security and privacy breach.

- \* If a packet is dropped on the path, the data collected are also lost.

The postcard mode complements the passport mode. The advantages of the postcard mode include:

- \* Either there is no packet header overhead (e.g., PBT-M) or the overhead is small and fixed (e.g., IOAM DEX).
- \* The encapsulation requirement may be avoided (e.g., PBT-M).
- \* The telemetry data can be secured before export.
- \* Even if a packet is dropped on the path, the partial data collected are still available.

The disadvantages of the postcard mode include:

- \* Telemetry data are spread in multiple postcards so extra effort is needed to correlate the data.
- \* Every node exports a postcard for a packet which increases the data export overhead.
- \* In case of PBT-M, every node on the path needs to be configured, so the configuration overhead is high.
- \* In case of IOAM DEX, the transport encapsulation requirement remains.

The hybrid mode either tailors for some specific application scenario (e.g., Multicast Telemetry) or provides some alternative approach (e.g., HTS). A postcard can be sent per segment of a path or the telemetry data can be carried in a companion packet following each monitored use packet. The hybrid mode combines the advantages of both the passport mode and the postcard mode, but it may incur extra processing complexity.

## 1.2. Requirements and Challenges

Although on-path telemetry is beneficial, successfully applying such techniques in carrier networks must consider performance, deployability, and flexibility. Specifically, we need to address the following practical deployment challenges:

- \* C1: On-path telemetry incurs extra packet processing which may cause stress on the network data plane. The potential impact on the forwarding performance creates an unfavorable "observer

effect" (where the actions of performing on-path telemetry may change the behavior of the traffic being measured). This will not only damage the fidelity of the measurement, but also defy the purpose of the measurement.

- \* C2: On-path telemetry can generate a considerable amount of data which may claim too much transport bandwidth and inundate the servers for data collection, storage, and analysis. For example, if the technique is applied to all the traffic, one node may collect a few tens of bytes as telemetry data for each packet. The whole forwarding path might accumulate telemetry data with a size similar to or even exceeding that of the original packet.
- \* C3: The collectible data defined currently are essential but limited. This, in turn, limits the management and operational techniques that can be applied. Flexibility and extensibility of data definition, aggregation, acquisition, and filtering, must be considered.
- \* C4: Applying only a single underlying on-path telemetry technique may miss some important events or lead to incorrect results. For example, packet drop can cause the loss of the flow telemetry data and the packet drop location and reason remains unknown if only the In-situ OAM trace option is used. A comprehensive solution needs the flexibility to switch between different underlying techniques and adjust the configurations and parameters at runtime. Thus, system-level orchestration is needed.
- \* C5: We must provide solutions to support an incremental deployment strategy. That is, we need to support established encapsulation schemes for various predominant protocols such as Ethernet, IPv6, and MPLS with backward compatibility and properly handle various transport tunnels.
- \* C6: The development of simplified on-path telemetry primitives and models for configuration and queries is essential. Telemetry models may be utilized via an API-based telemetry service for external applications, for end-to-end performance measurement and application performance monitoring. Standard-based protocols and methods are needed for network configuration and programming, and telemetry data pre-processing and export, to provide interoperability.

### 1.3. Scope

Following the network telemetry framework discussed in [I-D.ietf-opsawg-ntf], this document focuses on the on-path telemetry, a specific class of data-plane telemetry techniques, and provides a high-level framework which addresses the challenges for deployment listed in Section 1.2, especially in carrier networks.

This document aims to clarify the problem space, essential requirements, and summarizes best practices and general system design considerations. This document provides some examples to show the novel network telemetry applications under the framework.

As an informational document, it describes an open framework with a few key components. The framework does not enforce any specific implementation on each component, neither does it define interfaces (e.g., API, protocol) between components. The choice of underlying on-path telemetry techniques and other implementation details is determined by the application implementer. Therefore, the framework is not a solution specification. It only provides a high-level overview and is not necessarily a mandatory recommendation for on-path telemetry applications.

The standardization of the underlying techniques and interfaces mentioned in this document is undertaken by various working groups. Due to the limited scope and intended status of this document, it has no overlap or conflict with those works.

### 1.4. Relationship with Network Telemetry Framework (NTF)

[I-D.ietf-opsawg-ntf] describes a Network Telemetry Framework (NTF). One dimension used by NTF to partition network telemetry techniques and systems is based on the three planes in networks (i.e., control plane, management plane, and forwarding plane) and external data sources. IFIT fits in the category of forwarding-plane telemetry and deals with the specific on-path technical branch of the forwarding-plane telemetry.

According to NTF, an on-path telemetry application mainly subscribes to event-triggered or streaming data. The key functional components of IFIT described in Section 2.2 match the general components in NTF with more specific functions. "On-demand Technique Selection and Integration" is an application layer function, matching the "Data Query, Analysis, and Storage" component in NTF; "Flexible Flow, Packet, and Data Selection" matches the "Data Configuration and Subscription" component; "Flexible Data Export" matches the "Data Encoding and Export" component; "Dynamic Network Probe" matches the "Data Generation and Processing" component.

## 1.5. Glossary

This section defines and explains the acronyms and terms used in this document.

**On-path Telemetry:** Remotely acquiring performance and behavior data about network flows on a per-packet basis on the packet's forwarding path. The term refers to a class of data-plane telemetry techniques, including IOAM, PBT, EAM, and HTS. Such techniques may need to mark user packets, or insert instruction/metadata into the headers of user packets.

**IFIT:** In-situ Flow Information Telemetry is a high-level reference framework that shows how network data-plane monitoring and measurement applications can address the deployment challenges of the flow-oriented on-path telemetry techniques.

**Reflective Telemetry:** The reflective telemetry functions in a dynamic and closed-loop fashion. A new telemetry action is provisioned as a result of self-knowledge acquired through prior telemetry actions.

## 2. Architectural Concepts and Key Components

To address the challenges mentioned in Section 1.2, a high-level framework which can help to build a workable and efficient on-path telemetry application is presented. In-situ Flow Information Telemetry (IFIT) is dedicated to on-path telemetry data about user and application traffic flows. It covers a class of on-path telemetry techniques and works at a level higher than any specific underlying technique. The framework is comprised of some key functional components (Section 2.2). By assembling these components, IFIT supports reflective telemetry that enables autonomous network operations (Section 2.3).

### 2.1. Reference Deployment

Figure 2 shows a reference deployment scenario of on-path telemetry.

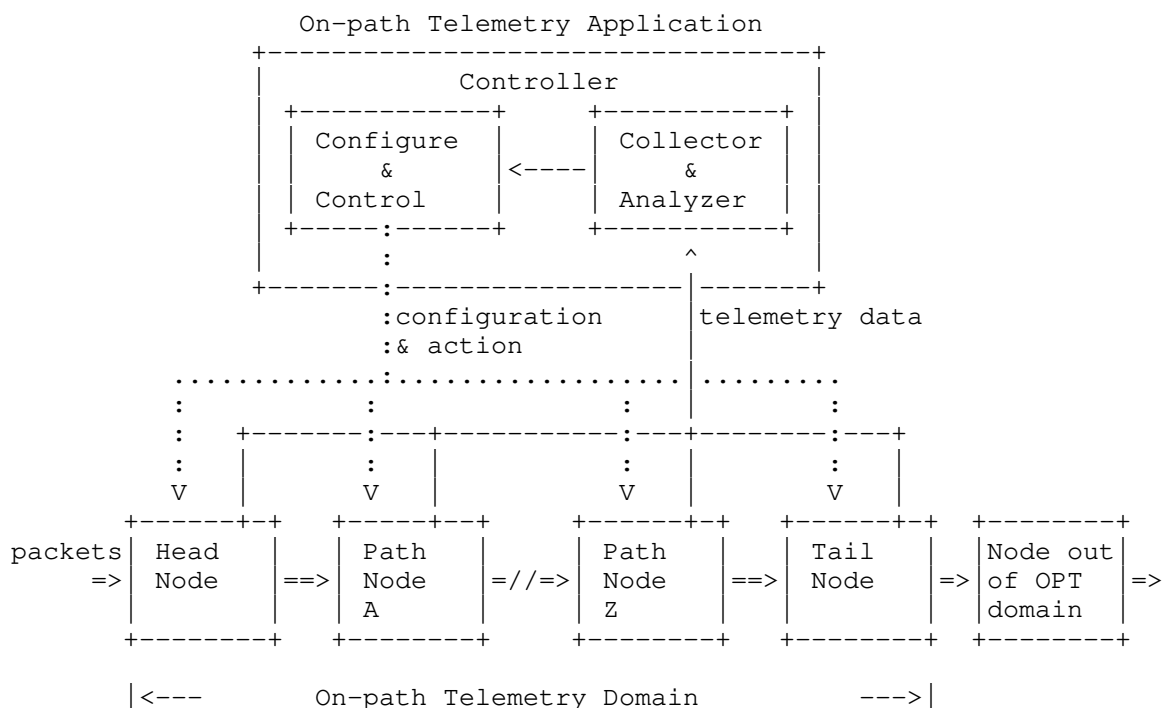


Figure 2: Deployment Scenario

An on-path telemetry application can conduct network data-plane monitoring and measurement tasks over a limited domain [RFC8799] by applying one or more underlying techniques. The application contains multiple elements, including configuring the network nodes and processing the telemetry data. The application usually uses a logically centralized controller for configuring the network nodes in the domain, and collecting and analyzing telemetry data. The configuration determines which underlying technique is used, what telemetry data are of interest, which flows and packets are concerned with, how the telemetry data are collected, etc. The process can be dynamic and interactive: after the telemetry data processing and analyzing, the application may instruct the controller to modify the configuration of the nodes, which affects the future telemetry data collection.

From the system-level view, it is recommended to use standardized configuration and data collection interfaces, regardless of the underlying technique. The specification of these interfaces and the implementation of the controller are out of scope for this document.



The on-path telemetry domain encompasses the head nodes and the end nodes, and may cross multiple network domains. The head nodes are responsible for enabling the on-path telemetry functions and the end nodes are responsible for terminating them. All capable nodes in this domain will be capable of executing the instructed on-path telemetry function. It is important to note that any application must, through configuration and policy, guarantee that any packet with on-path telemetry header and metadata will not leak out of the domain.

The underlying on-path telemetry techniques covered by the IFIT framework can be of any modes discussed in Section 1.1.

## 2.2. Key Components

The key components of IFIT to address the challenges listed in Section 1.2 are as follows. The components are described in more detail in the sections that follow.

- \* Flexible flow, packet, and data selection policy, addressing the challenge C1 described in Section 1;
- \* Flexible data export, addressing the challenge C2;
- \* Dynamic network probe, addressing C3;
- \* On-demand technique selection and integration, addressing C4.

Note that the challenges C5 and C6 are mostly standard-related, and are fundamental to IFIT. We discuss the protocol implications and guidance for solution developers in Section 3.

### 2.2.1. Flexible Flow, Packet, and Data Selection

In most cases, it is impractical to enable data collection for all the flows and for all the packets in a flow due to the potential performance and bandwidth impact. Therefore, a workable solution usually need to select only a subset of flows and flow packets on which to enable data collection, even though this means the loss of some information and accuracy.

In the data plane, a flow filter like those used for an Access Control List (ACL) provides an ideal means to determine the subset of flows. An application can set a sample rate or probability to a flow to allow only a subset of flow packets to be monitored, collect a different set of data for different packets, and disable or enable data collection on any specific network node. An application can further allow any node to accept or deny the data collection process in full or partially.

Based on these flexible mechanisms, IFIT allows applications to apply flexible flow and data selection policies to suit their requirements. The applications can dynamically change the policies at any time based on the network load, processing capability, focus of interest, and any other criteria.

#### 2.2.1.1. Block Diagram

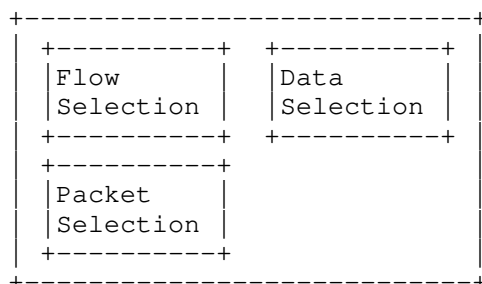


Figure 3: Flexible Flow, Packet, and Data Selection

Figure 3 shows the block diagram of this component. The flow selection block defines the policies to choose target flows for monitoring. Flow has different granularity. A basic flow is defined by 5-tuple IP header fields. Flow can also be aggregated at interface level, tunnel level, protocol level, and so on. The packet selection block defines the policies to choose packets from a target flow. The policy can be either a sampling interval, a sampling probability, or some specific packet signature. The data selection block defines the set of data to be collected. This can be changed on a per-packet or per-flow basis.

#### 2.2.1.2. Example: Sketch-guided Elephant Flow Selection

Network operators are usually more interested in elephant flows which consume more resource and are sensitive to changes in network conditions. A CountMin Sketch [CMSketch] can be used on the data path of the head nodes, which identifies and reports the elephant flows periodically. The controller maintains a current set of elephant flows and dynamically enables the on-path telemetry for only these flows.

#### 2.2.1.3. Example: Adaptive Packet Sampling

Applying on-path telemetry on all packets of the selected flows can still be out of reach. A sample rate should be set for these flows and telemetry should only be enabled on the sampled packets. However, the head nodes have no clue on the proper sampling rate. An overly high rate would exhaust the network resource and even cause packet drops; An overly low rate, on the contrary, would result in the loss of information and inaccuracy of measurements.

An adaptive approach can be used based on the network conditions to dynamically adjust the sampling rate. Every node gives user traffic forwarding higher priority than telemetry data export. In case of network congestion, the telemetry can sense some signals from the data collected (e.g., deep buffer size, long delay, packet drop, and data loss). The controller may use these signals to adjust the packet sampling rate. In each adjustment period (i.e., RTT of the feedback loop), the sampling rate is either decreased or increased in response of the signals. An Additive Increase/Multiplicative Decrease (AIMD) policy similar to the TCP flow control mechanism for rate adjustment can be used.

#### 2.2.2. Flexible Data Export

The flow telemetry data can catch the dynamics of the network and the interactions between user traffic and network. Nevertheless, the data may contain redundancy. It is advisable to remove the redundancy from the data in order to reduce the data transport bandwidth and server processing load.

In addition to efficient export data encoding (e.g., IPFIX [RFC7011] or protobuf (<https://developers.google.com/protocol-buffers/>)), nodes have several other ways to reduce the export data by taking advantage of network device's capability and programmability. Nodes can cache the data and send the accumulated data in batches if the data is not time sensitive. Various deduplication and compression techniques can be applied on the batched data.

From the application perspective, an application may only be interested in some special events which can be derived from the telemetry data. For example, in the case that the forwarding delay of a packet exceeds a threshold, or a flow changes its forwarding path is of interest, it is unnecessary to send the original raw data to the data collecting and processing servers. Rather, IFIT takes advantage of the in-network computing capability of network devices to process the raw data and only push the event notifications to the subscribing applications.

Such events can be expressed as policies. A policy can request data export only on change, on exception, on timeout, or on threshold.

#### 2.2.2.1. Block Diagram

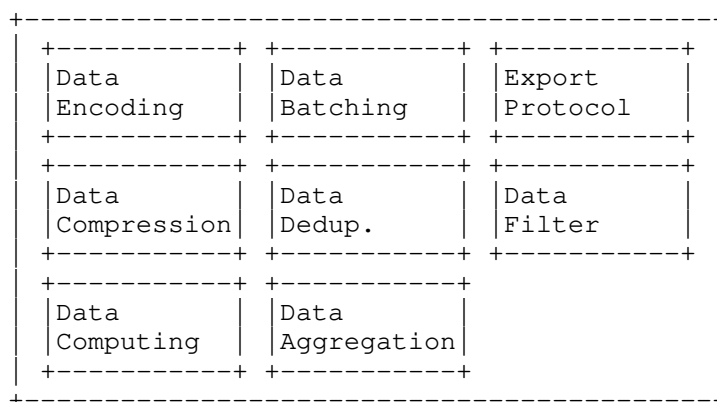


Figure 4: Flexible Data Export

Figure 4 shows the block diagram of this component. The data encoding block defines the method to encode the telemetry data. The data batching block defines the size of batch data buffered at the device side before export. The export protocol block defines the protocol used for telemetry data export. The data compression block defines the algorithm to compress the raw data. The data deduplication block defines the algorithm to remove the redundancy in the raw data. The data filter block defines the policies to filter the needed data. The data computing block defines the policies to preprocess the raw data and generate some new data. The data aggregation block defines the procedure to combine and synthesize the data.

#### 2.2.2.2. Example: Event-based Anomaly Monitor

Network operators are interested in anomalies such as path change, network congestion, and packet drop. Such anomalies are hidden in raw telemetry data (e.g., path trace, timestamp). Such anomalies can be described as events and programmed into the device data plane. Only the triggered events are exported. For example, if a new flow appears at any node, a path change event is triggered; if the packet delay exceeds a predefined threshold in a node, the congestion event is triggered; if a packet is dropped due to buffer overflow, a packet drop event is triggered.

The export data reduction due to such optimization is substantial. For example, given a single 5-hop 10Gbps path, assume a moderate number of 1 million packets per second are monitored, and the telemetry data plus the export packet overhead consume less than 30 bytes per hop. Without such optimization, the bandwidth consumed by the telemetry data can easily exceed 1Gbps (more than 10% of the path bandwidth). When the optimization is used, the bandwidth consumed by the telemetry data is negligible. Moreover, the pre-processed telemetry data greatly simplify the work of data analyzers.

#### 2.2.3. Dynamic Network Probe

Due to limited data plane resource and network bandwidth, it is unlikely one can monitor all the data all the time. On the other hand, the data needed by applications may be arbitrary but ephemeral. It is critical to meet the dynamic data requirements with limited resource.

Fortunately, data plane programmability allows new data probes to be dynamically loaded. These on-demand probes are called Dynamic Network Probes (DNP). DNP is the technique to enable probes for customized data collection in different network planes. When working with an on-path telemetry technique, DNP is loaded into the data plane through incremental programming or configuration. The DNP can effectively conduct data generation, processing, and aggregation.

DNP introduces flexibility and extensibility to IFIT. It can implement the optimizations for export data reduction motioned in the previous section. It can also generate custom data as required by today's and tomorrow's applications.

##### 2.2.3.1. Block Diagram

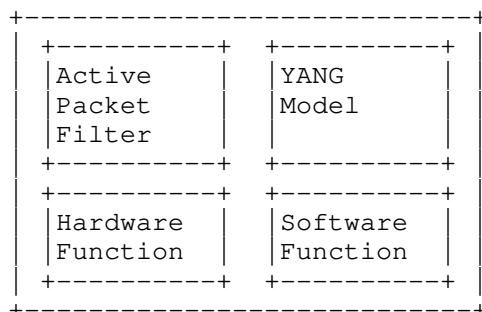


Figure 5: Dynamic Network Probes

Figure 5 shows the block diagram of this component. The active packet filter block is available in most hardware and it defines DNP through dynamically update the packet filtering policies (including flow selection and action). YANG models can be dynamically deployed to enable different data processing and filtering functions. Some hardware allows dynamically loading hardware-based functions into the forwarding path at runtime through mechanisms such as reserved pipelines and function stubs. Dynamically loadable software functions can be implemented in the control processors in capable nodes.

#### 2.2.3.2. Examples

Following are some possible DNP that can be dynamically deployed to support applications.

**On-demand Flow Sketch:** A flow sketch is a compact online data structure (usually a variation of multi-hashing table) for approximate estimation of multiple flow properties. It can be used to facilitate flow selection. The aforementioned CountMin Sketch [CMSketch] is such an example. Since a sketch consumes data plane resources, it should only be deployed when actually needed.

**Smart Flow Filter:** The policies that choose flows and packet sampling rate can change during the lifetime of an application.

**Smart Statistics:** An application may need to count flows based on different flow granularity or maintain hit counters for selected flow table entries.

**Smart Data Reduction:** DNP can be used to program the events that conditionally trigger data export.

#### 2.2.4. On-demand Technique Selection and Integration

With multiple underlying data collection and export techniques at its disposal, IFIT can flexibly adapt to different network conditions and different application requirements.

For example, depending on the types of data that are of interest, IFIT may choose either passport or postcard mode to collect the data; if an application needs to track down where the packets are lost, switching from passport mode to postcard mode should be supported.

IFIT can further integrate multiple data plane monitoring and measurement techniques together and present a comprehensive data plane telemetry solution.

Based on the application requirements and the real-time telemetry data analysis results, new configurations and actions can be deployed.

##### 2.2.4.1. Block Diagram

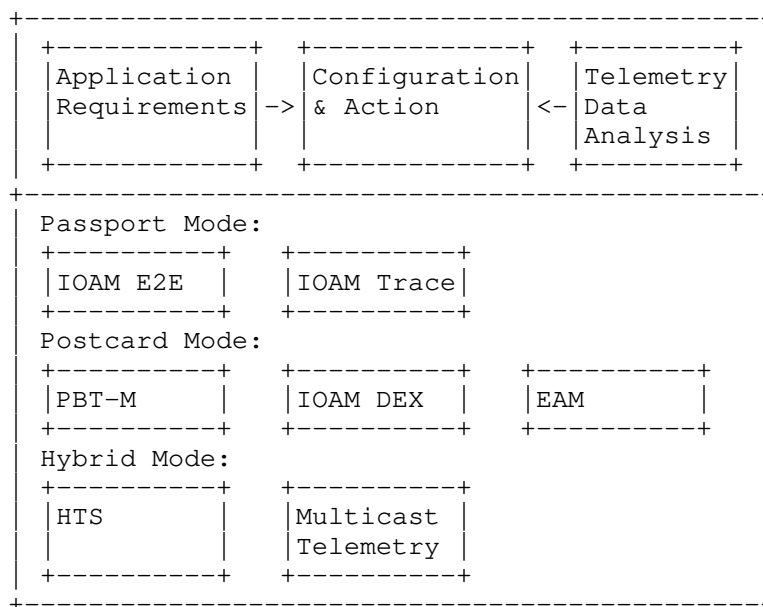


Figure 6: Technique Selection and Integration

Figure 6 shows the block diagram of this component, which lists the candidate on-path telemetry techniques.

Located in the logically centralized controller, this component makes all the control and configuration dynamically to the capable nodes in the domain which will affect the future telemetry data. The configuration and action decisions are based on the inputs from the application requirements and the realtime telemetry data analysis results. Note that here the telemetry data source is not limited to the data plane. The data can come form all the sources mentioned in [I-D.ietf-opsawg-ntf], including external data sources.

### 2.3. IFIT for Reflective Telemetry

The components described in Section 2.2 can work together to support reflective telemetry, as shown in Figure 7.

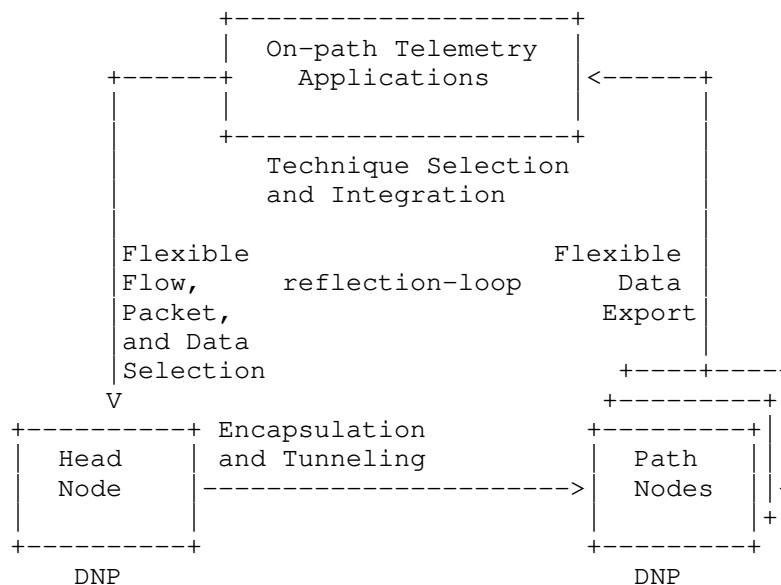


Figure 7: IFIT-based Reflective Telemetry

An application may pick a suite of telemetry techniques based on its requirements and apply an initial technique to the data plane. It then configures the head nodes to decide the initial target flows/packets and telemetry data set, the encapsulation and tunneling scheme based on the underlying network architecture, and the IFIT-capable nodes to decide the initial telemetry data export policy. Based on the network condition and the analysis results of the telemetry data, the application can change the telemetry technique, the flow/data selection policy, and the data export approach in real time without breaking the normal network operation. Many of such dynamic changes can be done through loading and unloading DNP.



The reflective telemetry enabled by the IFIT allows numerous new applications. Two examples are provided below.

### 2.3.1. Intelligent Multipoint Performance Monitoring

[RFC8889] describes an intelligent performance management based on the network condition. The idea is to split the monitoring network into clusters. The cluster partition that can be applied to every type of network graph and the possibility to combine clusters at different levels enable the so-called Network Zooming. It allows a controller to calibrate the network telemetry, so that it can start without examining in depth and monitor the network as a whole. In case of necessity (packet loss or too high delay), an immediate detailed analysis can be reconfigured. In particular, the controller, that is aware of the network topology, can set up the most suitable cluster partition by changing the traffic filter or activate new measurement points and the problem can be localized with a step-by-step process.

An application on top of the controllers can manage such mechanism, whose dynamic and reflective operations are supported by the IFIT framework.

### 2.3.2. Intent-based Network Monitoring

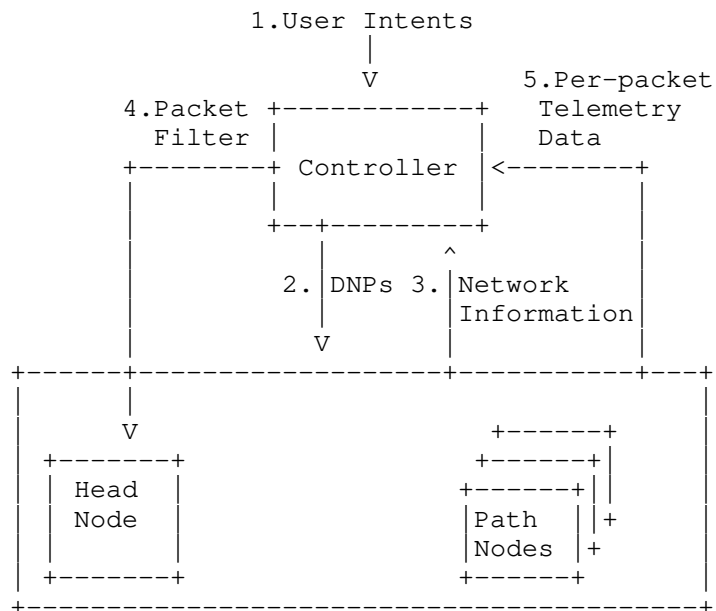


Figure 8: Intent-based Monitoring

In this example, a user can express high level intents for network monitoring. The controller translates an intent and configures the corresponding DNPs in capable nodes which collect necessary network information. Based on the real-time information feedback, the controller runs a local algorithm to determine the suspicious flows. It then deploys specific packet filters to the head node to initiate the high precision per-packet on-path telemetry for these flows.

### 3. Guidance for Solution Developers

Having a high-level framework covering a class of related techniques promotes a holistic approach for standard development and helps to avoid duplicated efforts and piecemeal solutions that only focus on a specific technique while omitting the compatibility and extensibility issues, which is important to a healthy ecosystem for network telemetry.

A complete IFIT-based solution needs standard interfaces for configuration and data extraction, and standard encapsulation on various transport protocols. It may also need standard API and primitives for application programming and deployment. [I-D.ietf-ippm-ioam-deployment] summarizes some techniques for encapsulation and data export for IOAM. Solution developers need to consider the aspects set out in the following subsections.

#### 3.1. Encapsulation in Transport Protocols

Since the introduction of IOAM, the IOAM option header encapsulation schemes in various network protocols have been defined (e.g., [I-D.ietf-ippm-ioam-ipv6-options]). Similar encapsulation schemes are needed to cover the other on-path telemetry techniques. Meanwhile, the on-path telemetry header/data encapsulation schemes in some popular protocols, such as MPLS and SRv6, are also needed. PBT-M [I-D.song-ippm-postcard-based-telemetry] does not introduce new headers to the packets so the trouble of encapsulation for a new header is avoided. While there are some proposals which allow new header encapsulation in MPLS packets (e.g., [I-D.song-mpls-extension-header]) or in SRv6 packets (e.g., [I-D.song-spring-siam]), they are still in their infancy stage and require further work. Before standards are available, in a confined domain, pre-standard encapsulation approaches may be applied.

### 3.2. Tunneling Support

In carrier networks, it is common for user traffic to traverse various tunnels for QoS, traffic engineering, or security. Both the uniform mode and the pipe mode for tunnel support are required and described in [I-D.song-ippm-ioam-tunnel-mode]. The uniform mode treats the nodes in a tunnel uniformly as the nodes outside of the tunnel on a path. In contrast, the pipe mode abstracts all the nodes between the tunnel ingress and egress as a circuit so no nodes in the tunnel is visible to the nodes outside of the tunnel. With such flexibility, the operator can either gain a true end-to-end visibility or apply a hierarchical approach which isolates the monitoring domain between customer and provider.

### 3.3. Deployment Automation

Standard approaches that automate the function configuration, and capability query and advertisement, could either be deployed in a centralized fashion or a distributed fashion. The draft [I-D.ietf-ippm-ioam-yang] provides a YANG model for IOAM configuration. Similar models needs to be defined for other techniques. It is also helpful to provide standards-based approaches for configuration in various network environments. For example, in Segment Routing (SR) networks, extensions to BGP or Path Computation Element Communication Protocol (PCEP) can be defined to distribute SR policies carrying on-path telemetry information, so that telemetry behavior can be enabled automatically when the SR policy is applied. [I-D.chen-pce-sr-policy-ifit] defines extensions to PCEP to configure SR policies for on-path telemetry. [I-D.ietf-idr-sr-policy-ifit] defines extensions to BGP for the same purpose. Additional capability discovery and dissemination will be needed for other types of networks.

To realize the potential of on-path telemetry, programming and deploying DNPs are important. ForCES [RFC5810] is a standard protocol for network device programming, which can be used for DNP deployment. Currently some related works such as [I-D.www-netmod-event-yang] and [I-D.bwd-netmod-eca-framework] have proposed to use YANG models to define the smart policies which can be used to implement DNPs. In the future, other approaches for hardware and software-based functions can be development to enhance the programmability and flexibility.

#### 4. Security Considerations

In addition to the specific security issues discussed in each individual document on on-path telemetry, this document considers the overall security issues at the system level. This should serve as a guide to the on-path telemetry application developers and users. General security and privacy considerations for any network telemetry system are also discussed in [I-D.ietf-opsawg-ntf].

Since the on-path telemetry techniques work on the network forwarding plane, the IFIT framework poses some security risks. The important and sensitive information about a network could be exposed to an attacker. Further, the on-path telemetry data might swamp various parts of the network, leading to a possible DoS attack.

Fortunately, security measures can be enforced on various parts of the framework to mitigate such threats. For example, the configuration can filter and rate limit the monitored traffic; encryption and authentication can be applied on the exported telemetry data; different underlying techniques can be chosen to adapt to the different network conditions.

#### 5. IANA Considerations

This document includes no request to IANA.

#### 6. Contributors

Other major contributors of this document include Giuseppe Fioccola, Daniel King, Zhenqiang Li, Zhenbin Li, Tianran Zhou, and James Guichard.

#### 7. Acknowledgments

We thank Diego Lopez, Shwetha Bhandari, Joe Clarke, Adrian Farrel, Frank Brockners, Al Morton, Alex Clemm, Alan DeKok, Benoit Claise, and Warren Kumari for their constructive suggestions for improving this document.

#### 8. References

##### 8.1. 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>>.

- [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>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.

## 8.2. Informative References

- [CMSketch] Cormode, G. and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications", 2005, <<http://dx.doi.org/10.1016/j.jalgor.2003.12.001>>.
- [I-D.bwd-netmod-eca-framework]  
Boucadair, M., Wu, Q., Wang, M., King, D., and C. Xie, "Framework for Use of ECA (Event Condition Action) in Network Self Management", Work in Progress, Internet-Draft, draft-bwd-netmod-eca-framework-00, 3 November 2019, <<https://www.ietf.org/archive/id/draft-bwd-netmod-eca-framework-00.txt>>.
- [I-D.chen-pce-sr-policy-ifit]  
Chen, H., Yuan, H., Zhou, T., Li, W., Fioccola, G., and Y. Wang, "PCEP SR Policy Extensions to Enable IFIT", Work in Progress, Internet-Draft, draft-chen-pce-sr-policy-ifit-02, 10 July 2020, <<https://www.ietf.org/archive/id/draft-chen-pce-sr-policy-ifit-02.txt>>.
- [I-D.herbert-ipv4-eh]  
Herbert, T., "IPv4 Extension Headers and Flow Label", Work in Progress, Internet-Draft, draft-herbert-ipv4-eh-01, 2 May 2019, <<https://www.ietf.org/archive/id/draft-herbert-ipv4-eh-01.txt>>.
- [I-D.ietf-idr-sr-policy-ifit]  
Qin, F., Yuan, H., Zhou, T., Fioccola, G., and Y. Wang, "BGP SR Policy Extensions to Enable IFIT", Work in Progress, Internet-Draft, draft-ietf-idr-sr-policy-ifit-03, 10 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-idr-sr-policy-ifit-03.txt>>.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-data-17, 13 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-data-17.txt>>.

[I-D.ietf-ippm-ioam-deployment]

Brockners, F., Bhandari, S., Bernier, D., and T. Mizrahi, "In-situ OAM Deployment", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-deployment-00, 19 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-deployment-00.txt>>.

[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", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-direct-export-07, 13 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-direct-export-07.txt>>.

[I-D.ietf-ippm-ioam-ipv6-options]

Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-ipv6-options-07, 6 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-ipv6-options-07.txt>>.

[I-D.ietf-ippm-ioam-yang]

Zhou, T., Guichard, J., Brockners, F., and S. Raghavan, "A YANG Data Model for In-Situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-yang-03, 25 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-yang-03.txt>>.

[I-D.ietf-mboned-multicast-telemetry]

Song, H., McBride, M., Mirsky, G., Mishra, G., Asaeda, H., and T. Zhou, "Multicast On-path Telemetry Solutions", Work in Progress, Internet-Draft, draft-ietf-mboned-multicast-telemetry-02, 4 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-mboned-multicast-telemetry-02.txt>>.

[I-D.ietf-opsawg-ntf]

Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", Work in Progress, Internet-Draft, draft-ietf-opsawg-ntf-13, 3 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-ntf-13.txt>>.

[I-D.li-apn-framework]

Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-04, 25 October 2021, <<https://www.ietf.org/archive/id/draft-li-apn-framework-04.txt>>.

[I-D.mirsky-ippm-hybrid-two-step]

Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", Work in Progress, Internet-Draft, draft-mirsky-ippm-hybrid-two-step-12, 26 January 2022, <<https://www.ietf.org/archive/id/draft-mirsky-ippm-hybrid-two-step-12.txt>>.

[I-D.song-ippm-ioam-tunnel-mode]

Song, H., Li, Z., Zhou, T., and Z. Wang, "In-situ OAM Processing in Tunnels", Work in Progress, Internet-Draft, draft-song-ippm-ioam-tunnel-mode-00, 27 June 2018, <<https://www.ietf.org/archive/id/draft-song-ippm-ioam-tunnel-mode-00.txt>>.

[I-D.song-ippm-postcard-based-telemetry]

Song, H., Mirsky, G., Filsfils, C., Abdelsalam, A., Zhou, T., Li, Z., Shin, J., and K. Lee, "In-Situ OAM Marking-based Direct Export", Work in Progress, Internet-Draft, draft-song-ippm-postcard-based-telemetry-11, 15 November 2021, <<https://www.ietf.org/archive/id/draft-song-ippm-postcard-based-telemetry-11.txt>>.

[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., Andersson, L., and Z. Zhang, "MPLS Extension Header", Work in Progress, Internet-Draft, draft-song-mpls-extension-header-06, 10 January 2022, <<https://www.ietf.org/archive/id/draft-song-mpls-extension-header-06.txt>>.

[I-D.song-spring-siam]

Song, H. and T. Pan, "SRv6 In-situ Active Measurement", Work in Progress, Internet-Draft, draft-song-spring-siam-02, 6 December 2021, <<https://www.ietf.org/archive/id/draft-song-spring-siam-02.txt>>.

[I-D.wwx-netmod-event-yang]

Wu, Q., Bryskin, I., Birkholz, H., Liu, X., and B. Claise, "A YANG Data model for ECA Policy Management", Work in Progress, Internet-Draft, draft-wwx-netmod-event-yang-10, 1 November 2020, <<https://www.ietf.org/archive/id/draft-wwx-netmod-event-yang-10.txt>>.

[I-D.zhou-ippm-enhanced-alternate-marking]

Zhou, T., Fioccola, G., Liu, Y., Lee, S., Cociglio, M., and W. Li, "Enhanced Alternate Marking Method", Work in Progress, Internet-Draft, draft-zhou-ippm-enhanced-alternate-marking-08, 4 January 2022, <<https://www.ietf.org/archive/id/draft-zhou-ippm-enhanced-alternate-marking-08.txt>>.

[passport-postcard]

Handigol, N., Heller, B., Jeyakumar, V., Mazieres, D., and N. McKeown, "Where is the debugger for my software-defined network?", 2012, <<https://doi.org/10.1145/2342441.2342453>>.

[RFC5810]

Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, DOI 10.17487/RFC5810, March 2010, <<https://www.rfc-editor.org/info/rfc5810>>.

[RFC7011]

Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

[RFC8889]

Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <<https://www.rfc-editor.org/info/rfc8889>>.



[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <<https://www.rfc-editor.org/info/rfc8993>>.

## Authors' Addresses

Haoyu Song  
Futurewei  
2330 Central Expressway  
Santa Clara,  
United States of America  
Email: [haoyu.song@futurewei.com](mailto:haoyu.song@futurewei.com)

Fengwei Qin  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing, 100032  
P.R. China  
Email: [qinfengwei@chinamobile.com](mailto:qinfengwei@chinamobile.com)

Huanan Chen  
China Telecom  
Email: [chenhuan6@chinatelecom.cn](mailto:chenhuan6@chinatelecom.cn)

Jaehwan Jin  
LG U+  
South Korea  
Email: [daenamul@lguplus.co.kr](mailto:daenamul@lguplus.co.kr)

Jongyoon Shin  
SK Telecom  
South Korea  
Email: [jongyoon.shin@sk.com](mailto:jongyoon.shin@sk.com)

IPPM  
Internet-Draft  
Intended status: Standards Track  
Expires: September 1, 2022

T. Zhou, Ed.  
G. Fioccola  
Huawei  
Y. Liu  
China Mobile  
M. Cociglio  
Telecom Italia  
S. Lee  
LG U+  
W. Li  
Huawei  
February 28, 2022

Enhanced Alternate Marking Method  
draft-zhou-ippm-enhanced-alternate-marking-09

Abstract

This document extends the IPv6 Alternate Marking Option to provide enhanced capabilities and allow advanced functionalities. With this extension, it can be possible to perform thicker packet loss measurements and more dense delay measurements with no limitation for the number of concurrent flows under monitoring.

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 September 1, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                       |   |
|---------------------------------------|---|
| 1. Introduction . . . . .             | 2 |
| 1.1. Requirements Language . . . . .  | 3 |
| 2. Data Fields Format . . . . .       | 3 |
| 3. Security Considerations . . . . .  | 6 |
| 4. IANA Considerations . . . . .      | 6 |
| 5. References . . . . .               | 7 |
| 5.1. Normative References . . . . .   | 7 |
| 5.2. Informative References . . . . . | 7 |
| Authors' Addresses . . . . .          | 8 |

## 1. Introduction

The Alternate Marking [RFC8321] and Multipoint Alternate Marking [RFC8889] define the Alternate Marking technique that is a 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.

The IPv6 AltMark Option [I-D.ietf-6man-ipv6-alt-mark] applies the Alternate Marking Method to IPv6, and defines an Extension Header Option to encode the Alternate Marking Method for both the Hop-by-Hop Options Header and the Destination Options Header. Similarly, SRv6 AltMark [I-D.fz-spring-srv6-alt-mark] defines how Alternate Marking data is carried as a TLV in the Segment Routing Header.

While the IPv6 AltMark Option implements the basic alternate marking methodology, this document defines extended data fields for the AltMark Option and provides enhanced capabilities to overcome some challenges and enable future proof applications.

It is worth mentioning that the enhanced capabilities are intended for further use and are optional.

Some possible enhanced applications MAY be:

1. thicker packet loss measurements: the single marking method of the base AltMark Option can be extended with additional marking bits in order to get shortest marking periods under the same timing conditions.
2. more dense delay measurements: than double marking method of the base AltMark Option can be extended with additional marking bits in order to identify down to each packet as delay sample.
3. increase the number of concurrent flows under monitoring: if the 20-bit FlowMonID is set independently and pseudo randomly, there is a 50% chance of collision for 1206 flows. The size of FlowMonID can be extended to raise the entropy and therefore to increase the number of concurrent flows that can be monitored.

#### 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. Data Fields Format

The Data Fields format is represented in Figure 1. A 4-bit NH(NextHeader) field is allocated from the Reserved field of IPv6 AltMark Option [I-D.ietf-6man-ipv6-alt-mark]. It is worth highlighting that remaining bits of the former Reserved field continue to be reserved.

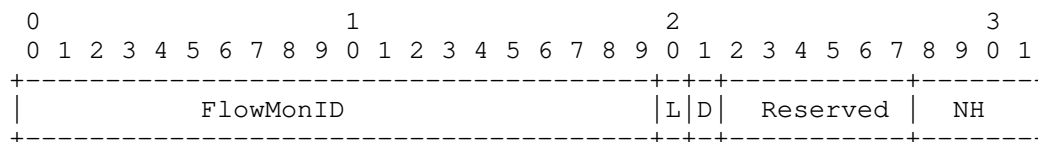


Figure 1: Data fields indicator for enhanced capabilities

The NH (NextHeader) field is used to indicate the extended data fields which are used for enhanced capabilities:

NextHeader value of 0x00 is reserved for backward compatibility. It means that there is no extended data field attached.

NextHeader values of 0x01-0x08 are reserved for private use or for experimentation.

NextHeader value of 0x09 indicates the extended data fields. The format is showed in Figure 2.

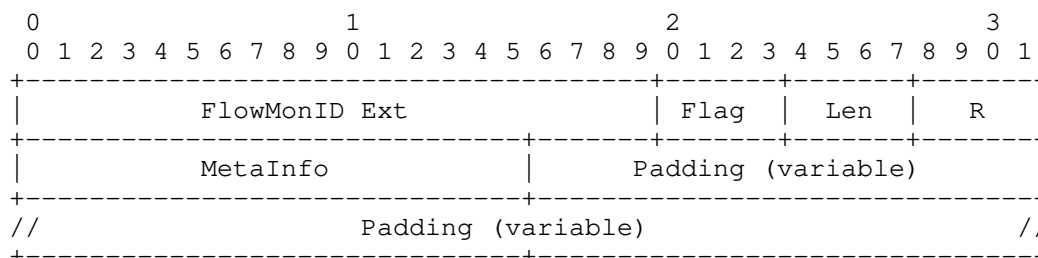


Figure 2: Data fields extension for enhanced alternate marking

where:

- o FlowMonID Ext - 20 bits unsigned integer. This is used to extend the FlowMonID in order to reduce the conflict when random allocation is applied. The disambiguation of the FlowMonID field is discussed in IPv6 AltMark Option [I-D.ietf-6man-ipv6-alt-mark].
- o Flag - A 4-bit flag to indicate the special purpose usage (see below).
- o Len - Length. It indicates the length of the enhanced alternate marking extension in bytes.
- o R - Reserved for further use. These bits MUST be set to zero on transmission and ignored on receipt.
- o MetaInfo - A 16-bit Bitmap to indicate more meta data attached for the enhanced function (see below).
- o Padding - These bits MUST be set to zero when not being used.

The Flag is defined in Figure 3 as:

- o bit 0 - Measurement mode, M bit. If M=0, it indicates that it is for hop-by-hop monitoring. If M=1, it indicates that it is for end-to-end monitoring.
- o bit 2 - Flow direction identification, F bit. This flag is used in the case backward direction flow monitoring is requested to be set up automatically. If F=1, it indicates that the flow direction is forward. If F=0, it indicates that the flow direction is backward.

- o others (shown as R) - Reserved. These bits MUST be set to zero and ignored on receipt.

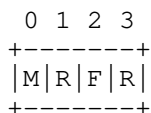


Figure 3: Flag data field

The MetaInfo is defined in the following Figure 4 as a bit map:

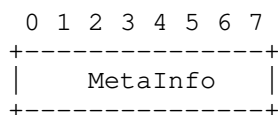


Figure 4: MetaInfo data field

- o bit 0: it indicates a 6 bytes Timestamp that is attached as Padding after the MetaInfo. Timestamp(s) stands for the number of seconds in the timestamp. It will overwrite the Padding after MetaInfo. Timestamp(ns) stands for the number of sub-seconds in the timestamp with the unit of nano second. This Timestamp is filled by the encapsulation node, and is taken all the way to the decapsulation node. So that all the intermediate nodes could compare it with its local time, and measure the one way delay.

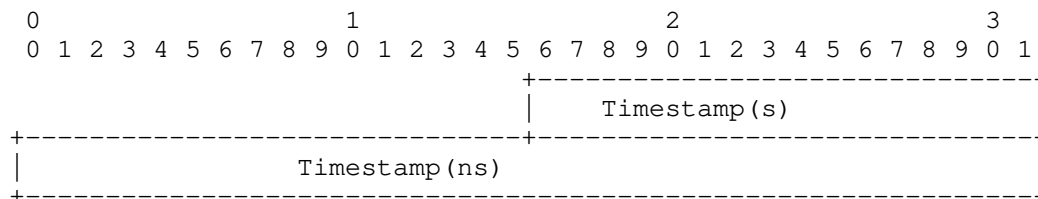


Figure 5: Timestamp data field

- o bit 1: it indicates the control information with the following data format that is attached as Padding after the MetaInfo:

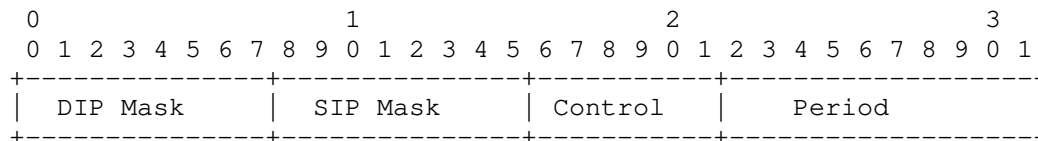


Figure 6: Control words for backward direction flow monitoring

This is used to set up the backward direction flow monitoring.  
Where:

- \* DIP Mask: it is the length of the destination IP prefix.
  - \* SIP Mask: it is the length of the source IP prefix.
  - \* Control: it indicates more match fields to set up the backward direction flow monitoring.
  - \* Period: it indicates the alternate marking period with the unit of second.
- o bit 2: it indicates a 4 bytes Sequence number with the following data format that is attached as Padding after the MetaInfo. The unique Sequence could be used to detect the out-of-order packets, in addition to the normal loss measurement. More over, the Sequence can be used together with the latency measurement, so as to get the per packet timestamp.

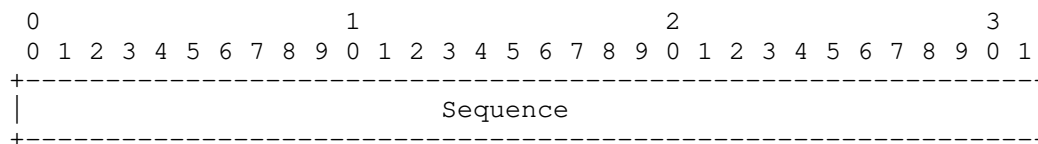


Figure 7: Sequence number data field

It is worth noting that the meta data information forming the Padding and specified above in Figure 5, Figure 6 and Figure 7 must be ordered according to the order of the MetaInfo bits.

### 3. Security Considerations

IPv6 AltMark Option [I-D.ietf-6man-ipv6-alt-mark] analyzes different security concerns and related solutions. These aspects are valid and applicable also to this document. In particular the fundamental security requirement is that Alternate Marking MUST only be applied in a specific limited domain, as also mentioned in [RFC8799].

### 4. IANA Considerations

This document has no request to IANA.

## 5. References

### 5.1. Normative References

- [I-D.fz-spring-srv6-alt-mark]  
Fioccola, G., Zhou, T., and M. Cociglio, "Segment Routing Header encapsulation for Alternate Marking Method", draft-fz-spring-srv6-alt-mark-02 (work in progress), February 2022.
- [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-12 (work in progress), October 2021.
- [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>>.
- [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>>.

### 5.2. Informative References

- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8889] Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <<https://www.rfc-editor.org/info/rfc8889>>.



## 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

Yisong Liu  
China Mobile  
Beijing  
China

Email: liuyisong@chinamobile.com

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

Email: mauro.cociglio@telecomitalia.it

Shinyoung Lee  
LG U+  
71, Magokjungang 8-ro, Gangseo-gu  
Seoul  
Republic of Korea

Email: leesy@lguplus.co.kr

Weidong Li  
Huawei  
156 Beiqing Rd.  
Beijing 100095  
China

Email: poly.li@huawei.com