

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 14, 2021

D. Dang, Ed.
Huawei
W. Wang
China Telecom
L. LEE
LG U+
C. Cheng
Huawei
August 13, 2020

Multi-Path Concurrent Measurement for IPPM
draft-dang-ippm-multiple-path-measurement-05

Abstract

This test method can test multi-paths concurrently from one edge node to another edge node. This document details Multi-Path Concurrent Measurement (MPCM).

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. Requirements Language	3
1.2. Terminology & Abbreviations	3
2. Overview of MPCM	4
2.1. Principle	4
2.1.1. Single Path Measurement	4
2.1.2. Multiple Path Measurement	6
3. MPCM-Test Packet Format and Content	7
4. Expansion based on various measurement methods	10
4.1. IOAM	10
5. Data Export	10
6. IANA Considerations	10
7. Security Considerations	10
8. Acknowledgements	11
9. Normative References	11
Authors' Addresses	12

1. Introduction

As we know, the current network has been already being in load balancing mode, however it is partially congested. In other words, from the same source node to the same destination node, some paths have been congested to cause a decline in service quality, but some paths carry less traffic and are lightly loaded. To solve the problem of unbalanced network load[draft-liu-ican], the first is to have the ability to detect the quality of the load sharing paths. And then the traffic from the Src node to the Dst node is required to be steered from the congested paths into the lightly loaded path/paths basing on the SLA's requirement. So it's necessary to measure the multi-paths in load-balancing mode.

In the traditional method, the paths are measured separately because they aren't maintained by the path group. If the multiple load sharing paths are required to be selected based on the SLA information, the measured SLA information needs to be comparable. If you want to ensure that the data obtained by the test is available and accurate, the multi-paths are required to maintain by the path group in order that the test start and end points must be same.

For example, the low latency services require millisecond delays. If the start time and the end time aren't same, the measured data may

not be in one test cycle, and the accuracy of this data is relatively low and the data cannot be compared Figure 1.

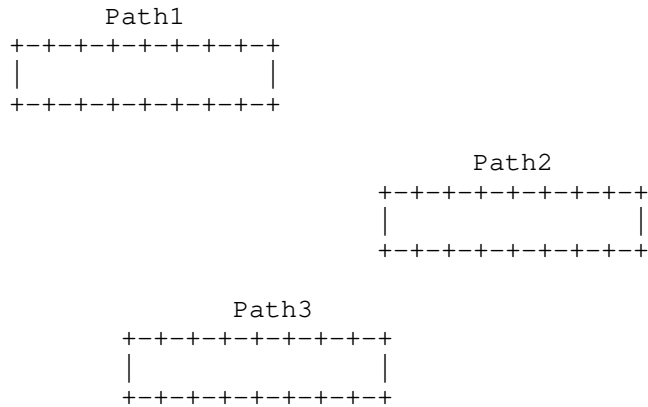


Figure 1: Measured Data in the Different Cycles

The Multi-Path Concurrent Measurement (MPCM) is required, which can be used bi-directionally to concurrently measure multi-paths metrics between two network elements. At the same time, this method also consider saving the number of test messages to reduces the load on the network.

1.1. 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.

1.2. Terminology & Abbreviations

o Muti-paths

- * There are multiple paths between two nodes in the network. These paths may be equal-cost multi-path (ECMP) mode or unequal-cost multiple (UCMP) mode. In a real network, they might be one [draft-ietf-spring-segment-routing-policy] or [RFC7348] tunnel group.

o Concurrent

- * In order to ensure comparability between multiple paths, the test start point and the test end point are required to be same.

2. Overview of MPCM

The Multi-Path Concurrent Measurement (MPCM) is the way of measurement of multi-paths metrics.

MPCM can be embedded into a variety of transports such as NSH, Segment Routing, VxLAN, native IPv6 (via extension header), or IPv4.

2.1. Principle

To complete the target scenario, we need to optimize the single-path measurement mechanism, and then further diffuse the single-path measurement mechanism to multiple-path.

1. For a single tunnel, the Dst needs to know when to start timing in order to delimit. The Dst needs to solve various problems such as congestion and discarding of measurement packets. Therefore, the Dst needs to initiate a periodic response.

2. For multiple paths, the Dst needs to respond one measurement message with multiple path information in its specific time, solving the problems such as inconsistent initiation of any path, inconsistent measurement periods, clock drift, and different delays.

2.1.1. Single Path Measurement

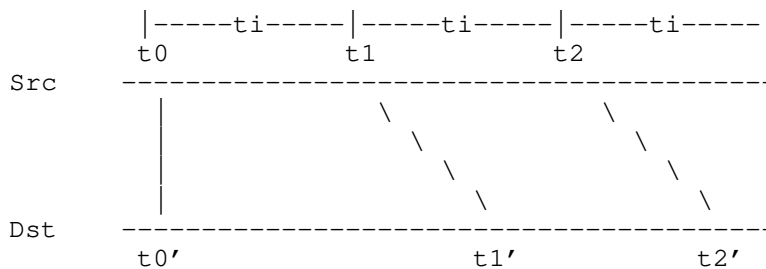


Figure 2: Single path measurement

A path between Scr node and Dst node in the network to obtain measurement results at equal intervals is as follows:

- 1) Set the measurement interval t_i .

- a) Before the test starts, the Scr sends a protocol packet to the Dst and sets the test interval t_i .
- b) After receiving the protocol message, the Dst sets the test interval t_i for the Scr and Dst, and replies to the Scr to confirm that the setting is successful. The congestion at the Dst will be counted at intervals t_i .
- c) After receiving the interval setting successfully, the Scr starts to start measurement.
- 2) The Scr sends the first delimited message, which includes the sending timestamp t_0 , and starts to count the data packets sent.
- 3) After receiving the first delimited message, the Dst end stamps the time stamp t_0' and starts to count the received data messages.
- 4) The Scr sends the second delimited message at time t_1 , where $t_1 = t_0 + t_i$, the message includes the sending timestamp t_1 , and counts the number of data packets sent. The first delimited message uses high priority, and the second delimited message uses normal priority. Because the second delimitation message has a low priority and a large queuing delay, the interval between the first delimitation message and the second delimitation message shall become larger at the Dest.
- 5) At the time $t_0' + t_i$, the Dst counts the number of packets received between t_0' and $t_0' + t_i$, and sends the message back to the Src with the number of packets, the sending time t_0 and the receiving time t_0' . If the delimitation message has not been received at $t_0' + 2 * t_i$ time, the Dst repeats the previous actions, and so on.
- 6) When the second delimited message arrives at the Dst, the Dst counts the number of packets received between t_0' and t_1' at t_1' time, and sends the message back to the Src with the number of packets, the packet sending time t_0 and the packet receiving time t_0' .
- 7) After t_1' , the sending time in the message from the Dst is updated to t_1 , and the receiving time in the message from the Dst is updated to t_1' . The number of packets is still the number of packets received within t_i time.
- 8) Assuming that t_1' is between $t_0' + (x-1) * t_i$ and $t_0' + x * t_i$, then the congestion in the interval t_i is calculated in two parts. The first part is from $t_0' + (x-1) * t_i$ to t_1' , The statistics packets sent at t_1' must include the packet statistics and time t_0' ;

the second part is from t_1' to $t_0' + x * t_i$, $t_0' + x * t_i$ need to include the packet statistics and t_1' .

9) Repeat the above steps.

2.1.2. Multiple Path Measurement

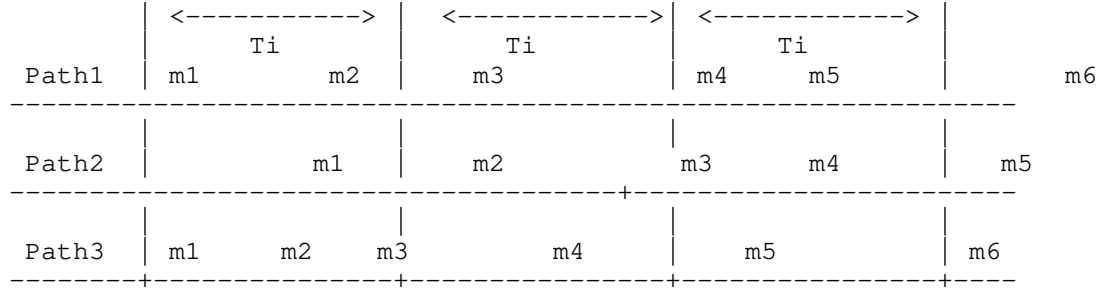


Figure 3: Multiple path measurement

There are multiple paths in the tunnel between Src node and Dst nodes in the network. This method is mainly implemented at the Dst.

1) Set the measurement interval t_i .

a) Before the test starts, the Src sends a protocol packet to the Dst, setting the number of paths and the measurement interval t_i . The measurement result of each path is a message with measurement data.

b) After receiving the protocol message, the Dst sets the number of paths and measurement interval t_i , and replies to the source to confirm the successful setting.

c) After receiving the message with the number of paths and measurement interval, the Src starts to start measurement.

2) On each path, the Src continuously sends measurement packets, and the Dst continuously calculates the measurement results at intervals t_i .

3) The Dst collects the measurement results of each path at intervals t_i after the earliest measurement result of multiple paths is came out.

4) The results of multiple paths in the same interval time t_i are counted as a group. If there is no measured results on the specific path in the interval t_i , the relevant information is set 0 in the

group results. A set of measurement results packaged of multiple paths are taken back to the Src.

5) The measurement results of multiple paths on the Dst are continuously packaged at intervals t_i and sent back to the Src. The packaged message carries the sequence number within the message to prevent out of order.

3. MPCM-Test Packet Format and Content

This section defines path header and associated data types required for MPCM.

Firstly one path packet formatFigure 4 of multi-path can be defined.

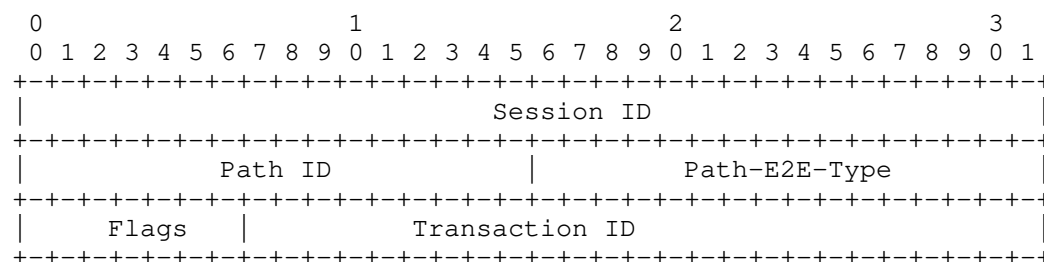


Figure 4: MPCM Path header

- o Session ID: A set of load sharing paths
- o Path ID: One path of the session.
- o Path-E2E-Type: A 16-bit identifier which Indicates whether the packet type is a send message or a request message.
- o Flags: 8-bit field. Identify the query or response type. Following flags are defined:
 - * Bit 0 Identify the query type
 - * Bit 1 Identify the response type
 - * Reserved
- o Transaction ID: 16-bit identifier of one measurement transaction. The sender and receiver to identify measurement transactions based on Transaction ID.

When a measurement is for a set of paths, each query message is made for each path, but only one unified response message repliesFigure 5.

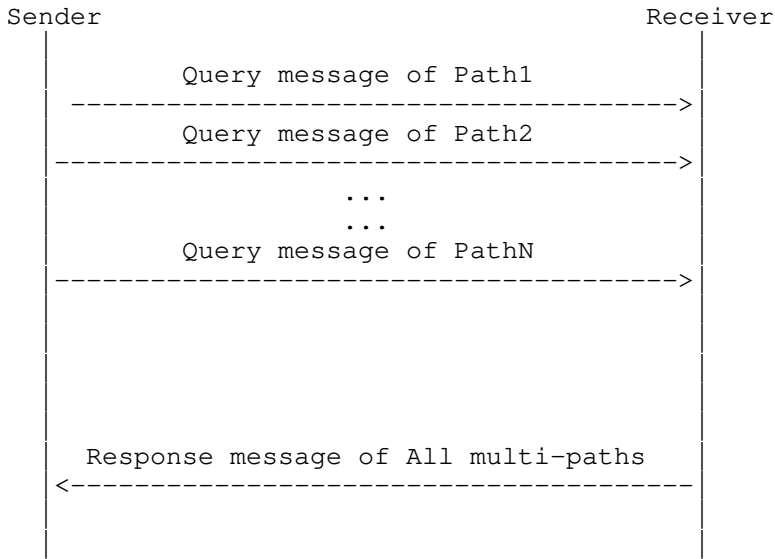


Figure 5: Query and Response message

The measurement response packet format of a path is as followsFigure 6.

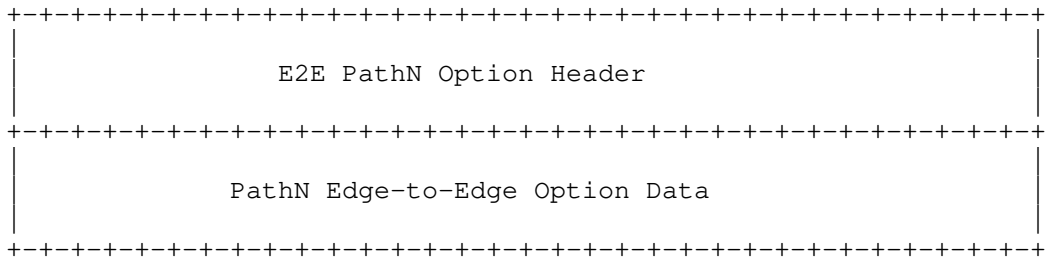


Figure 6: Query message

The field of PathN Edge-to-Edge Option Data can refer to Edge-to-Edge Option Data of [draft-ietf-ippm-ioam-data-04].

It suppose there are N paths between two points.The measurement response packet format of multi-paths is as followsFigure 6.

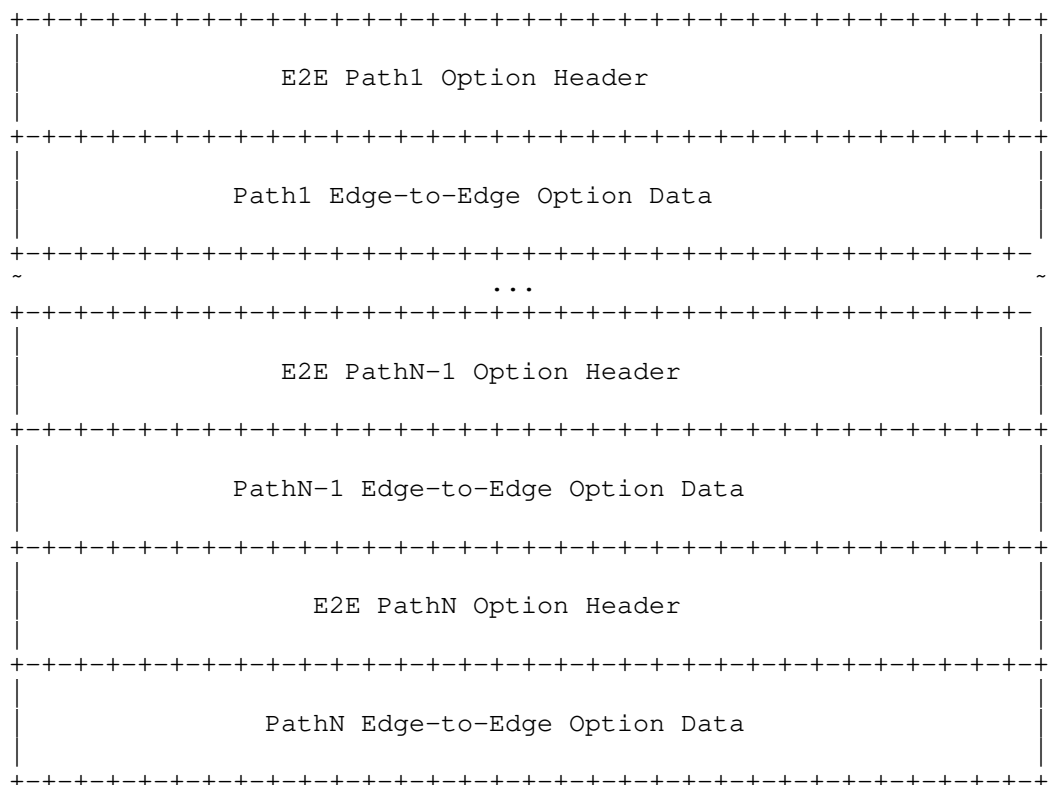


Figure 7: Response message

- o Long-term measurement
 - * The receiver can wait until it receives all measurement requests of a set of path and then responds.
- o Short-term measurement
 - * The Sender can query once t.
 - * The receiver can reply once t.

The overall solution needs to consider two methods of long-period measurement and short-period measurement.

4. Expansion based on various measurement methods

The measurement message format defined by this document can be extended based on various measurement methods.

4.1. IOAM

A new type may be added in IOAM-E2E-Type of IOAM Edge-to-Edge Option header[draft-ietf-ippm-ioam-data-04-section4.4] as follow.

- o Bit 4: Multiple paths measurement.

This bit is set by the headend node if Multi-Path Concurrent Measurement is activated.

A common registry is maintained for IOAM-Types, see Section 6.

For path-based quality measurements, there is no need to measure each message because the large-scale deployment consumes too much network resources. Here, the way of periodic measurement is recommended. In a period, if there is a packet, the appropriate packet is selected to be inserted into the IOAM packet; if there is no packet, a measurement packet is directly generated[draft-dang-ippm-congestion].

5. Data Export

MPCM nodes collect information for packets traversing a domain that supports MPCM. MPCM process the information further and export the information using e.g., IPFIX. Raw data export of IOAM data using IPFIX is discussed in [draft-spiegel-ippm-ioam-rawexport-00].

6. IANA Considerations

This document requests the following IANA Actions.

IOAM E2E Type Registry:

Bit 4 Multiple ways measurement

7. Security Considerations

The Proof of Transit option (Section Section 4.3 In-situ OAM [draft-ietf-ippm-ioam-data-04-section4.4]) is used for verifying the path of data packets.

8. Acknowledgements

TBD

9. Normative References

- [draft-dang-ippm-congestion]
"A One-Path Congestion Metric for IPPM",
<<https://tools.ietf.org/html/draft-dang-ippm-congestion-02>>.
- [draft-ietf-ippm-ioam-data-04]
"A Variety of Transports",
<<https://datatracker.ietf.org/doc/draft-ietf-ippm-ioam-data/>>.
- [draft-ietf-ippm-ioam-data-04-section4.4]
"IOAM Edge-to-Edge Option",
<<https://datatracker.ietf.org/doc/draft-ietf-ippm-ioam-data/>>.
- [draft-ietf-spring-segment-routing-policy]
"Segment Routing Policy Architecture",
<<https://tools.ietf.org/html/draft-ietf-spring-segment-routing-policy-02>>.
- [draft-liu-ican]
"Instant Congestion Assessment Network (iCAN) for Traffic Engineering", <<https://tools.ietf.org/html/draft-dang-ippm-congestion-02>>.
- [draft-spiegel-ippm-ioam-rawexport-00]
"In-situ OAM raw data export with IPFIX",
<<https://tools.ietf.org/html/draft-spiegel-ippm-ioam-rawexport-00>>.
- [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>>.
- [RFC7348] "Virtual eXtensible Local Area Network (VXLAN)", <<https://datatracker.ietf.org/doc/rfc7348/>>.

Authors' Addresses

Joanna Dang (editor)
Huawei
Beijing
China

Email: dangjuanna@huawei.com

Jianglong
China Telecom
Beijing
China

Email: wangjl50@chinatelecom.cn

Shinyoung
LG U+
Seoul
Korea

Email: leesy@lguplus.co.kr

Liang
Huawei
Beijing
China

Email: liang.cheng@huawei.com

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 λ is the average packet spacing, thus the Run-time Parameter is $\text{Reciprocal_}\lambda = 1/\lambda$, 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, 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 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 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", 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

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>

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: 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

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-Namespaces. This means that an IOAM node which is, e.g., an IOAM-decapsulating node for IOAM-Namespaces "A" but not for IOAM-Namespaces "B" will only remove the IOAM-Option-Types for IOAM-Namespaces "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.

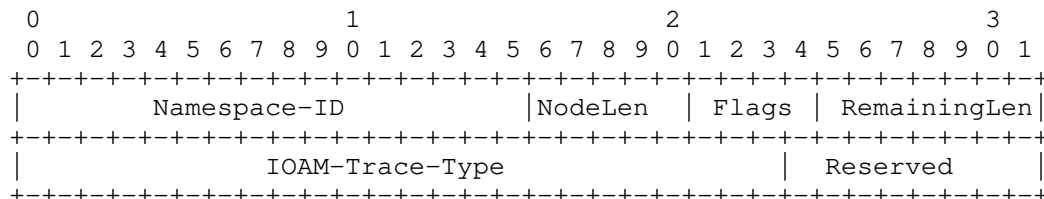
- o Generic data: Format-free information where syntax and semantic of the information is defined by the operator in a specific deployment. For a specific IOAM-Namespace, all IOAM nodes have to interpret the generic data the same way. Examples for generic IOAM data include geo-location information (location 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.

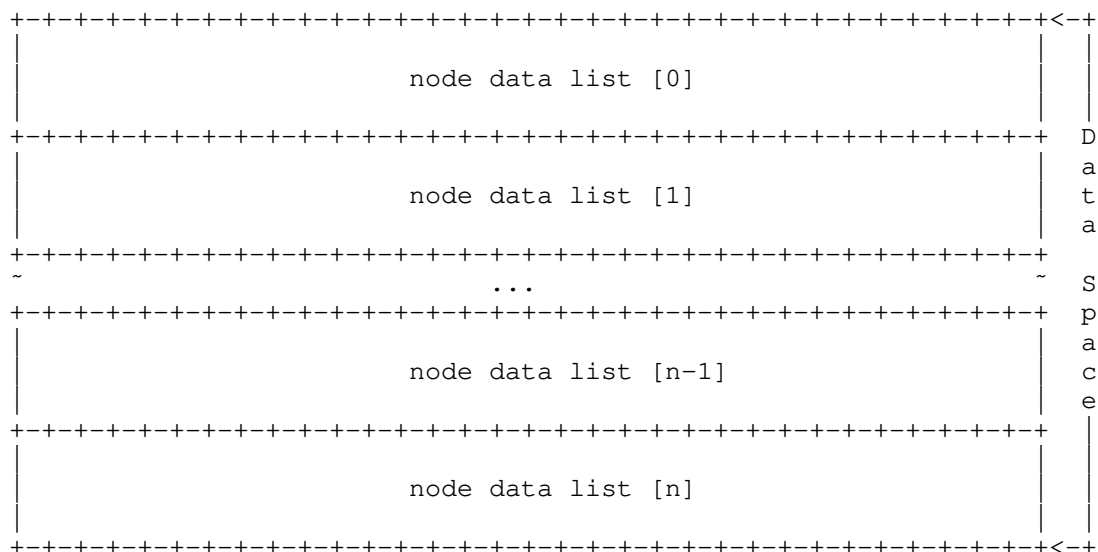
5.4.1. Pre-allocated and Incremental Trace Option-Types

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.

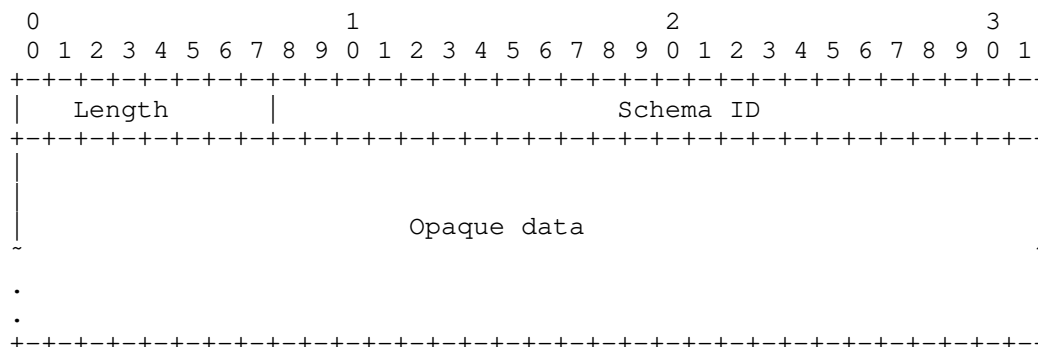
```

 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.

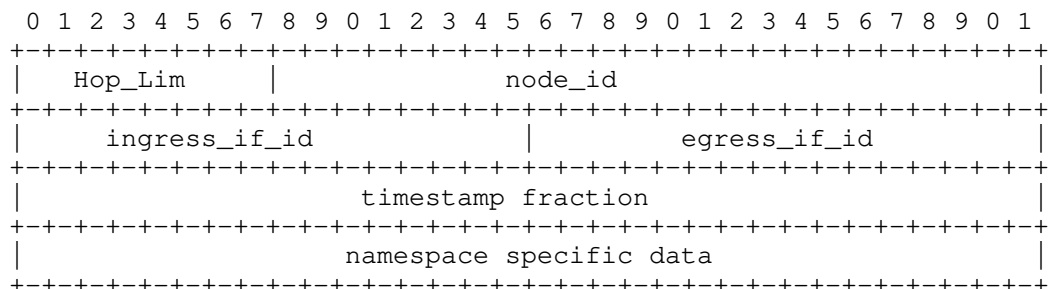
Opaque data: Variable length field. This field is interpreted as specified by the schema identified by the Schema ID.

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.

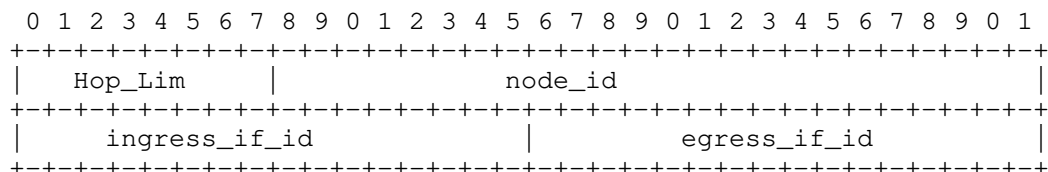
5.4.3. Examples of IOAM node data

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".

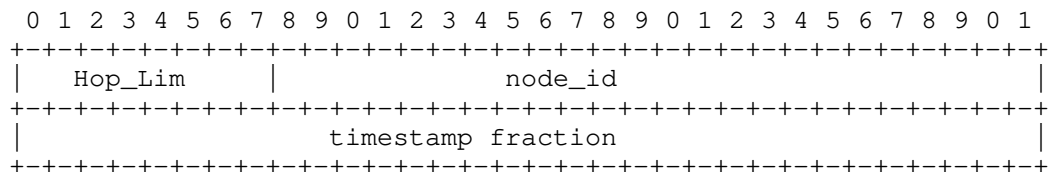
0xD40000: IOAM-Trace-Type is 0xD40000 (0b110101000000000000000000)
then the format of node data is:



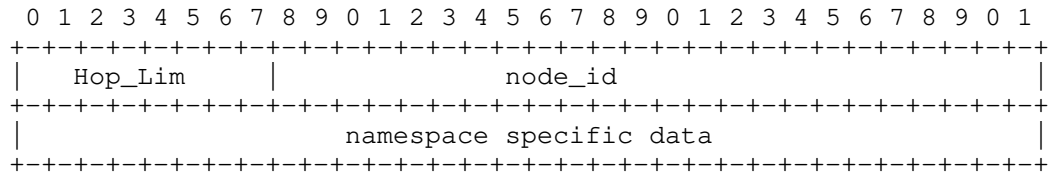
0xC00000: IOAM-Trace-Type is 0xC00000 (0b110000000000000000000000)
then the format is:



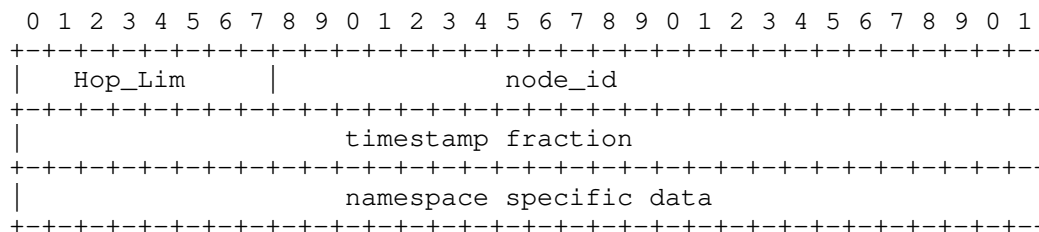
0x900000: IOAM-Trace-Type is 0x900000 (0b100100000000000000000000)
then the format is:



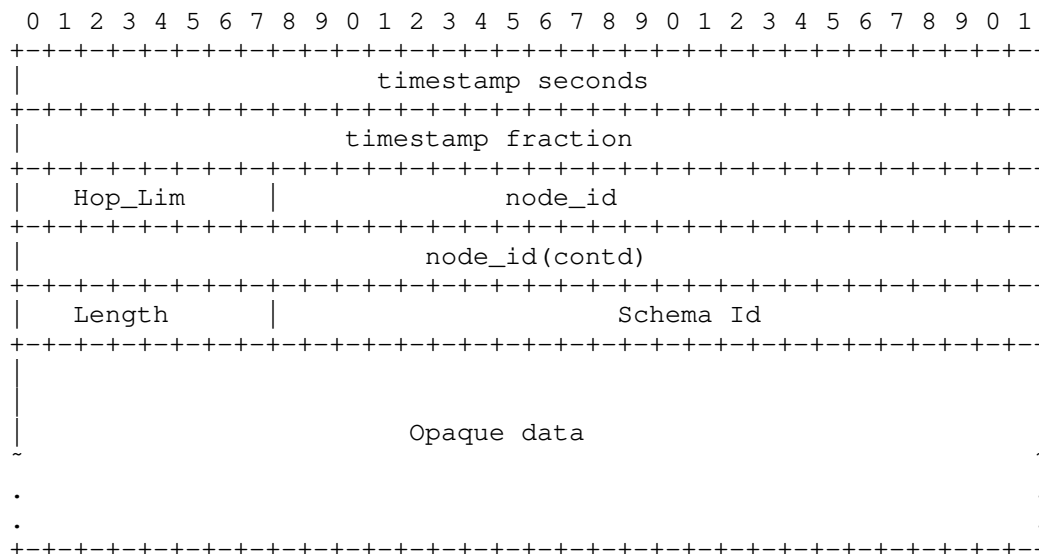
0x840000: IOAM-Trace-Type is 0x840000 (0b100001000000000000000000)
then the format is:



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



0x308002: IOAM-Trace-Type is 0x308002 (0b00110000010000000000000010)
 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.

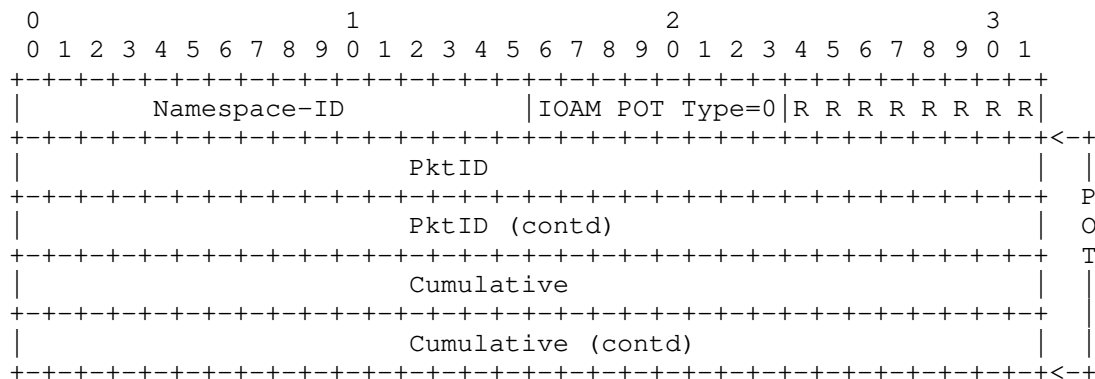
If a node receives an IOAM POT Type value that it does not understand, the node MUST NOT change, add to, or remove the contents of the OAM-Data-Fields.

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.

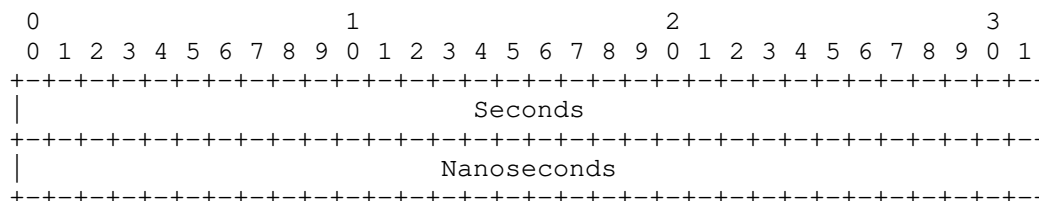


Figure 1: PTP Truncated Timestamp Format

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.

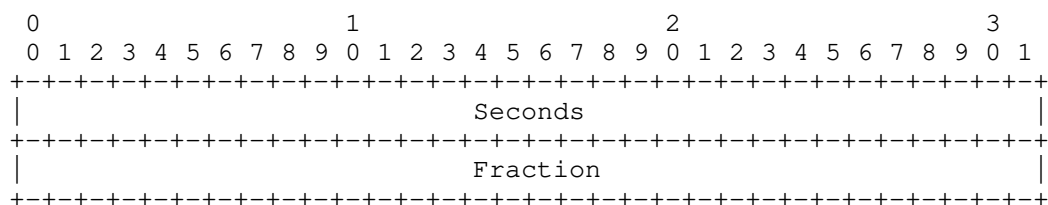


Figure 2: NTP [RFC5905] 64-bit Timestamp Format

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.

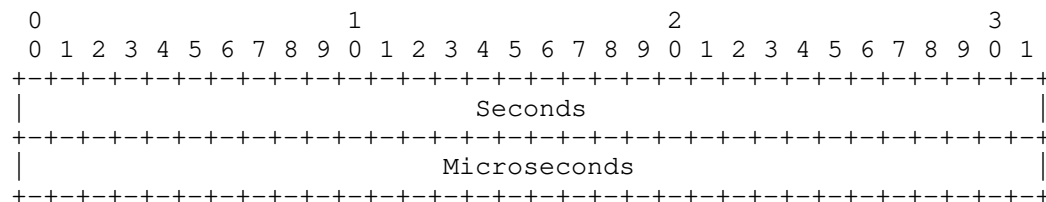


Figure 3: POSIX-based Timestamp Format

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

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

IPPM
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2022

T. Mizrahi
Huawei
F. Brockners
Cisco
S. Bhandari, Ed.
Thoughtspot
R. Sivakolundu
C. Pignataro
Cisco
A. Kfir
B. Gafni
Nvidia
M. Spiegel
Barefoot Networks, an Intel company
J. Lemon
Broadcom
October 13, 2021

In-situ OAM Loopback and Active Flags
draft-ietf-ippm-ioam-flags-07

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) collects operational and telemetry information in packets while they traverse a path between two points in the network. This document defines two new flags in the IOAM Trace Option headers, specifically the Loopback and Active flags.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 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	2
2. Conventions	3
2.1. Requirements Language	3
2.2. Terminology	3
3. New IOAM Trace Option Flags	3
4. Loopback in IOAM	3
4.1. Loopback: Encapsulating Node Functionality	4
4.1.1. Loopback Packet Selection	5
4.2. Receiving and Processing Loopback	6
4.3. Loopback on the Return Path	7
4.4. Terminating a Looped Back Packet	7
5. Active Measurement with IOAM	7
6. IANA Considerations	9
7. Performance Considerations	9
8. Security Considerations	10
9. Acknowledgments	11
10. References	11
10.1. Normative References	11
10.2. Informative References	12
Authors' Addresses	12

1. Introduction

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

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

two new flags in the Pre-allocated and Incremental Trace options: the Loopback and Active flags.

The Loopback flag is used to request that each transit device along the path loops back a truncated copy of the data packet to the sender. The Active flag indicates that a packet is used for active measurement. The term active measurement in the context of this document is as defined in [RFC7799].

2. Conventions

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

3. New IOAM Trace Option Flags

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

Bit 1 "Loopback" (L-bit). When set, the Loopback flag triggers sending a copy of a packet back towards the source, as further described in Section 4.

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

4. Loopback in IOAM

The Loopback flag is used to request that each transit device along the path loops back a truncated copy of the data packet to the sender. Loopback allows an IOAM encapsulating node to trace the path to a given destination, and to receive per-hop data about both the

forward and the return path. Loopback is intended to provide an accelerated alternative to Traceroute, that allows the encapsulating node to receive responses from multiple transit nodes along the path in less than one round-trip-time, and by sending a single packet.

As illustrated in Figure 1, an IOAM encapsulating node can push an IOAM encapsulation that includes the Loopback flag onto some or all of the packets it forwards. The IOAM transit node and the decapsulating node both create copies of the packet and loop them back to the encapsulating node. The decapsulating node also terminates the IOAM encapsulation, and then forwards the packet towards the destination. The two IOAM looped back copies are terminated by the encapsulating node.

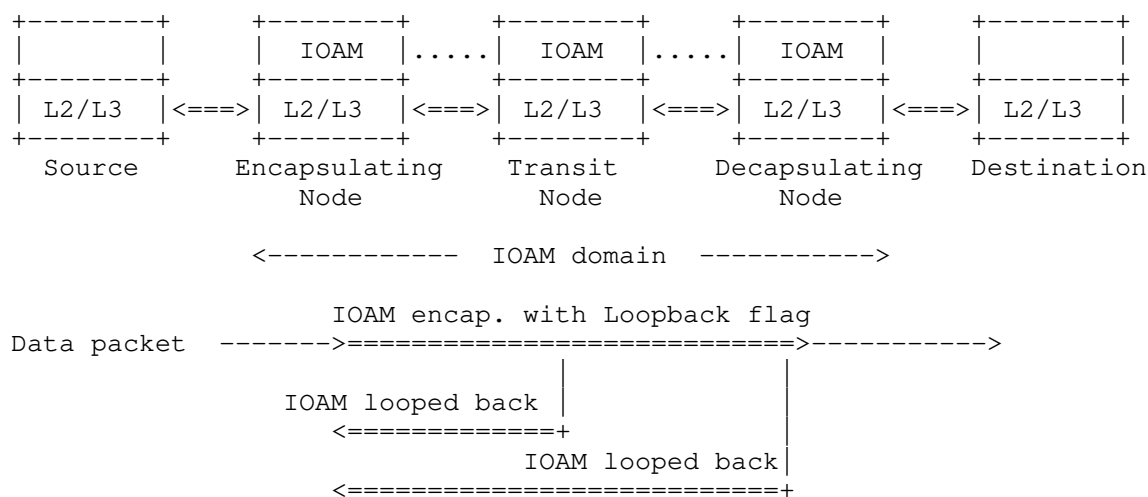


Figure 1: Loopback in IOAM.

Loopback can be used only if a return path from transit nodes and destination nodes towards the source (encapsulating node) exists. Specifically, loopback is only applicable in encapsulations in which the identity of the encapsulating node is available in the encapsulation header. If an encapsulating node receives a looped back packet that was not originated from the current encapsulating node, the packet is dropped.

4.1. Loopback: Encapsulating Node Functionality

The encapsulating node either generates synthetic packets with an IOAM trace option that has the Loopback flag set, or sets the loopback flag in a subset of the in-transit data packets. Loopback is used

either proactively or on-demand, i.e., when a failure is detected. The encapsulating node also needs to ensure that sufficient space is available in the IOAM header for loopback operation, which includes transit nodes adding trace data on the original path and then again on the return path.

An IOAM trace option that has the Loopback flag set MUST have the value '1' in the most significant bit of IOAM-Trace-Type, and '0' in the rest of the bits of IOAM-Trace-Type. Thus, every transit node that processes this trace option only adds a single data field, which is the Hop_Lim and node_id data field. A transit node that receives a packet with an IOAM trace option that has the Loopback flag set and the IOAM-Trace-Type is not equal to '1' in the most significant bit and '0' in the rest of the bits, MUST NOT loop back a copy of the packet. The reason for allowing a single data field per hop is to minimize the impact of amplification attacks.

IOAM encapsulating nodes MUST NOT push an IOAM encapsulation with the Loopback flag onto data packets that already include an IOAM encapsulation. This requirement is intended to prevent IOAM Loopback nesting, where looped back packets may be subject to loopback in a nested IOAM domain.

4.1.1. Loopback Packet Selection

If an IOAM encapsulating node incorporates the Loopback flag into all the traffic it forwards it may lead to an excessive amount of looped back packets, which may overload the network and the encapsulating node. Therefore, an IOAM encapsulating node that supports the Loopback flag MUST support the ability to incorporate the Loopback flag selectively into a subset of the packets that are forwarded by it.

Various methods of packet selection and sampling have been previously defined, such as [RFC7014] and [RFC5475]. Similar techniques can be applied by an IOAM encapsulating node to apply Loopback to a subset of the forwarded traffic.

The subset of traffic that is forwarded or transmitted with a Loopback flag SHOULD NOT exceed $1/N$ of the interface capacity on any of the IOAM encapsulating node's interfaces. It is noted that this requirement applies to the total traffic that incorporates a Loopback flag, including traffic that is forwarded by the IOAM encapsulating node and probe packets that are generated by the IOAM encapsulating node. In this context N is a parameter that can be configurable by network operators. If there is an upper bound, M , on the number of IOAM transit nodes in any path in the network, then it is recommended to use an N such that $N \gg M$. The rationale is that a packet that

includes the Loopback flag triggers a looped back packet from each IOAM transit node along the path for a total of M looped back packets. Thus, if $N \gg M$ then the number of looped back packets is significantly lower than the number of data packets forwarded by the IOAM encapsulating node. If there is no prior knowledge about the network topology or size, it is recommended to use $N > 100$.

The loopback flag MUST NOT be set if it is not guaranteed that there is a return path from each of the IOAM transit and IOAM decapsulating nodes, or if the encapsulating node's identity is not available in the encapsulation header.

4.2. Receiving and Processing Loopback

A Loopback flag that is set indicates to the transit nodes processing this option that they are to create a copy of the received packet and send the copy back to the source of the packet. In this context the source is the IOAM encapsulating node, and it is assumed that the source address is available in the encapsulation header. Thus, the source address of the original packet is used as the destination address in the copied packet. If the address of the encapsulating node is not available in the encapsulation header, then the transit/decapsulating node does not loop back a copy of the original packet. The address of the node performing the copy operation is used as the source address. The IOAM transit node pushes the required data field *after* creating the copy of the packet, in order to allow any egress-dependent information to be set based on the egress of the copy rather than the original packet. The copy is also truncated, i.e., any payload that resides after the IOAM option(s) is removed before transmitting the looped back packet back towards the encapsulating node. The original packet continues towards its destination. The L-bit MUST be cleared in the copy of the packet that a node sends back towards the source.

An IOAM node that supports the reception and processing of the Loopback flag MUST support the ability to limit the rate of the looped back packets. The rate of looped back packets SHOULD be limited so that the number of looped back packets is significantly lower than the number of packets that are forwarded by the device. The looped back data rate SHOULD NOT exceed $1/N$ of the interface capacity on any of the IOAM node's interfaces. It is recommended to use $N > 100$. Depending on the IOAM node's architecture considerations, the loopback response rate may be limited to a lower number in order to avoid loading the IOAM node.

4.3. Loopback on the Return Path

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

4.4. Terminating a Looped Back Packet

Once the return packet reaches the IOAM domain boundary, IOAM decapsulation occurs as with any other packet containing IOAM information. Note that the looped back packet does not have the L-bit set. The IOAM encapsulating node that initiated the original loopback packet recognizes a received packet as an IOAM looped-back packet by checking the Node ID in the Hop_Lim/node_id field that corresponds to the first hop. If the Node ID and IOAM-Namespace match the current IOAM node, it indicates that this is a looped back packet that was initiated by the current IOAM node, and processed accordingly. If there is no match in the Node ID, the packet is processed like a conventional IOAM-encapsulated packet.

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

5. Active Measurement with IOAM

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

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

An example of an IOAM deployment scenario is illustrated in Figure 2. The figure depicts two endpoints, a source and a destination. The data traffic from the source to the destination is forwarded through a set of network devices, including an IOAM encapsulating node, which incorporates one or more IOAM options, a decapsulating node, which removes the IOAM options, optionally one or more transit nodes. The IOAM options are encapsulated in one of the IOAM encapsulation types, e.g., [I-D.ietf-sfc-ioam-nsh], or [I-D.ietf-ippm-ioam-ipv6-options].

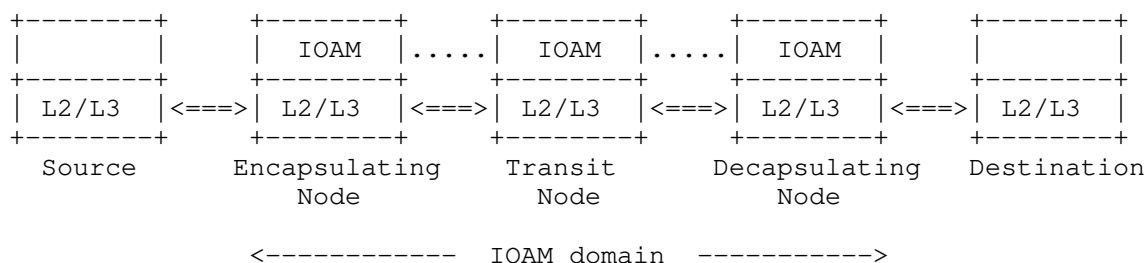


Figure 2: Network using IOAM.

This draft focuses on three possible use cases of active measurement using IOAM. These use cases are described using the example of Figure 2.

- o Endpoint active measurement: synthetic probe packets are sent between the source and destination, traversing the IOAM domain. Since the probe packets are sent between the endpoints, these packets are treated as data packets by the IOAM domain, and do not require special treatment at the IOAM layer. Specifically, the Active flag is not used in this case, and the IOAM layer needs not be aware that an active measurement mechanism is used at a higher layer.
- o IOAM active measurement using probe packets within the IOAM domain: probe packets are generated and transmitted by the IOAM encapsulating node, and are expected to be terminated by the decapsulating node. IOAM data related to probe packets may be exported by one or more nodes along its path, by an exporting protocol that is outside the scope of this document (e.g., [I-D.spiegel-ippm-ioam-rawexport]). Probe packets include a Trace Option which has its Active flag set, indicating that the decapsulating node must terminate them.
- o IOAM active measurement using replicated data packets: probe packets are created by the encapsulating node by selecting some or

all of the en route data packets and replicating them. A selected data packet that is replicated, and its (possibly truncated) copy is forwarded with one or more IOAM option, while the original packet is forwarded normally, without IOAM options. To the extent possible, the original data packet and its replica are forwarded through the same path. The replica includes a Trace Option that has its Active flag set, indicating that the decapsulating node should terminate it. It should be noted that the current document defines the role of the Active flag in allowing the decapsulating node to terminate the packet, but the replication functionality in this context is outside the scope of this document.

If the volume of traffic that incorporates the Active flag is large, it may overload the network and the IOAM node(s) that process the active measurement packet. Thus, the rate of the traffic that includes the Active flag rate SHOULD NOT exceed $1/N$ of the interface capacity on any of the IOAM node's interfaces. It is recommended to use $N > 100$. Depending on the IOAM node's architecture considerations, the rate of Active-enabled IOAM packets may be limited to a lower number in order to avoid loading the IOAM node.

6. IANA Considerations

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

Bit 1 "Loopback" (L-bit)

Bit 2 "Active" (A-bit)

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

7. Performance Considerations

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

Each of the mechanisms that use the flags above has a cost in terms of the network bandwidth, and may potentially load the node that analyzes the data. Therefore, it MUST be possible to use each of the mechanisms on a subset of the data traffic; an encapsulating node needs to be able to set the Loopback and Active flag selectively, in a way that considers the effect on the network performance, as further discussed in Section 4.1.1 and Section 5.

Transit and decapsulating nodes that support Loopback need to be able to limit the looped back packets (Section 4.2) so as to ensure that the mechanisms are used at a rate that does not significantly affect the network bandwidth, and does not overload the source node in the case of loopback.

8. Security Considerations

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

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their effect. This is a fundamental assumption with respect to the security aspects of IOAM, as further discussed in [I-D.ietf-ippm-ioam-data]. However, even given this limited scope, security threats should still be considered and mitigated. Specifically, an attacker may attempt to overload network devices by injecting synthetic packets that include an IOAM Trace Option with one or more of the flags defined in this document. Similarly, an on-path attacker may maliciously set one or more of the flags of transit packets.

- o Loopback flag: an attacker that sets this flag, either in synthetic packets or transit packet, can potentially cause an amplification, since each device along the path creates a copy of the data packet and sends it back to the source. The attacker can potentially leverage the Loopback flag for a Distributed Denial of Service (DDoS) attack, as multiple devices send looped-back copies of a packet to a single source.
- o Active flag: the impact of synthetic packets with the Active flag is no worse than synthetic data packets in which the Active flag is not set. By setting the Active flag in en route packets an attacker can prevent these packets from reaching their destination, since the packet is terminated by the decapsulating device; however, note that an on-path attacker may achieve the same goal by changing the destination address of a packet. Another potential threat is amplification; if an attacker causes transit switches to replicate more packets than they are intended to replicate, either by setting the Active flag or by sending synthetic packets, then traffic is amplified, causing bandwidth degradation. As mentioned in Section 5, the specification of the replication mechanism is not within the scope of this document. A specification that defines the replication functionality should also address the security aspects of this mechanism.

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

In order to mitigate the performance-related attacks described above, as described in Section 7 it should be possible for IOAM-enabled devices to selectively apply the mechanisms that use the flags defined in this document to a subset of the traffic, and to limit the performance of synthetically generated packets to a configurable rate. Specifically, IOAM nodes should be able to:

- o Limit the rate of IOAM packets with the Loopback flag (IOAM encapsulating nodes), as discussed in Section 4.1.1.
- o Limit the rate of looped back packets (IOAM transit and decapsulating nodes), as discussed in Section 4.2.
- o Limit the rate of IOAM packets with the Active flag (IOAM encapsulating nodes), as discussed in Section 5.

As defined in Section 4, transit nodes that process a packet with the Loopback flag only add a single data field, and truncate any payload that follows the IOAM option(s), thus significantly limiting the possible impact of an amplification attack.

9. Acknowledgments

The authors thank Martin Duke, Tommy Pauly, Greg Mirsky, and other members of the IPPM working group for many helpful comments.

10. References

10.1. 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-15 (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>>.

- [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>>.
- [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>>.
- [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>>.

10.2. Informative References

- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-06 (work in progress), July 2021.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-06 (work in progress), July 2021.
- [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.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.

Authors' Addresses

Tal Mizrahi
Huawei
Israel

Email: tal.mizrahi.phd@gmail.com

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

Email: fbrockne@cisco.com

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

Email: shwetha.bhandari@thoughtspot.com

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

Email: sramesh@cisco.com

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

Email: cpignata@cisco.com

Aviv Kfir
Nvidia

Email: avivk@nvidia.com

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

Email: gbarak@nvidia.com

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

Email: mickey.spiegel@intel.com

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

Email: jennifer.lemon@broadcom.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: August 10, 2022

S. Bhandari, Ed.
Thoughtspot
F. Brockners, Ed.
Cisco
February 6, 2022

In-situ OAM IPv6 Options
draft-ietf-ippm-ioam-ipv6-options-07

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 10, 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
2. Contributors	2
3. Conventions	3
3.1. Requirements Language	3
3.2. Abbreviations	3
4. In-situ OAM Metadata Transport in IPv6	3
5. IOAM Deployment In IPv6 Networks	6
5.1. Considerations for IOAM deployment in IPv6 networks	6
5.2. IOAM domains bounded by hosts	7
5.3. IOAM domains bounded by network devices	7
5.4. Deployment options	8
5.4.1. IP-in-IPv6 encapsulation with ULA	8
5.4.2. x-in-IPv6 Encapsulation that is used Independently	8
6. Security Considerations	9
7. IANA Considerations	9
8. Acknowledgements	9
9. References	9
9.1. Normative References	9
9.2. Informative References	10
Contributors' Addresses	11
Authors' Addresses	12

1. Introduction

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

2. 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 Hannes Gredler
- o John Leddy

- o Stephen Youell
- o Tal Mizrahi
- o Aviv Kfir
- o Barak Gafni
- o Petr Lapukhov
- o Mickey Spiegel
- o Suresh Krishnan
- o Rajiv Asati
- o Mark Smith

3. Conventions

3.1. 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.

3.2. Abbreviations

Abbreviations used in this document:

E2E:	Edge-to-Edge
IOAM:	In-situ Operations, Administration, and Maintenance
ION:	IOAM Overlay Network
OAM:	Operations, Administration, and Maintenance
POT:	Proof of Transit

4. In-situ OAM Metadata Transport in IPv6

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

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

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

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

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

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

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

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

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

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

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

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

IOAM Option-Type: IOAM POT Option-Type.

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

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

IOAM Option-Type: IOAM E2E Option-Type.

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

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

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

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

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

5. IOAM Deployment In IPv6 Networks

5.1. Considerations for IOAM deployment in IPv6 networks

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

- C1 It is desirable that the addition of IOAM data fields neither changes the way routers forward packets nor the forwarding decisions the routers take. Packets with added OAM information should follow the same path within the domain that an identical packet without OAM information would follow, even in the presence of ECMP. Such behavior is particularly important for deployments where IOAM data fields are only added "on-demand", e.g., to provide further insights in case of undesired network behavior for certain flows. Implementations of IOAM SHOULD ensure that ECMP behavior for packets with and without IOAM data fields is the same.
- C2 Given that IOAM data fields increase the total size of a packet, the size of a packet including the IOAM data could exceed the PMTU. In particular, the incremental trace IOAM Hop-by-Hop (HbH) Option, which is intended to support hardware implementations of IOAM, changes Option Data Length en-route. Operators of an IOAM domain SHOULD ensure that the addition of OAM information does not lead to fragmentation of the packet, e.g., by configuring the MTU of transit routers and switches to a sufficiently high value. Careful control of the MTU in a network is one of the reasons why IOAM is considered a domain-specific feature (see also

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

- C3 Packets with IOAM data or associated ICMP errors, should not arrive at destinations that have no knowledge of IOAM. For example, if IOAM is used in in transit devices, misleading ICMP errors due to addition and/or presence of OAM data in a packet could confuse the host that sent the packet if it did not insert the OAM information.
- C4 OAM data leaks can affect the forwarding behavior and state of network elements outside an IOAM domain. IOAM domains SHOULD provide a mechanism to prevent data leaks or be able to ensure that if a leak occurs, network elements outside the domain are not affected (i.e., they continue to process other valid packets).
- C5 The source that inserts and leaks the IOAM data needs to be easy to identify for the purpose of troubleshooting, due to the high complexity of troubleshooting a source that inserted the IOAM data and did not remove it when the packet traversed across an Autonomous System (AS). Such a troubleshooting process might require coordination between multiple operators, complex configuration verification, packet capture analysis, etc.
- C6 Compliance with [RFC8200] requires OAM data to be encapsulated instead of header/option insertion directly into in-flight packets using the original IPv6 header.

5.2. IOAM domains bounded by hosts

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

5.3. IOAM domains bounded by network devices

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

5.4. Deployment options

This section lists out possible deployment options that can be employed to meet the requirements listed in Section 5.1.

5.4.1. IP-in-IPv6 encapsulation with ULA

The "IP-in-IPv6 encapsulation with ULA" [RFC4193] approach can be used to apply IOAM to either an IPv6 or an IPv4 network. In addition, it fulfills requirement C4 (avoid leaks) by using ULA for the ION. Similar to the IPv6-in-IPv6 encapsulation approach above, the original IP packet is preserved. An IPv6 header including IOAM data fields in an extension header is added in front of it, to forward traffic within and across the IOAM domain. IPv6 addresses for the ION, i.e. the outer IPv6 addresses are assigned from the ULA space. Addressing and routing in the ION are to be configured so that the IP-in-IPv6 encapsulated packets follow the same path as the original, non-encapsulated packet would have taken. This would create an internal IPv6 forwarding topology using the IOAM domain's interior ULA address space which is parallel with the forwarding topology that exists with the non-IOAM address space (the topology and address space that would be followed by packets that do not have supplemental IOAM information). Establishment and maintenance of the parallel IOAM ULA forwarding topology could be automated, e.g., similar to how LDP [RFC5036] is used in MPLS to establish and maintain an LSP forwarding topology that is parallel to the network's IGP forwarding topology.

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

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

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

is also the IOAM encapsulating node, and the tunnel end point is also the IOAM decapsulating node.

6. Security Considerations

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

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

7. IANA Considerations

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

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

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

8. Acknowledgements

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

9. References

9.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-17 (work in progress), December 2021.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-07 (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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.
- [RFC1772] Rekhter, Y. and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, DOI 10.17487/RFC1772, March 1995, <<https://www.rfc-editor.org/info/rfc1772>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

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

Contributors' Addresses

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States
Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.
Email: hannes@rtbrick.com

John Leddy
Email: john@leddy.net

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom
Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel
Email: tal.mizrahi.phd@gmail.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.
Email: avivk@mellanox.com

Barak Gafni

Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.
Email: gbarak@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US
Email: petr@fb.com

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US
Email: mickey.spiegel@intel.com

Suresh Krishnan
Kaloom
Email: suresh@kaloom.com

Rajiv Asati
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709
US
Email: rajiva@cisco.com

Mark Smith
PO BOX 521
HEIDELBERG, VIC 3084
AU
Email: markzzzzsmith+id@gmail.com

Authors' Addresses

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

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

Email: fbrockne@cisco.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

Philip Eardley
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: philip.eardley@bt.com

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

Email: acmorton@att.com

Aamer Akhter
Consultant
118 Timber Hitch
Cary, NC
USA

Email: 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. Sapia
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.

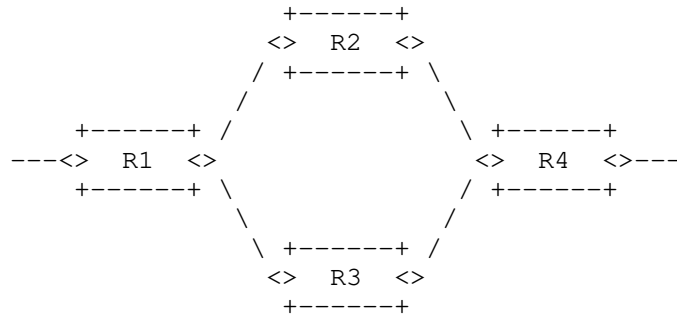
The Alternate Marking method is applicable only to a single path (and partially to a one-to-one multipath), so the extension proposed in this document is suitable also for the most general case of multipoint-to-multipoint, which embraces all the other patterns of Figure 1.

```

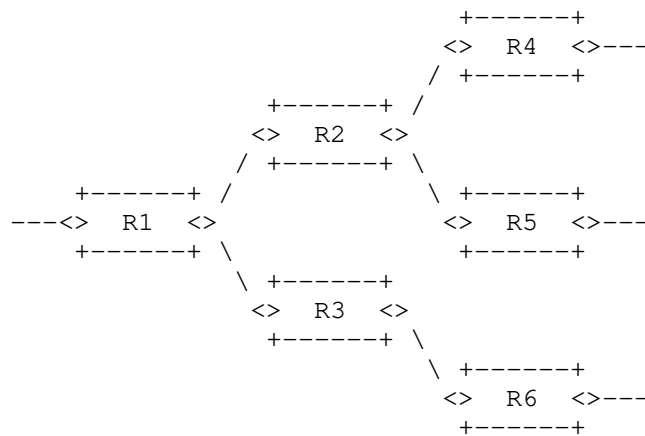
point-to-point single path
  +-----+   +-----+   +-----+
---<>  R1  <>---<>  R2  <>---<>  R3  <>---
  +-----+   +-----+   +-----+

```

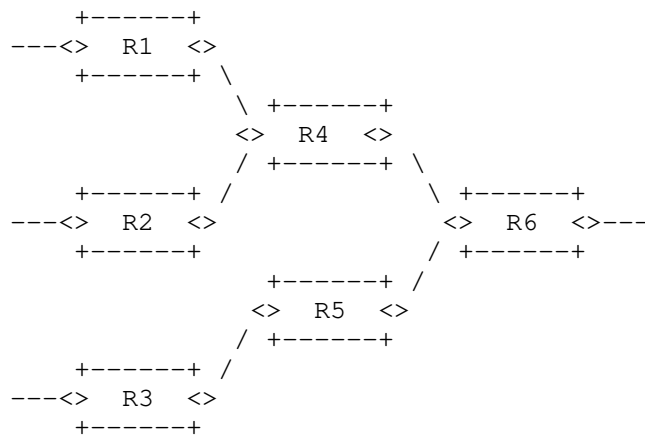
point-to-point multipath



point-to-multipoint



multipoint-to-point



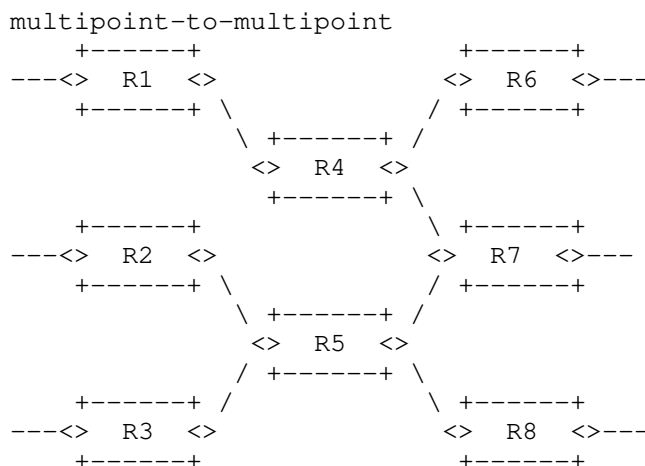


Figure 1: Flow classification

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:

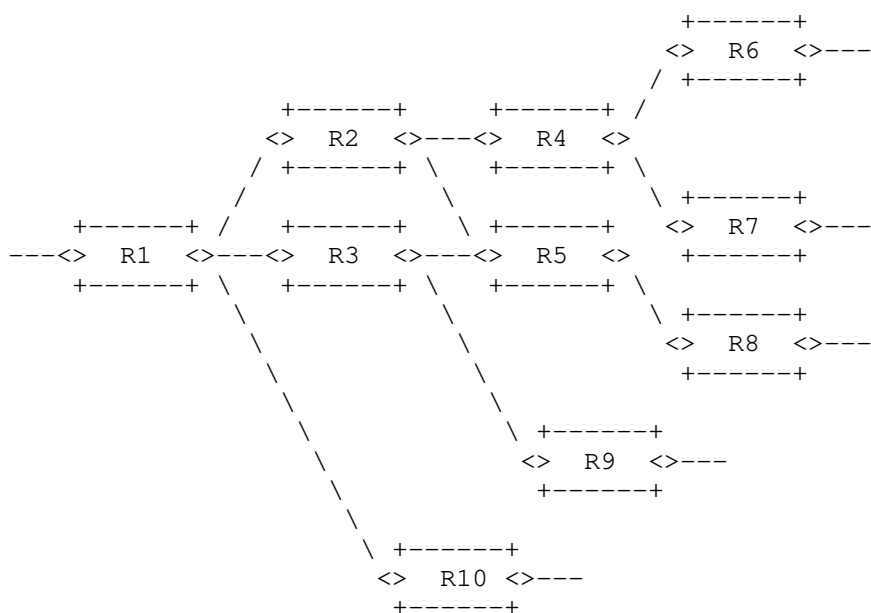


Figure 2: Monitoring Network Graph

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:

$$PL = (PI_1 + PI_2 + \dots + PI_n) - (PO_1 + PO_2 + \dots + PO_m)$$

where:

PL is the Network Packet Loss (number of lost packets)

PI_i is the Number of packets flowed through the i-th Input node in this period

PO_j 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:

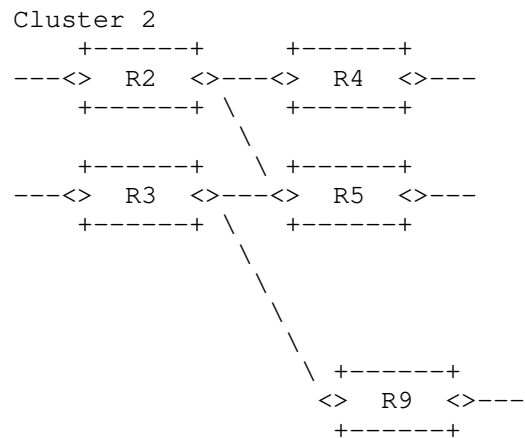
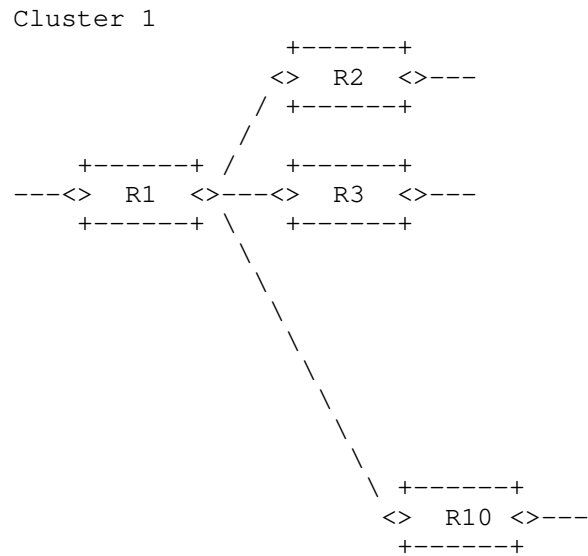
1. Group 1: (R1-R2), (R1-R3), (R1-R10)
2. Group 2: (R2-R4), (R2-R5)
3. Group 3: (R3-R5), (R3-R9)
4. Group 4: (R4-R6), (R4-R7)
5. Group 5: (R5-R8)

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:



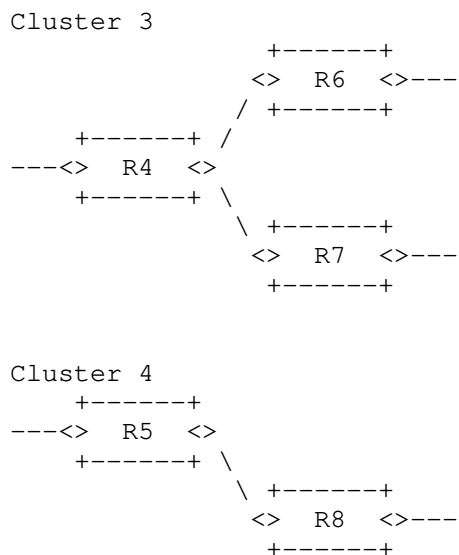


Figure 3: Clusters example

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.

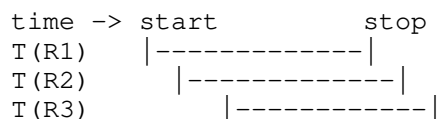


Figure 4: Measurement Interval

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.

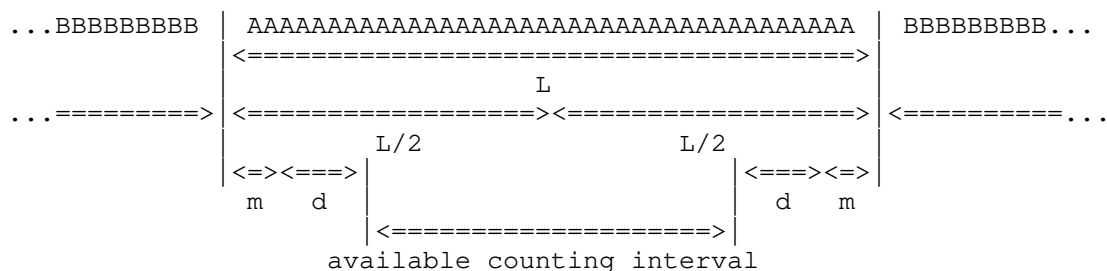


Figure 5: Timing Aspects for Multipoint paths

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

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

Email: mauro.cociglio@telecomitalia.it

Amedeo Sapio
Politecnico di Torino
Corso Duca degli Abruzzi, 24
Torino 10129
Italy

Email: amedeo.sapio@polito.it

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

Email: 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_{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_{max} , the largest value of i needed for a packet to be received at Dst (sent between T_0 and T_f). N_{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.

4.1.2. 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
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

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
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
Updates: 8762 (if approved)
Intended status: Standards Track
Expires: May 19, 2021

G. Mirsky
X. Min
ZTE Corp.
H. Nydell
Accedian Networks
R. Foote
Nokia
A. Masputra
Apple Inc.
E. Ruffini
OutSys
November 15, 2020

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

Abstract

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

Status of This Memo

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

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

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

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

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. Conventions Used in This Document	3
2.1. Acronyms	3
2.2. Requirements Language	3
3. STAMP Test Session Identifier	4
4. TLV Extensions to STAMP	8
4.1. Extra Padding TLV	11
4.2. Location TLV	12
4.2.1. Location Sub-TLVs	13
4.2.2. Theory of Operation of Location TLV	14
4.3. Timestamp Information TLV	16
4.4. Class of Service TLV	17
4.5. Direct Measurement TLV	18
4.6. Access Report TLV	20
4.7. Follow-up Telemetry TLV	21
4.8. HMAC TLV	23
5. IANA Considerations	24
5.1. STAMP TLV Registry	24
5.2. STAMP TLV Flags Sub-registry	25
5.3. Sub-TLV Type Sub-registry	26
5.4. Synchronization Source Sub-registry	26
5.5. Timestamping Method Sub-registry	27
5.6. Return Code Sub-registry	28
6. Security Considerations	29
7. Acknowledgments	29
8. Contributors	30
9. References	30
9.1. Normative References	30
9.2. Informative References	30
Authors' Addresses	31

1. Introduction

Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] defined the STAMP base functionalities. This document specifies the use of optional extensions that use Type-Length-Value (TLV) encoding. Such extensions enhance the STAMP base functions, such as measurement of one-way and round-trip delay, latency, packet loss, packet

duplication, and out-of-order delivery of test packets. This specification defines optional STAMP extensions, their formats, and the theory of operation. Also, a STAMP Test Session Identifier is defined as an update of the base STAMP specification [RFC8762].

2. Conventions Used in This Document

2.1. Acronyms

BDS BeiDou Navigation Satellite System

BITS Building Integrated Timing Supply

CoS Class of Service

DSCP Differentiated Services Code Point

ECN Explicit Congestion Notification

GLONASS Global Orbiting Navigation Satellite System

GPS Global Positioning System [GPS]

HMAC Hashed Message Authentication Code

LORAN-C Long Range Navigation System Version C

MBZ Must Be Zero

NTP Network Time Protocol [RFC5905]

PMF Performance Measurement Function

PTP Precision Time Protocol [IEEE.1588.2008]

TLV Type-Length-Value

SSID STAMP Session Identifier

SSU Synchronization Supply Unit

STAMP Simple Two-way Active Measurement Protocol

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

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

3. STAMP Test Session Identifier

The STAMP Session-Sender transmits test packets to the STAMP Session-Reflector. The STAMP Session-Reflector receives the Session-Sender's packet and acts according to the configuration and optional control information communicated in the Session-Sender's test packet. STAMP defines two different test packet formats, one for packets transmitted by the STAMP Session-Sender and one for packets transmitted by the STAMP Session-Reflector. STAMP supports two modes: unauthenticated and authenticated. Unauthenticated STAMP test packets are compatible on the wire with unauthenticated TWAMP-Test [RFC5357] packets.

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

A STAMP Session is identified by the 4-tuple (source and destination IP addresses, source and destination UDP port numbers). A STAMP Session-Sender MAY generate a locally unique STAMP Session Identifier (SSID). The SSID is a two-octet-long non-zero unsigned integer. SSID generation policy is implementation-specific. [I-D.gont-numeric-ids-generation] thoroughly analyzes common algorithms for identifier generation and their vulnerabilities. For example, an implementation can use algorithms described in Section 7.1 of [I-D.gont-numeric-ids-generation]. An implementation MUST NOT assign the same identifier to different STAMP test sessions. A Session-Sender MAY use the SSID to identify a STAMP test session. If the SSID is used, it MUST be present in each test packet of the given test session. In the unauthenticated mode, the SSID is located as displayed in Figure 1.

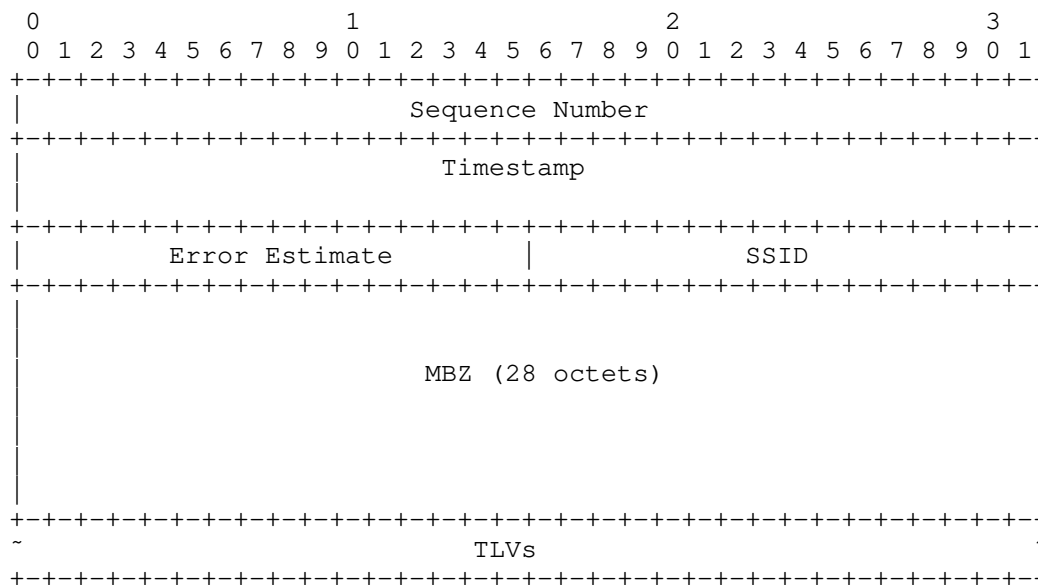


Figure 1: The format of an extended STAMP Session-Sender test packet in unauthenticated mode

An implementation of the STAMP Session-Reflector that supports this specification MUST identify a STAMP Session using the SSID in combination with elements of the usual 4-tuple for the session. Before a test session commences, a Session-Reflector MUST be provisioned with all the elements that identify the STAMP Session. A STAMP Session-Reflector MUST discard non-matching STAMP test packet(s). The means of provisioning the STAMP Session identification is outside the scope of this specification. A conforming implementation of STAMP Session-Reflector MUST copy the SSID value from the received test packet and put it into the reflected packet, as displayed in Figure 2.

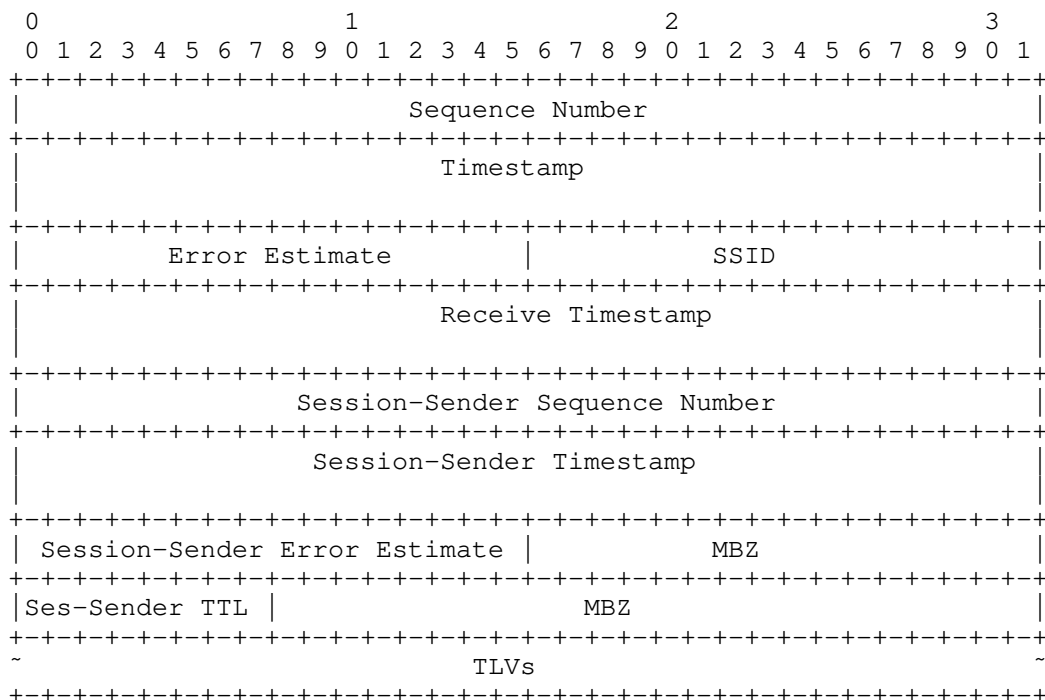


Figure 2: The format of an extended STAMP Session-Reflector test packet in unauthenticated mode

A STAMP Session-Reflector that does not support this specification will return the zeroed SSID field in the reflected STAMP test packet. The Session-Sender MAY stop the session if it receives a zeroed SSID field. An implementation of a Session-Sender MUST support control of its behavior in such a scenario. If the test session is not stopped, the Session-Sender, can, for example, send a base STAMP packet [RFC8762] or continue transmitting STAMP test packets with the SSID.

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

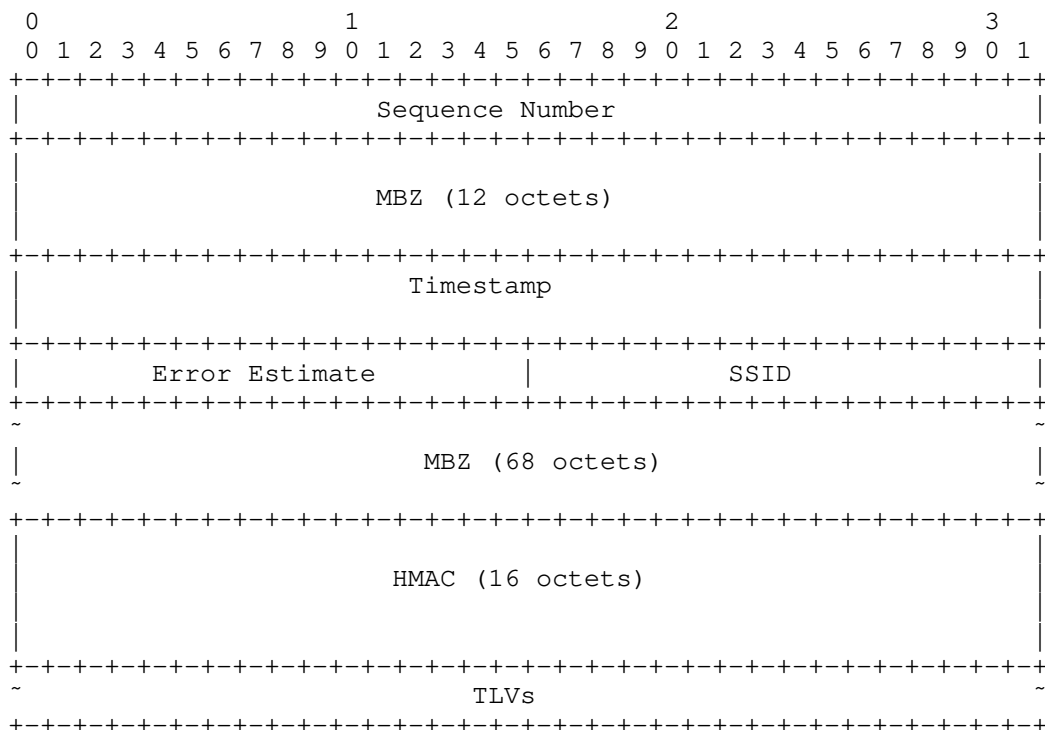
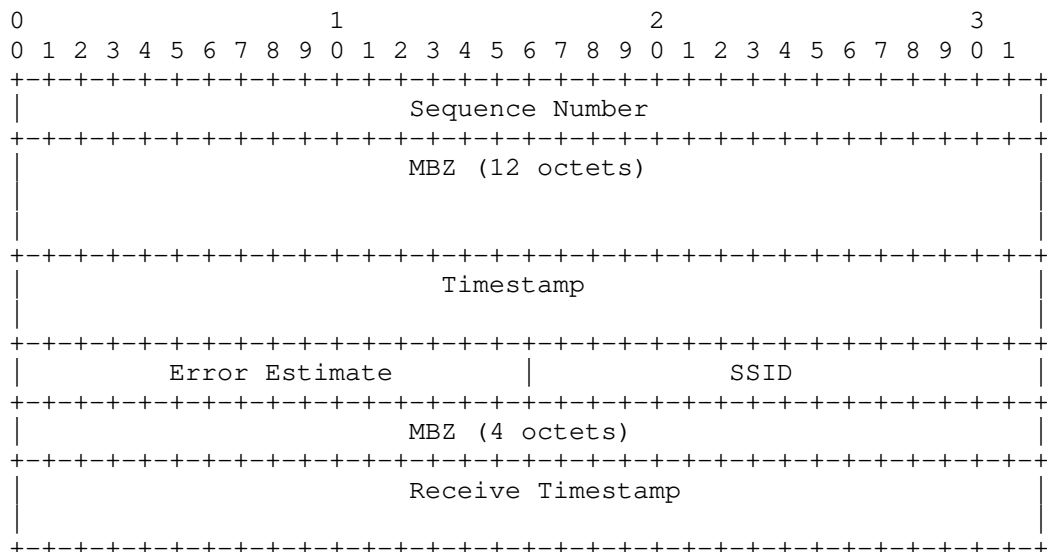


Figure 3: Base STAMP Session-Sender test packet format in authenticated mode



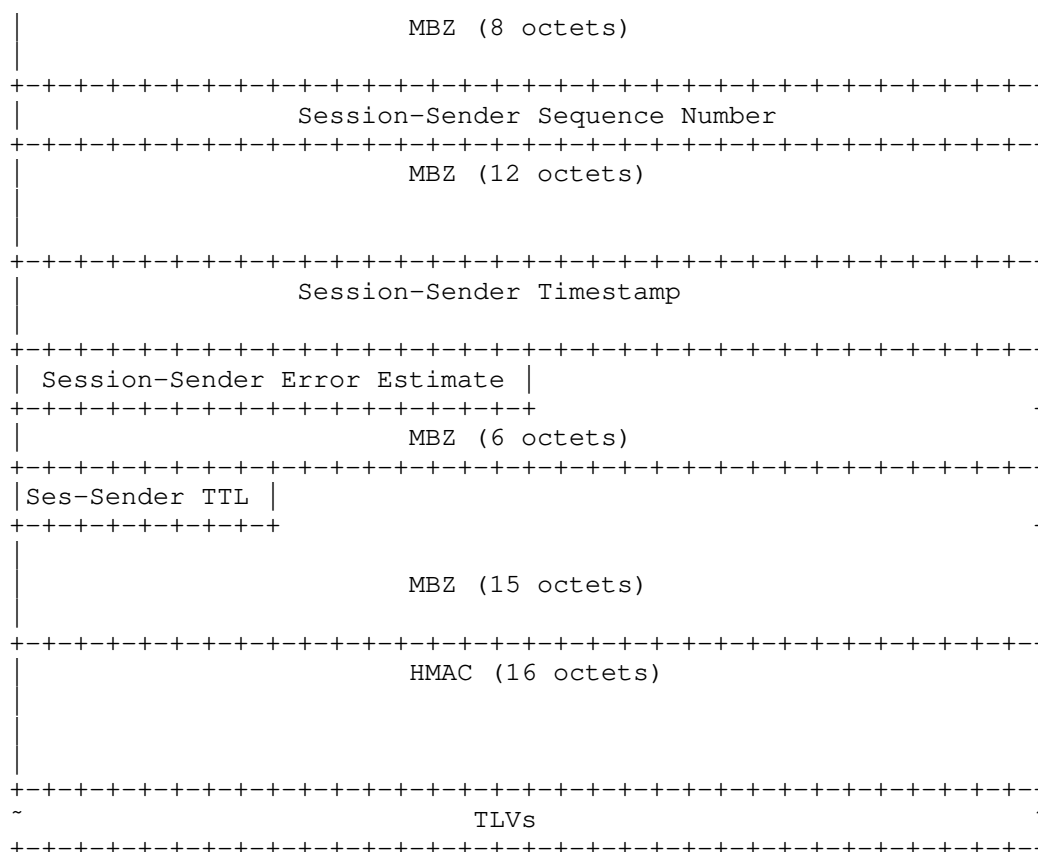


Figure 4: Base STAMP Session-Reflector test packet format in authenticated mode

4. TLV Extensions to STAMP

The Type-Length-Value (TLV) encoding scheme provides a flexible extension mechanism for optional informational elements. TLV is an optional field in the STAMP test packet. Multiple TLVs MAY be placed in a STAMP test packet. Additional TLVs may be enclosed within a given TLV, subject to the semantics of the (outer) TLV in question. TLVs have a one-octet-long STAMP TLV Flags field, a one-octet-long Type field, and a two-octet-long Length field that is equal to the length of the Value field in octets. If a Type value for TLV or sub-TLV is in the range for Vendor Private Use, the Length MUST be at least 4, and the first four octets MUST be that vendor's Structure of Management Information (SMI) Private Enterprise Code, as recorded in IANA's SMI Private Enterprise Codes sub-registry, in network octet

order. The rest of the Value field is private to the vendor. The following sections describe the use of TLVs for STAMP that extend the STAMP capability beyond its base specification.

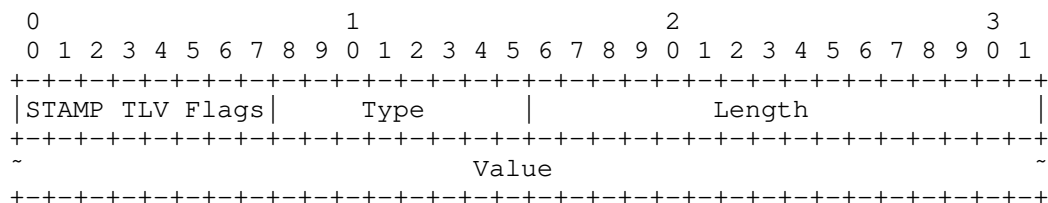


Figure 5: TLV Format in a STAMP Extended Packet

where fields are defined as the following:

- o STAMP TLV Flags - eight-bit-long field. Detailed format and interpretation of flags defined in this specification is below.
- o Type - one-octet-long field that characterizes the interpretation of the Value field. It is allocated by IANA, as specified in Section 5.1.
- o Length - two-octet-long field equal to the length of the Value field in octets.
- o Value - a variable-length field. Its interpretation and encoding is determined by the value of the Type field.

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

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

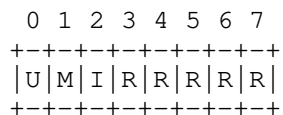


Figure 6: STAMP TLV Flags Format

where fields are defined as the following:

- o U (Unrecognized) is a one-bit flag. A Session-Sender MUST set the U flag to 1 before transmitting an extended STAMP test packet. A Session-Reflector MUST set the U flag to 1 if the Session-

Reflector has not understood the TLV. Otherwise, the Session-Reflector MUST set the U flag in the reflected packet to 0.

- o M (Malformed) is a one-bit flag. A Session-Sender MUST set the M flag to 0 before transmitting an extended STAMP test packet. A Session-Reflector MUST set the M flag to 1 if the Session-Reflector determined the TLV is malformed, i.e., the Length field value is not valid for the particular type, or the remaining length of the extended STAMP packet is less than the size of the TLV. Otherwise, the Session-Reflector MUST set the M flag in the reflected packet to 0.
- o I (Integrity) is a one-bit flag. A Session-Sender MUST set the I flag to 0 before transmitting an extended STAMP test packet. A Session-Reflector MUST set the I flag to 1 if the STAMP extensions have failed HMAC verification (Section 4.8). Otherwise, the Session-Reflector MUST set the I flag in the reflected packet to 0.
- o R - reserved flags for future use. These flags MUST be zeroed on transmit and ignored on receipt.

A STAMP node, whether Session-Sender or Session-Reflector, receiving a test packet MUST determine whether the packet is a base STAMP packet or includes one or more TLVs. The node MUST compare the value in the Length field of the UDP header and the length of the base STAMP test packet in the mode, unauthenticated or authenticated based on the configuration of the particular STAMP test session. If the difference between the two values is larger than the length of the UDP header, then the test packet includes one or more STAMP TLVs that immediately follow the base STAMP test packet. A Session-Reflector that does not support STAMP extensions will not process but copy them into the reflected packet, as defined in Section 4.3 [RFC8762]. A Session-Reflector that supports TLVs will indicate specific TLVs that it did not process by setting the U flag to 1 in those TLVs.

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

If the U flag is set, the STAMP system MUST skip the processing of the TLV.

If the M flag is set, the STAMP system MUST stop processing the remainder of the extended STAMP packet.

If the I flag is set, the STAMP system MUST discard all TLVs and MUST stop processing the remainder of the extended STAMP packet.

If an implementation of a Session-Reflector does not recognize the Type field value, it MUST include a copy of the TLV into the reflected STAMP packet. The Session-Reflector MUST set the U flag to 1. The Session-Reflector MUST skip the processing of the unrecognized TLV.

If a TLV is malformed, the processing of extension TLVs MUST be stopped. The Session-Reflector MUST copy the remainder of the received extended STAMP packet into the reflected STAMP packet. The Session-Reflector MUST set the M flag to 1.

4.1. Extra Padding TLV

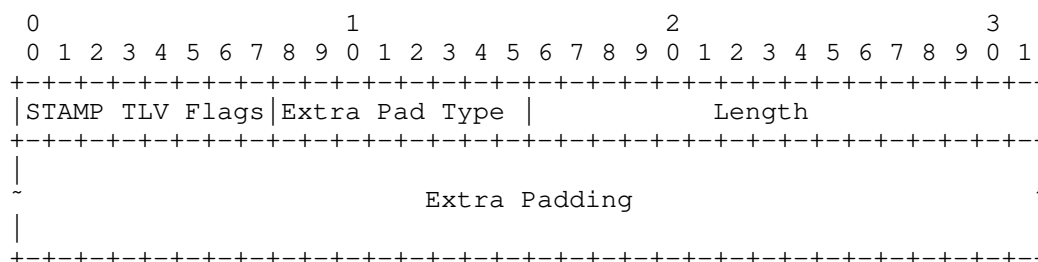


Figure 7: Extra Padding TLV

where fields are defined as the following:

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

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

4.2. Location TLV

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

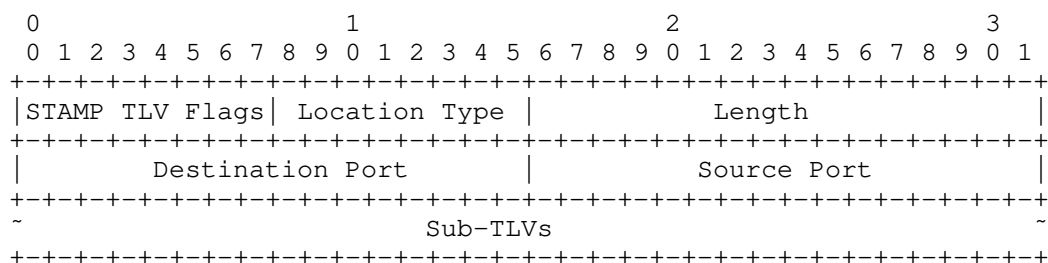


Figure 8: Location TLV

where fields are defined as the following:

- o STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- o Location Type - is a one-octet-long field, value TBA2 allocated by IANA Section 5.1.
- o Length - two-octet-long field equal to the length of the Value field in octets.
- o Destination Port - two-octet-long UDP destination port number of the received STAMP packet.
- o Source Port - two-octet-long UDP source port number of the received STAMP packet.
- o Sub-TLVs - a sequence of sub-TLVs, as defined further in this section. The sub-TLVs are used by the Session-Sender to request location information with generic sub-TLV types, and the Session-Reflector responds with the corresponding more-specific sub-TLVs for the type of address (e.g., IPv4 or IPv6) used at the Session-Reflector.

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

4.2.1. Location Sub-TLVs

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

- o Source MAC Address sub-TLV - is a 12-octet-long sub-TLV. The Type value is TBA9. The value of the Length field MUST equal to 8. The Value field is an 8-octet-long MBZ field that MUST be zeroed on transmission and ignored on receipt.
- o Source EUI-48 Address sub-TLV - is a 12-octet-long sub-TLV that includes the EUI-48 source MAC address. The Type value is TBA10. The value of the Length field MUST equal to 8.

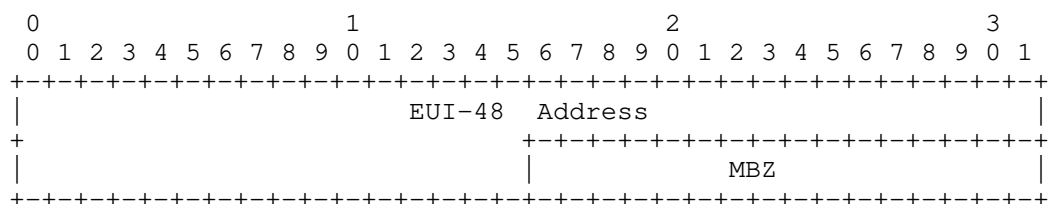


Figure 9: The Value Field of the Source EUI-48 Address sub-TLV

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

- * The EUI-48 is a six-octet-long field.
- * Two-octet-long MBZ field MUST be zeroed on transmission and ignored on receipt.
- o Source EUI-64 Address sub-TLV - is a 12-octet-long sub-TLV that includes the EUI-64 source MAC address. The Type value is TBA11. The value of the Length field MUST equal to 8. The Value field consists of an eight-octet-long EUI-64 field.
- o Destination IP Address sub-TLV - is a 20-octet-long sub-TLV. The Type value is TBA12. The value of the Length field MUST equal to 16. The Value field consists of a 16-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt
- o Destination IPv4 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv4 destination address. The Type value is TBA13. The value of the Length field MUST equal to 16.

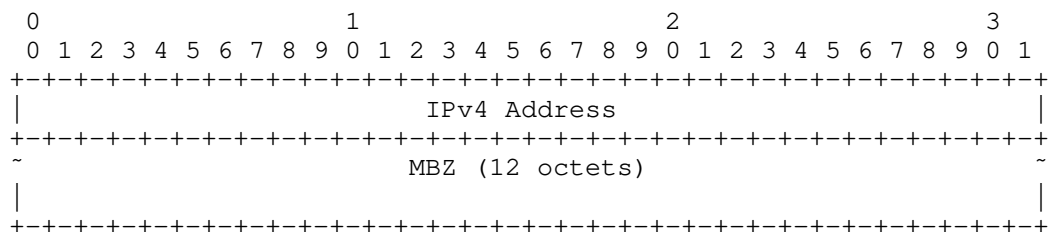


Figure 10: IPv4 Address in a Sub-TLV's Value Field

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

- * The IPv4 Address is a four-octet-long field.
 - * 12-octet-long MBZ field MUST be zeroed on transmit and ignored on receipt.
- o Destination IPv6 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv6 destination address. The Type value is TBA14. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long IP v6 Address field.
 - o Source IP Address sub-TLV - is a 20-octet-long sub-TLV. The Type value is TBA15. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt
 - o Source IPv4 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv4 source address. The Type value is TBA16. The value of the Length field MUST equal to 16. The Value field consists of the following fields (Figure 10):
 - * The IPv4 Address is a four-octet-long field.
 - * 12-octet-long MBZ field that MUST be zeroed on transmit and ignored on receipt.
 - o Source IPv6 Address sub-TLV - is a 20-octet-long sub-TLV that includes IPv6 source address. The Type value is TBA17. The value of the Length field MUST equal to 16. The Value field is a 16-octet-long IPv6 Address field.

4.2.2. Theory of Operation of Location TLV

The Session-Reflector that received an extended STAMP packet with the Location TLV MUST include the Location TLV of the size equal to the size of Location TLV in the received packet in the reflected packet.

Based on the local policy, the Session-Reflector MAY leave some fields unreported by filling them with zeroes. An implementation of the stateful Session-Reflector MUST provide control for managing such policies.

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

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

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

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

4.3. Timestamp Information TLV

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

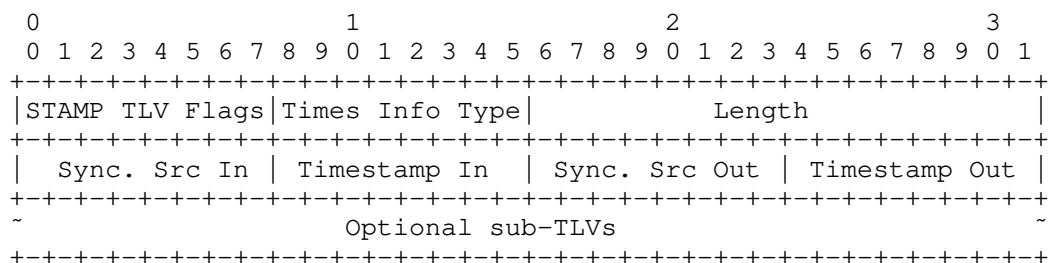


Figure 11: Timestamp Information TLV

where fields are defined as the following:

- o STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- o Timestamp Information Type - is a one-octet-long field, value TBA3 allocated by IANA Section 5.1.
- o Length - two-octet-long field, set equal to the length of the Value field in octets (Figure 5).
- o Sync Src In - one-octet-long field that characterizes the source of clock synchronization at the ingress of a Session-Reflector. There are several methods to synchronize the clock, e.g., Network Time Protocol (NTP) [RFC5905]. The value is one of those listed in Table 7.
- o Timestamp In - one-octet-long field that characterizes the method by which the ingress of the Session-Reflector obtained the timestamp T2. A timestamp may be obtained with hardware assistance, via software API from a local wall clock, or from a remote clock (the latter is referred to as "control plane"). The value is one of those listed in Table 9.

- o Sync Src Out - one-octet-long field that characterizes the source of clock synchronization at the egress of the Session-Reflector. The value is one of those listed in Table 7.
- o Timestamp Out - one-octet-long field that characterizes the method by which the egress of the Session-Reflector obtained the timestamp T3. The value is one of those listed in Table 9.
- o Optional sub-TLVs - optional variable-length field.

4.4. Class of Service TLV

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

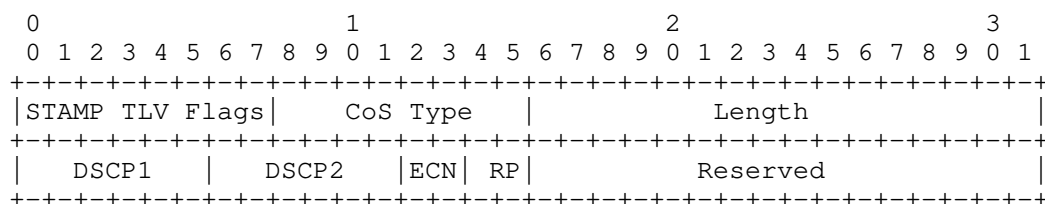


Figure 12: Class of Service TLV

where fields are defined as the following:

- o STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- o CoS (Class of Service) Type - is a one-octet-long field, value TBA4 allocated by IANA Section 5.1.
- o Length - two-octet-long field, set equal to the value 4.
- o DSCP1 - The Differentiated Services Code Point (DSCP) intended by the Session-Sender to be used as the DSCP value of the reflected test packet.
- o DSCP2 - The received value in the DSCP field at the ingress of the Session-Reflector.
- o ECN - The received value in the ECN field at the ingress of the Session-Reflector.
- o RP (Reverse Path) - is a two-bit-long field. A Session-Sender MUST set the value of the RP field to 0 on transmission.

- o Reserved - 16-bit-long field, MUST be zeroed on transmission and ignored on receipt.

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

Re-mapping of CoS can be used to provide multiple services (e.g., 2G, 3G, LTE in mobile backhaul networks) over the same network. But if it is misconfigured, then it is often difficult to diagnose the root cause of excessive packet drops of higher-level service while packet drops for lower service packets are at a normal level. Using a CoS TLV in STAMP testing helps to troubleshoot the existing problem and also verify whether DiffServ policies are processing CoS as required by the configuration.

4.5. Direct Measurement TLV

The Direct Measurement TLV enables collection of the number of in-profile packets, i.e., packets that form a specific data flow, that had been transmitted and received by the Session-Sender and Session-Reflector, respectively. The definition of "in-profile packet" is outside the scope of this document and is left to the test operators to determine.

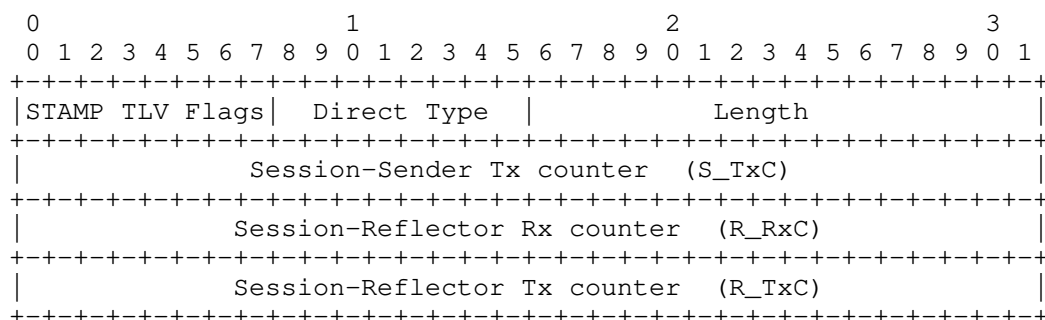


Figure 13: Direct Measurement TLV

where fields are defined as the following:

- o STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- o Direct (Measurement) Type - is a one-octet-long field, value TBA5 allocated by IANA Section 5.1.
- o Length - two-octet-long field equals the length of the Value field in octets. The Length field value MUST equal 12 octets.
- o Session-Sender Tx counter (S_TxC) is a four-octet-long field. The Session-Sender MUST set its value equal to the number of the transmitted in-profile packets.
- o Session-Reflector Rx counter (R_RxC) is a four-octet-long field. MUST be zeroed by the Session-Sender on transmit and ignored by the Session-Reflector on receipt. The Session-Reflector MUST fill it with the value of in-profile packets received.
- o Session-Reflector Tx counter (R_TxC) is a four-octet-long field. MUST be zeroed by the Session-Sender and ignored by the Session-Reflector on receipt. The Session-Reflector MUST fill it with the value of the transmitted in-profile packets.

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

4.6. Access Report TLV

A STAMP Session-Sender MAY include an Access Report TLV (Figure 14) to indicate changes to the access network status to the Session-Reflector. The definition of an access network is outside the scope of this document.

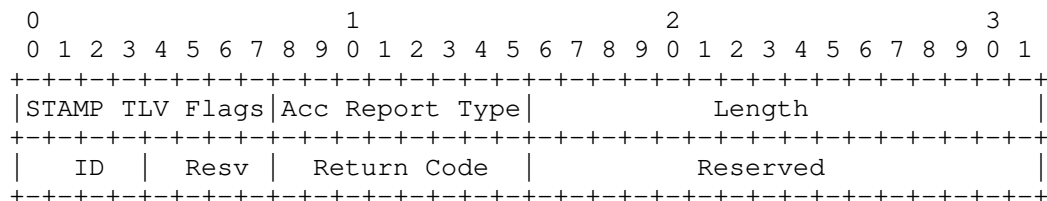


Figure 14: Access Report TLV

where fields are defined as follows:

- o STAMP TLV Flags - is an eight-bit-long field. Its format presented in Figure 6.
- o Access Report Type - is a one-octet-long field, value TBA6 allocated by IANA Section 5.1.
- o Length - two-octet-long field, set equal to the value 4.
- o ID (Access ID) - four-bit-long field that identifies the access network, e.g., 3GPP (Radio Access Technologies specified by 3GPP) or Non-3GPP (accesses that are not specified by 3GPP) [TS23501]. The value is one of those listed below:
 - * 1 - 3GPP Network
 - * 2 - Non-3GPP Network

All other values are invalid and the TLV that contains it MUST be discarded.
- o Resv - four-bit-long field, MUST be zeroed on transmission and ignored on receipt.
- o Return Code - one-octet-long field that identifies the report signal, e.g., available or unavailable. The value is supplied to the STAMP end-point through some mechanism that is outside the scope of this document. The value is one of those listed in Section 5.6.

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

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

The Session-Sender MUST also arm a retransmission timer after sending a test packet that includes the Access Report TLV. This timer MUST be disarmed upon reception of the reflected STAMP test packet that includes the Access Report TLV. In the event the timer expires before such a packet is received, the Session-Sender MUST retransmit the STAMP test packet that contains the Access Report TLV. This retransmission SHOULD be repeated up to four times before the procedure is aborted. Setting the value for the retransmission timer is based on local policies and network environment. The default value of the retransmission timer for the Access Report TLV SHOULD be three seconds. An implementation MUST provide control of the retransmission timer value and the number of retransmissions.

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

4.7. Follow-up Telemetry TLV

A Session-Reflector might be able to put in the Timestamp field only an "SW Local" (see Table 9) timestamp. But the hosting system might provide a timestamp closer to the start of the actual packet transmission even though it is not possible to deliver the information to the Session-Sender in time for the packet itself. This timestamp might nevertheless be important for the Session-Sender, as it improves the accuracy of measuring network delay by minimizing the impact of egress queuing delays on the measurement.

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

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

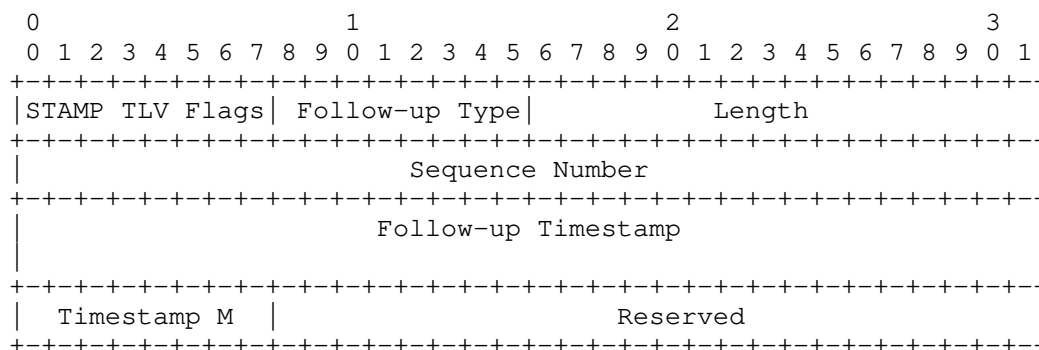


Figure 15: Follow-up Telemetry TLV

where fields are defined as follows:

- o STAMP TLV Flags - is an eight-bit-long field. Its format presented in Figure 6.
- o Follow-up (Telemetry) Type - is a one-octet-long field, value TBA7 allocated by IANA Section 5.1.
- o Length - two-octet-long field, set equal to the value 16 octets.
- o Sequence Number - four-octet-long field indicating the sequence number of the last packet reflected in the same STAMP-test session. Since the Session-Reflector runs in the stateful mode (defined in Section 4.2 [RFC8762]), it is the Session-Reflector's Sequence Number of the previous reflected packet.
- o Follow-up Timestamp - eight-octet-long field, with the format indicated by the Z flag of the Error Estimate field of the STAMP base packet, which is contained in this reflected test packet transmitted by a Session-Reflector, as described in Section 4.2.1 [RFC8762]. It carries the timestamp when the reflected packet with the specified sequence number was sent.

- o Timestamp M(ode) - one-octet-long field that characterizes the method by which the entity that transmits a reflected STAMP packet obtained the Follow-up Timestamp. The value is one of those listed in Table 9.
- o Reserved - three-octet-long field. Its value MUST be zeroed on transmission and ignored on receipt.

4.8. HMAC TLV

The STAMP authenticated mode protects the integrity of data collected in the STAMP base packet. STAMP extensions are designed to provide valuable information about the condition of a network, and protecting the integrity of that data is also essential. All authenticated STAMP base packets (per Section 4.2.2 and Section 4.3.2 [RFC8762]) compatible with this specification MUST additionally authenticate the option TLVs by including the keyed Hashed Message Authentication Code (HMAC) TLV, with the sole exception of when there is only one TLV present, and it is the Extended Padding TLV. The HMAC TLV MUST follow all TLVs included in a STAMP test packet, except for the Extra Padding TLV. If the HMAC TLV appears in any other position in a STAMP extended test packet, then the situation MUST be processed as HMAC verification failure, as defined in this section, further below. The HMAC TLV MAY be used to protect the integrity of STAMP extensions in STAMP unauthenticated mode. An implementation of STAMP extensions MUST provide controls to enable the integrity protection of STAMP extensions in STAMP unauthenticated mode.

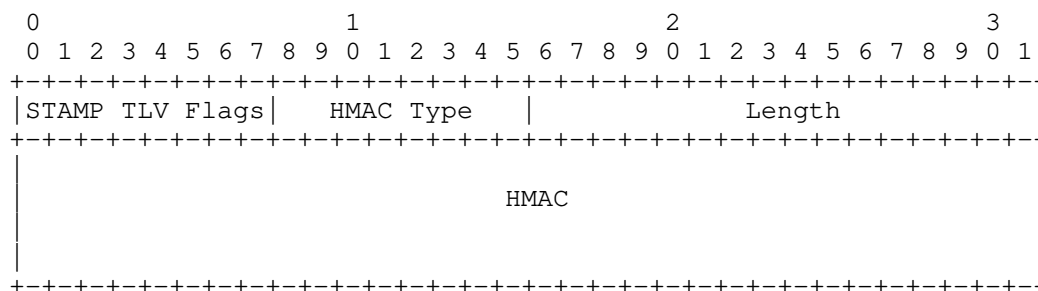


Figure 16: HMAC TLV

where fields are defined as follows:

- o STAMP TLV Flags - is an eight-bit-long field. Its format is presented in Figure 6.
- o HMAC Type - is a one-octet-long field, value TBA8 allocated by IANA Section 5.1.

- o Length - two-octet-long field, set equal to 16 octets.
- o HMAC - is a 16-octet-long field that carries HMAC digest of the text of all preceding TLVs.

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

5. IANA Considerations

5.1. STAMP TLV Registry

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

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

Table 1: STAMP TLV Type Registry

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

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

Table 2: STAMP TLV Types

5.2. STAMP TLV Flags Sub-registry

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

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

Table 3: STAMP TLV Flags

5.3. Sub-TLV Type Sub-registry

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

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

Table 4: Location Sub-TLV Type Sub-registry

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

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

Table 5: STAMP sub-TLV Types

5.4. Synchronization Source Sub-registry

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

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

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

Table 6: Synchronization Source Sub-registry

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

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

Table 7: Synchronization Sources

5.5. Timestamping Method Sub-registry

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

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

Table 8: Timestamping Method Sub-registry

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

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

Table 9: Timestamping Methods

5.6. Return Code Sub-registry

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

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

Table 10: Return Code Sub-registry

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

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

Table 11: Return Codes

6. Security Considerations

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

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

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

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

7. Acknowledgments

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

8. Contributors

The following people contributed text to this document:

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

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

9. References

9.1. Normative References

- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <<https://www.rfc-editor.org/info/rfc8762>>.

9.2. Informative References

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

- [I-D.gont-numeric-ids-generation]
Gont, F. and I. Arce, "On the Generation of Transient Numeric Identifiers", draft-gont-numeric-ids-generation-04 (work in progress), July 2019.
- [IEEE.1588.2008]
"Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Standard 1588, March 2008.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <<https://www.rfc-editor.org/info/rfc4868>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [TS23501] 3GPP (3rd Generation Partnership Project), "Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2 (Release 16)", 3GPP TS23501, 2019.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

Henrik Nydell
Accedian Networks

Email: hnydell@accedian.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

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

Email: adi@apple.com

Ernesto Ruffini
OutSys
via Caracciolo, 65
Milano 20155
Italy

Email: eruffini@outsys.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 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].

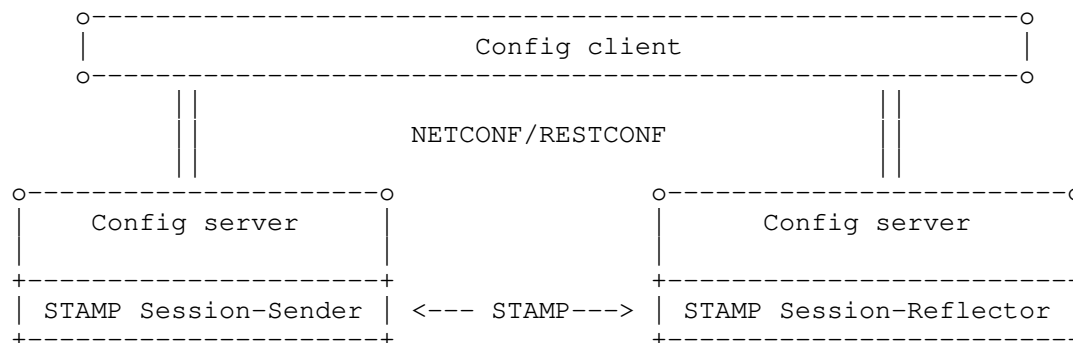


Figure 1: STAMP Reference Model

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

Editor: Greg Mirsky
gregimirsky@gmail.com

Editor: Xiao Min
xiao.min2@zte.com.cn

Editor: Wei S Luo
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

IPPM
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2020

H. Song
Futurewei
B. Gafni
Mellanox Technologies, Inc.
T. Zhou
Z. Li
Huawei
F. Brockners
S. Bhandari
R. Sivakolundu
Cisco
T. Mizrahi, Ed.
Huawei Smart Platforms iLab
October 12, 2019

In-situ OAM Direct Exporting
draft-ioamteam-ippm-ioam-direct-export-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 14, 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	3
2.1. Requirement Language	3
2.2. Terminology	3
3. The Direct Exporting (DEX) IOAM Option Type	3
3.1. Overview	3
3.2. The DEX Option Format	5
4. IANA Considerations	6
4.1. IOAM Type	6
4.2. IOAM DEX Flags	6
5. Performance Considerations	7
6. Security Considerations	7
7. Topics for Further Discussion	7
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Authors' Addresses	9

1. Introduction

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

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

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

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

2. Conventions

2.1. Requirement 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].

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

DEX: Direct EXporting

3. The Direct Exporting (DEX) IOAM Option Type

3.1. Overview

The DEX option is used as a trigger for exporting telemetry data to a collector.

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

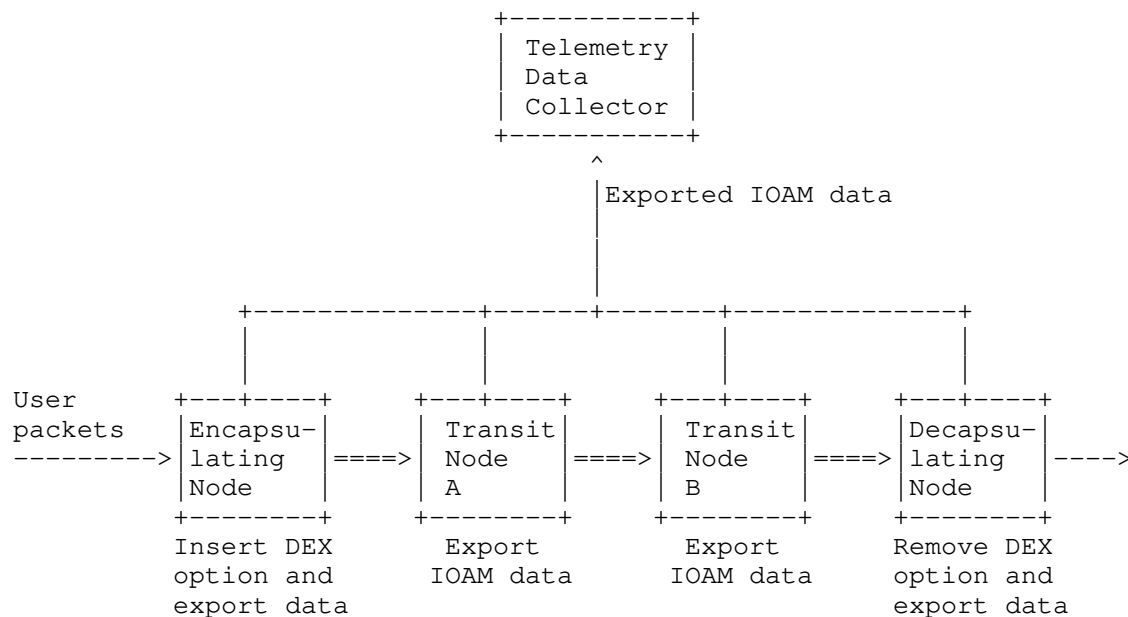


Figure 1: DEX Architecture

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

- o An IOAM encapsulating node configured to incorporate the DEX option encapsulates the packet with the DEX option, and exports the requested IOAM data immediately. The IOAM encapsulating node is the only type of node allowed to push the DEX option.
- o A transit node that processes a packet with the DEX option is expected to export the requested IOAM data.
- o An IOAM decapsulating node that processes a packet with the DEX option is expected to export the requested IOAM data, and decapsulate the IOAM header.

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

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

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

3.2. The DEX Option Format

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

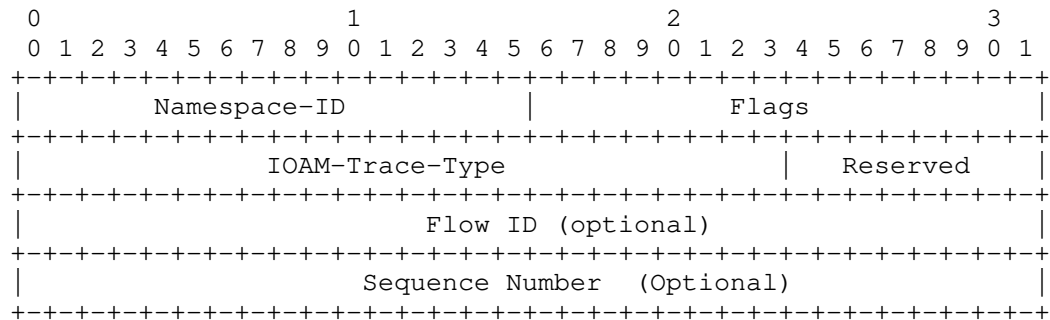


Figure 2: DEX Option Format

Namespace-ID	A 16-bit identifier of the IOAM namespace, as defined in [I-D.ietf-ippm-ioam-data].
Flags	A 16-bit field, comprised of 16 one-bit subfields. Flags are allocated by IANA, as defined in Section 4.2.
IOAM-Trace-Type	A 24-bit identifier which specifies which data fields should be exported. The format of this field is as defined in [I-D.ietf-ippm-ioam-data]. Specifically, bit 23, which corresponds to the Checksum Complement data field, should be assigned to be zero by the IOAM

encapsulating node, and ignored by transit and decapsulating nodes. The reason for this is that the Checksum Complement is intended for in-flight packet modifications and is not relevant for direct exporting.

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

4. IANA Considerations

4.1. IOAM Type

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

TBD-type IOAM Direct Export (DEX) Option Type

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

4.2. IOAM DEX Flags

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

5. Performance Considerations

The DEX option triggers exported packets to be exported to a collector, which in some cases may impact the collector's performance, or the performance along the paths leading to the collector.

Therefore, rate limiting may be enabled so as to ensure that direct exporting is used at a rate that does not significantly affect the network bandwidth, and does not overload the collector (or the source node in the case of loopback). It should be possible to use each DEX on a subset of the data traffic.

6. Security Considerations

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

An attacker may attempt to overload network devices by injecting synthetic packets that include the DEX option. Similarly, an on-path attacker may maliciously incorporate the DEX option into transit packets.

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

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

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

7. Topics for Further Discussion

- o Hop Limit / Hop Count: in order to help correlate and order the exported packets, it is possible to include a 1-octet Hop Count field in the DEX header (presumably by claiming some space from the Flags field). Its value starts from 0 at the encapsulating node and is incremented by each IOAM transit node that supports the DEX option. The Hop Count field value is also included in the

exported packet. An alternative approach is to use the Hop_Lim/Node_ID data field; if the IOAM-Trace-Type [I-D.ietf-ippm-ioam-data] has the Hop_Lim/Node_ID bit set, then exported packets include the Hop_Lim/Node_ID data field, which contains the TTL/Hop Limit value from a lower layer protocol. The main advantage of the Hop_Lim/Node_ID approach is that it provides information about the current hop count without requiring each transit node to modify the DEX option, thus simplifying the data plane functionality of Direct Exporting. The main advantage of the Hop Count approach is that it counts the number of IOAM-capable nodes without relying on the lower layer TTL, especially when the lower layer cannot provide the accurate TTL information, e.g., Layer 2 Ethernet or hierarchical VPN. It also explicitly allows to detect a case where an IOAM-capable node fails to export packets to the collector. In order to facilitate the Hop Count approach it is possible to use a flag to indicate an optional Hop Count field, which enables to control the tradeoff. On one hand it addresses the use cases that the Hop_Lim/Node_ID cannot cover, and on the other hand it does not require transit switches to update the option if it is not supported or disabled. Further discussion is required about the tradeoff between the two alternatives.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-07 (work in progress), September 2019.

[I-D.mizrahi-ippm-ioam-flags]

Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-mizrahi-ippm-ioam-flags-00 (work in progress), July 2019.

[I-D.song-ippm-postcard-based-telemetry]

Song, H., Zhou, T., Li, Z., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry", draft-song-ippm-postcard-based-telemetry-05 (work in progress), September 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>>.

8.2. Informative References

- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-02 (work in progress), September 2019.
- [I-D.ioametal-ippm-6man-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., and R. Asati, "In-situ OAM IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-options-02 (work in progress), March 2019.
- [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-02 (work in progress), July 2019.
- [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>>.

Authors' Addresses

Haoyu Song
Futurewei
2330 Central Expressway
Santa Clara 95050
USA

Email: haoyu.song@huawei.com

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

Email: gbarak@mellanox.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Zhenbin Li
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

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

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

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

Email: sramesh@cisco.com

Tal Mizrahi (editor)
Huawei Smart Platforms iLab
Israel

Email: tal.mizrahi.phd@gmail.com

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: 27 October 2022

G. Mirsky
Ericsson
W. Lingqiang
G. Zhui
ZTE Corporation
H. Song
Futurewei Technologies
P. Thubert
Cisco Systems, Inc
25 April 2022

Hybrid Two-Step Performance Measurement Method
draft-mirsky-ippm-hybrid-two-step-13

Abstract

Development of, and advancements in, automation of network operations brought new requirements for measurement methodology. Among them is the ability to collect instant network state as the packet being processed by the networking elements along its path through the domain. This document introduces a new hybrid measurement method, referred to as hybrid two-step, as it separates the act of measuring and/or calculating the performance metric from the act of collecting and transporting network state.

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 27 October 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	2
2. Conventions used in this document	3
2.1. Acronyms	3
2.2. Requirements Language	4
3. Problem Overview	4
4. Theory of Operation	5
4.1. Operation of the HTS Ingress Node	7
4.2. Operation of the HTS Intermediate Node	9
4.3. Operation of the HTS Egress Node	10
4.4. Considerations for HTS Timers	11
4.5. Deploying HTS in a Multicast Network	11
5. Authentication in HTS	12
6. IANA Considerations	13
6.1. IOAM Option-Type for HTS	13
6.2. HTS TLV Registry	13
6.3. HTS Sub-TLV Type Sub-registry	14
6.4. HMAC Type Sub-registry	15
7. Security Considerations	16
8. Acknowledgments	16
9. References	16
9.1. Normative References	16
9.2. Informative References	17
Authors' Addresses	19

1. Introduction

Successful resolution of challenges of automated network operation, as part of, for example, overall service orchestration or data center operation, relies on a timely collection of accurate information that reflects the state of network elements on an unprecedented scale. Because performing the analysis and act upon the collected information requires considerable computing and storage resources, the network state information is unlikely to be processed by the network elements themselves but will be relayed into the data storage facilities, e.g., data lakes. The process of producing, collecting network state information also referred to in this document as network telemetry, and transporting it for post-processing should

work equally well with data flows or injected in the network test packets. RFC 7799 [RFC7799] describes a combination of elements of passive and active measurement as a hybrid measurement.

Several technical methods have been proposed to enable the collection of network state information instantaneous to the packet processing, among them [P4.INT] and [I-D.ietf-ippm-ioam-data]. The instantaneous, i.e., in the data packet itself, collection of telemetry information simplifies the process of attribution of telemetry information to the particular monitored flow. On the other hand, this collection method impacts the data packets, potentially changing their treatment by the networking nodes. Also, the amount of information the instantaneous method collects might be incomplete because of the limited space it can be allotted. Other proposals defined methods to collect telemetry information in a separate packet from each node traversed by the monitored data flow. Examples of this approach to collecting telemetry information are [I-D.ietf-ippm-ioam-direct-export] and [I-D.song-ippm-postcard-based-telemetry]. These methods allow data collection from any arbitrary path and avoid directly impacting data packets. On the other hand, the correlation of data and the monitored flow requires that each packet with telemetry information also includes characteristic information about the monitored flow.

This document introduces Hybrid Two-Step (HTS) as a new method of telemetry collection that improves accuracy of a measurement by separating the act of measuring or calculating the performance metric from the collecting and transporting this information while minimizing the overhead of the generated load in a network. HTS method extends the two-step mode of Residence Time Measurement (RTM) defined in [RFC8169] to on-path network state collection and transport. HTS allows the collection of telemetry information from any arbitrary path, does not change data packets of the monitored flow and makes the process of attribution of telemetry to the data flow simple.

2. Conventions used in this document

2.1. Acronyms

RTM Residence Time Measurement

ECMP Equal Cost Multipath

MTU Maximum Transmission Unit

HTS Hybrid Two-Step

HMAC Hashed Message Authentication Code

Network telemetry - the process of collecting and reporting of network state

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. Problem Overview

Performance measurements are meant to provide data that characterize conditions experienced by traffic flows in the network and possibly trigger operational changes (e.g., re-route of flows, or changes in resource allocations). Modifications to a network are determined based on the performance metric information available when a change is to be made. The correctness of this determination is based on the quality of the collected metrics data. The quality of collected measurement data is defined by:

- * the resolution and accuracy of each measurement;
- * predictability of both the time at which each measurement is made and the timeliness of measurement collection data delivery for use.

Consider the case of delay measurement that relies on collecting time of packet arrival at the ingress interface and time of the packet transmission at the egress interface. The method includes recording a local clock value on receiving the first octet of an affected message at the device ingress, and again recording the clock value on transmitting the first byte of the same message at the device egress. In this ideal case, the difference between the two recorded clock times corresponds to the time that the message spent in traversing the device. In practice, the time recorded can differ from the ideal case by any fixed amount. A correction can be applied to compute the same time difference taking into account the known fixed time associated with the actual measurement. In this way, the resulting time difference reflects any variable delay associated with queuing.

Depending on the implementation, it may be a challenge to compute the difference between message arrival and departure times and - on the fly - add the necessary residence time information to the same message. And that task may become even more challenging if the

packet is encrypted. Recording the departure of a packet time in the same packet may be detrimental to the accuracy of the measurement because the departure time includes the variable time component (such as that associated with buffering and queuing of the packet). A similar problem may lower the quality of, for example, information that characterizes utilization of the egress interface. If unable to obtain the data consistently, without variable delays for additional processing, information may not accurately reflect the egress interface state. To mitigate this problem [RFC8169] defined an RTM two-step mode.

Another challenge associated with methods that collect network state information into the actual data packet is the risk to exceed the Maximum Transmission Unit (MTU) size on the path, especially if the packet traverses overlay domains or VPNs. Since the fragmentation is not available at the transport network, operators may have to reduce MTU size advertised to the client layer or risk missing network state data for the part, most probably the latter part, of the path.

In some networks, for example, wireless that are in the scope of [I-D.ietf-raw-use-cases], it is beneficial to collect the telemetry, including the calculated performance metrics, that reflects conditions experienced by the monitored flow at a node, other than the egress. For example, a head-end can optimize path selection based on the compounded information that reflects network conditions, resource utilization. This mode is referred to as the upstream collection and the other - downstream collection to differentiate between two modes of telemetry collection.

4. Theory of Operation

The HTS method consists of two phases:

- * performing a measurement and/or obtaining network state information on a node;
- * collecting and transporting the measurement and/or the telemetry information.

HTS may use an HTS Trigger carried in a data packet or a specially constructed test packet. For example, an HTS Trigger could be a packet that has IOAM Option-Type set to the "IOAM Hybrid Two-Step Option-Type" value (TBA1) allocated by IANA (see Section 6.1). The HTS Trigger also includes IOAM Namespace-ID and IOAM-Trace-Type information s defined in Section 5.3 and Section 5.4.1 [I-D.ietf-ippm-ioam-data] respectively (shown in Figure 1). A packet in the flow to which the Alternate-Marking method, defined in [RFC8321] and [RFC8889], is applied can be used as an HTS Trigger.

The nature of the HTS Trigger is a transport network layer-specific, and its description is outside the scope of this document. The packet that includes the HTS Trigger in this document is also referred to as the trigger packet.

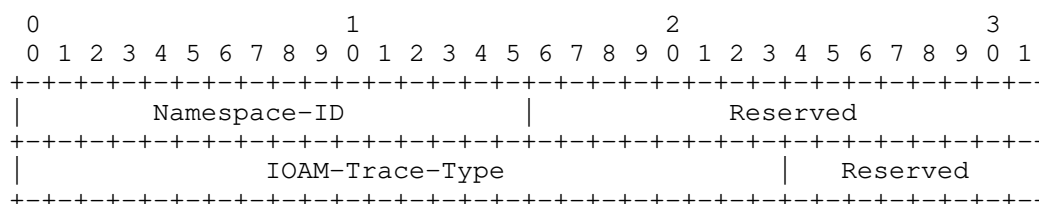


Figure 1: Hybrid Two-Step Trace IOAM Header

The HTS method uses the HTS Follow-up packet, referred to as the follow-up packet, to collect measurement and network state data from the nodes. The node that creates the HTS Trigger also generates the HTS Follow-up packet. In some use cases, e.g., when HTS is used to collect the telemetry, including performance metrics, calculated based on a series of measurements, an HTS follow-up packet can be originated without using the HTS Trigger. The follow-up packet contains characteristic information sufficient for participating HTS nodes to associate it with the monitored data flow. The characteristic information can be obtained using the information of the trigger packet or constructed by a node that originates the follow-up packet. As the follow-up packet is expected to traverse the same sequence of nodes, one element of the characteristic information is the information that determines the path in the data plane. For example, in a segment routing domain [RFC8402], a list of segment identifiers of the trigger packet is applied to the follow-up packet. And in the case of the service function chain based on the Network Service Header [RFC8300], the Base Header and Service Path Header of the trigger packet will be applied to the follow-up packet. Also, when HTS is used to collect the telemetry information in an IOAM domain, the IOAM trace option header [I-D.ietf-ippm-ioam-data] of the trigger packet is applied in the follow-up packet. The follow-up packet also uses the same network information used to load-balance flows in equal-cost multipath (ECMP) as the trigger packet, e.g., IPv6 Flow Label [RFC6437] or an entropy label [RFC6790]. The exact composition of the characteristic information is specific for each transport network, and its definition is outside the scope of this document.

Only one outstanding follow-up packet MUST be on the node for the given path. That means that if the node receives an HTS Trigger for the flow on which it still waits for the follow-up packet to the

previous HTS Trigger, the node will originate the follow-up packet to transport the former set of the network state data and transmit it before it sends the follow-up packet with the latest collection of network state information.

The following sections describe the operation of HTS nodes in the downstream mode of collecting the telemetry information. In the upstream mode, the behavior of HTS nodes, in general, identical with the exception that the HTS Trigger packet does not precede the HTS Follow-up packet.

4.1. Operation of the HTS Ingress Node

A node that originates the HTS Trigger is referred to as the HTS ingress node. As stated, the ingress node originates the follow-up packet. The follow-up packet has the transport network encapsulation identical with the trigger packet followed by the HTS shim and one or more telemetry information elements encoded as Type-Length-Value {TLV}. Figure 2 displays an example of the follow-up packet format.

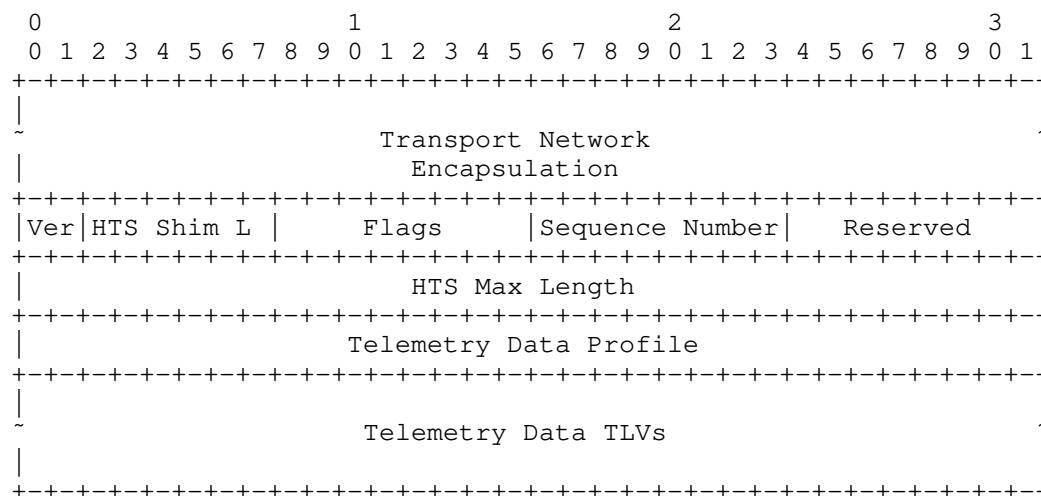


Figure 2: Follow-up Packet Format

Fields of the HTS shim are as follows:

Version (Ver) is the two-bits long field. It specifies the version of the HTS shim format. This document defines the format for the 0b00 value of the field.

HTS Shim Length is the six bits-long field. It defines the length of the HTS shim in octets. The minimal value of the field is eight octets.

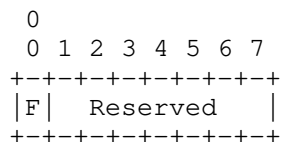


Figure 3: Flags Field Format

Flags is eight-bits long. The format of the Flags field displayed in Figure 3.

- Full (F) flag MUST be set to zero by the node originating the HTS follow-up packet and MUST be set to one by the node that does not add its telemetry data to avoid exceeding MTU size.
- The node originating the follow-up packet MUST zero the Reserved field and ignore it on the receipt.

Sequence Number is one octet-long field. The zero-based value of the field reflects the place of the HTS follow-up packet in the sequence of the HTS follow-up packets that originated in response to the same HTS trigger. The ingress node MUST set the value of the field to zero.

Reserved is one octet-long field. It MUST be zeroed on transmission and ignored on receipt.

HTS Max Length is four octet-long field. The value of the HTS Max Length field indicates the maximum length of the HTS Follow-up packet in octets. An operator MUST be able to configure the HTS Max Length field's value. The value SHOULD be set equal to the path MTU.

Telemetry Data Profile is the optional variable-length field of bit-size flags. Each flag indicates the requested type of telemetry data to be collected at each HTS node. The increment of the field is four bytes with a minimum length of zero. For example, IOAM-Trace-Type information defined in [I-D.ietf-ippm-ioam-data] can be used in the Telemetry Data Profile field.

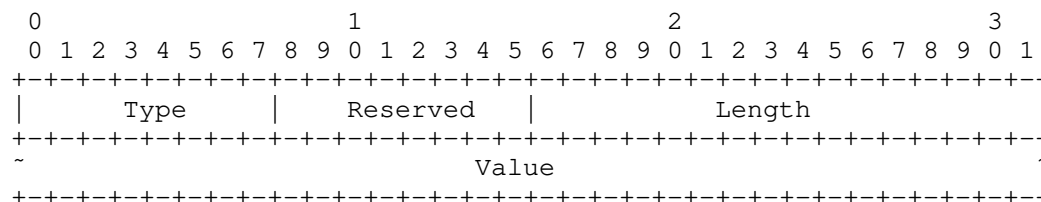


Figure 4: Telemetry Data TLV Format

Telemetry Data TLV is a variable-length field. Multiple TLVs MAY be placed in an HTS packet. Additional TLVs may be enclosed within a given TLV, subject to the semantics of the (outer) TLV in question. Figure 4 presents the format of a Telemetry Data TLV, where fields are defined as the following:

- Type - a one-octet-long field that characterizes the interpretation of the Value field.
- Reserved - one-octet-long field.
- Length - two-octet-long field equal to the length of the Value field in octets.
- Value - a variable-length field. The value of the Type field determines its interpretation and encoding. IOAM data fields, defined in [I-D.ietf-ippm-ioam-data], MAY be carried in the Value field.

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

4.2. Operation of the HTS Intermediate Node

Upon receiving the trigger packet, the HTS intermediate node MUST:

- * copy the transport information;
- * start the HTS Follow-up Timer for the obtained flow;
- * transmit the trigger packet.

Upon receiving the follow-up packet, the HTS intermediate node MUST:

1. verify that the matching transport information exists and the Full flag is cleared, then stop the associated HTS Follow-up Timer;

2. otherwise, transmit the received packet. Proceed to Step 8;
3. collect telemetry data requested in the Telemetry Data Profile field or defined by the local HTS policy;
4. if adding the collected telemetry would not exceed HTS Max Length field's value, then append data as a new Telemetry Data TLV and transmit the follow-up packet. Proceed to Step 8;
5. otherwise, set the value of the Full flag to one, copy the transport information from the received follow-up packet and transmit it accordingly. Proceed to Step 8;
6. originate the new follow-up packet using the transport information copied from the received follow-up packet. The value of the Sequence Number field in the HTS shim MUST be set to the value of the field in the received follow-up packet incremented by one;
7. copy collected telemetry data into the first Telemetry Data TLV's Value field and then transmit the packet;
8. processing completed.

If the HTS Follow-up Timer expires, the intermediate node MUST:

- * originate the follow-up packet using transport information associated with the expired timer;
- * initialize the HTS shim by setting the Version field's value to 0b00 and Sequence Number field to 0. Values of HTS Shim Length and Telemetry Data Profile fields MAY be set according to the local policy.
- * copy telemetry information into Telemetry Data TLV's Value field and transmit the packet.

If the intermediate node receives a "late" follow-up packet, i.e., a packet to which the node has no associated HTS Follow-up timer, the node MUST forward the "late" packet.

4.3. Operation of the HTS Egress Node

Upon receiving the trigger packet, the HTS egress node MUST:

- * copy the transport information;
- * start the HTS Collection timer for the obtained flow.

When the egress node receives the follow-up packet for the known flow, i.e., the flow to which the Collection timer is running, the node for each of Telemetry Data TLVs MUST:

- * if HTS is used in the authenticated mode, verify the authentication of the Telemetry Data TLV using the Authentication sub-TLV (see Section 5);
- * copy telemetry information from the Value field;
- * restart the corresponding Collection timer.

When the Collection timer expires, the egress relays the collected telemetry information for processing and analysis to a local or remote agent.

4.4. Considerations for HTS Timers

This specification defines two timers - HTS Follow-up and HTS Collection. For the particular flow, there MUST be no more than one HTS Trigger, values of HTS timers bounded by the rate of the trigger generation for that flow.

4.5. Deploying HTS in a Multicast Network

Previous sections discussed the operation of HTS in a unicast network. Multicast services are important, and the ability to collect telemetry information is invaluable in delivering a high quality of experience. While the replication of data packets is necessary, replication of HTS follow-up packets is not. Replication of multicast data packets down a multicast tree may be set based on multicast routing information or explicit information included in the special header, as, for example, in Bit-Indexed Explicit Replication [RFC8296]. A replicating node processes the HTS packet as defined below:

- * the first transmitted multicast packet MUST be followed by the received corresponding HTS packet as described in Section 4.2;
- * each consecutively transmitted copy of the original multicast packet MUST be followed by the new HTS packet originated by the replicating node that acts as an intermediate HTS node when the HTS Follow-up timer expired.

As a result, there are no duplicate copies of Telemetry Data TLV for the same pair of ingress and egress interfaces. At the same time, all ingress/egress pairs traversed by the given multicast packet reflected in their respective Telemetry Data TLV. Consequently, a

centralized controller would reconstruct and analyze the state of the particular multicast distribution tree based on HTS packets collected from egress nodes.

5. Authentication in HTS

Telemetry information may be used to drive network operation, closing the control loop for self-driving, self-healing networks. Thus it is critical to provide a mechanism to protect the telemetry information collected using the HTS method. This document defines an optional authentication of a Telemetry Data TLV that protects the collected information's integrity.

The format of the Authentication sub-TLV is displayed in Figure 5.

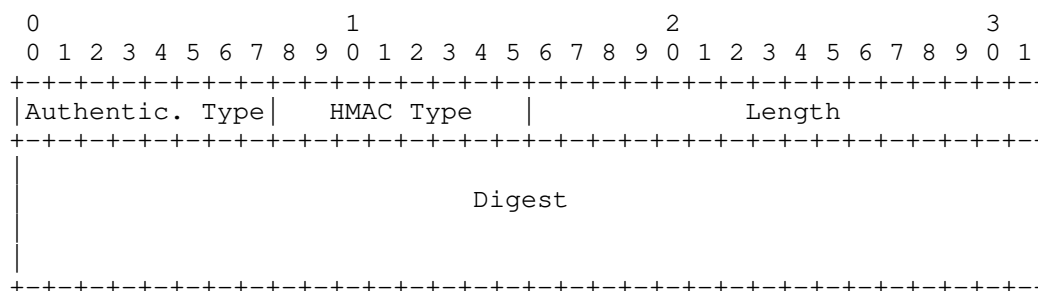


Figure 5: HMAC sub-TLV

where fields are defined as follows:

- * Authentication Type - is a one-octet-long field, value TBA2 allocated by IANA Section 6.2.
- * Length - two-octet-long field, set equal to the length of the Digest field in octets.
- * HMAC Type - is a one-octet-long field that identifies the type of the HMAC and the length of the digest and the length of the digest according to the HTS HMAC Type sub-registry (see Section 6.4).
- * Digest - is a variable-length field that carries HMAC digest of the text that includes the encompassing TLV.

This specification defines the use of HMAC-SHA-256 truncated to 128 bits ([RFC4868]) in HTS. Future specifications may define the use in HTS of more advanced cryptographic algorithms or the use of digest of a different length. HMAC is calculated as defined in [RFC2104] over

text as the concatenation of the Sequence Number field of the follow-up packet (see Figure 2) and the preceding data collected in the Telemetry Data TLV. The digest then MUST be truncated to 128 bits and written into the Digest field. Distribution and management of shared keys are outside the scope of this document. In the HTS authenticated mode, the Authentication sub-TLV MUST be present in each Telemetry Data TLV. HMAC MUST be verified before using any data in the included Telemetry Data TLV. If HMAC verification fails, the system MUST stop processing corresponding Telemetry Data TLV and notify an operator. Specification of the notification mechanism is outside the scope of this document.

6. IANA Considerations

6.1. IOAM Option-Type for HTS

The IOAM Option-Type registry is requested in [I-D.ietf-ippm-ioam-data]. IANA is requested to allocate a new code point as listed in Table 1.

Value	Description	Reference
TBA1	IOAM Hybrid Two-Step Option-Type	This document

Table 1: IOAM Option-Type for HTS

6.2. HTS TLV Registry

IANA is requested to create the HTS TLV Type registry. All code points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure specified in [RFC8126]. The remaining code points are allocated according to Table 2:

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

Table 2: HTS TLV Type Registry

6.3. HTS Sub-TLV Type Sub-registry

IANA is requested to create the HTS sub-TLV Type sub-registry as part of the HTS TLV Type registry. All code points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure specified in [RFC8126]. The remaining code points are allocated according to Table 3:

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

Table 3: HTS Sub-TLV Type Sub-registry

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

Value	Description	TLV Used	Reference
TBA2	HMAC	Any	This document

Table 4: HTS sub-TLV Types

6.4. HMAC Type Sub-registry

IANA is requested to create the HMAC Type sub-registry as part of the HTS TLV Type registry. All code points in the range 1 through 127 in this registry shall be allocated according to the "IETF Review" procedure 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 specified in [RFC8126]. The remaining code points are allocated according to Table 5:

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

Table 5: HMAC Type Sub-registry

This document defines the following new values in the HMAC Type sub-registry:

Value	Description	Reference
1	HMAC-SHA-256 16 octets long	This document

Table 6: HMAC Types

7. Security Considerations

Nodes that practice the HTS method are presumed to share a trust model that depends on the existence of a trusted relationship among nodes. This is necessary as these nodes are expected to correctly modify the specific content of the data in the follow-up packet, and the degree to which HTS measurement is useful for network operation depends on this ability. In practice, this means either confidentiality or integrity protection cannot cover those portions of messages that contain the network state data. Though there are methods that make it possible in theory to provide either or both such protections and still allow for intermediate nodes to make detectable yet authenticated modifications, such methods do not seem practical at present, particularly for protocols that used to measure latency and/or jitter.

This document defines the use of authentication (Section 5) to protect the integrity of the telemetry information collected using the HTS method. Privacy protection can be achieved by, for example, sharing the IPsec tunnel with a data flow that generates information that is collected using HTS.

While it is possible for a supposed compromised node to intercept and modify the network state information in the follow-up packet; this is an issue that exists for nodes in general - for all data that to be carried over the particular networking technology - and is therefore the basis for an additional presumed trust model associated with an existing network.

8. Acknowledgments

Authors express their gratitude and appreciation to Joel Halpern for the most helpful and insightful discussion on the applicability of HTS in a Service Function Chaining domain.

9. References

9.1. Normative References

- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [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://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-data-17>>.
- [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://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-direct-export-07>>.
- [I-D.ietf-raw-use-cases]
Bernardos, C. J., Papadopoulos, G. Z., Thubert, P., and F. Theoleyre, "RAW use-cases", Work in Progress, Internet-Draft, draft-ietf-raw-use-cases-05, 23 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-raw-use-cases-05>>.
- [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://datatracker.ietf.org/doc/html/draft-song-ippm-postcard-based-telemetry-11>>.
- [P4.INT] "In-band Network Telemetry (INT)", P4.org Specification, October 2017.
- [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>>.

- [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>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [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>>.
- [RFC8169] Mirsky, G., Ruffini, S., Gray, E., Drake, J., Bryant, S., and A. Vainshtein, "Residence Time Measurement in MPLS Networks", RFC 8169, DOI 10.17487/RFC8169, May 2017, <<https://www.rfc-editor.org/info/rfc8169>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [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>>.
- [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>>.
- [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>>.
- [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

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com

Wang Lingqiang
ZTE Corporation
No 19 ,East Huayuan Road
Beijing
Phone: +86 10 82963945
Email: wang.lingqiang@zte.com.cn

Guo Zhui
ZTE Corporation
No 19 ,East Huayuan Road
Beijing
Phone: +86 10 82963945
Email: guo.zhui@zte.com.cn

Haoyu Song
Futurewei Technologies
2330 Central Expressway
Santa Clara,
United States of America
Email: hsong@futurewei.com

Pascal Thubert
Cisco Systems, Inc
Building D
45 Allée des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France
Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Network Working Group
Internet-Draft
Updates: ???? (if approved)
Intended status: Standards Track
Expires: May 7, 2020

A. Morton
AT&T Labs
R. Geib
Deutsche Telekom
L. Ciavattone
AT&T Labs
November 4, 2019

Metrics and Methods for IP Capacity
draft-morton-ippm-capacity-metric-method-01

Abstract

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

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 May 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
2. Scope and Goals	4
3. Motivation	4
4. General Parameters and Definitions	5
5. IP-Layer Capacity Singleton Metric Definitions	6
5.1. Formal Name	6
5.2. Parameters	6
5.3. Metric Definitions	6
5.4. Related Round-Trip Delay and Loss Definitions	8
5.5. Discussion	8
5.6. Reporting the Metric	8
6. Maximum IP-Layer Capacity Metric Definitions (Statistic)	8
6.1. Formal Name	8
6.2. Parameters	8
6.3. Metric Definitions	9
6.4. Related Round-Trip Delay and Loss Definitions	10
6.5. Discussion	10
6.6. Reporting the Metric	11
7. IP-Layer Sender Bit Rate Singleton Metric Definitions	11
7.1. Formal Name	11
7.2. Parameters	11
7.3. Metric Definition	12
7.4. Discussion	12
7.5. Reporting the Metric	12
8. Method of Measurement	12
8.1. Load Rate Adjustment Algorithm (from udpst)	13
8.2. Measurement Qualification or Verification	14
8.3. Measurement Considerations	15
9. Reporting	16
10. Security Considerations	17
11. IANA Considerations	17

12. Acknowledgements	17
13. References	17
13.1. Normative References	17
13.2. Informative References	19
Authors' Addresses	20

1. Introduction

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

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

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

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

This memo also recognizes the many updates to the IP Performance Metrics Framework [RFC2330] published over twenty years, and makes use of [RFC7312] for Advanced Stream and Sampling Framework, and RFC 8468 [RFC8468] IPv4, IPv6, and IPv4-IPv6 Coexistence Updates.

NOTE: The text contains a few Author comments, in brackets [RG: ,
acm:]

2. Scope and Goals

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

The main goal is to harmonize the specified metric and method across the industry, and this memo is the vehicle through which working group (and eventually, IETF) consensus will be captured and communicated to achieve broad agreement, and possibly result in changes in the specifications of other Standards Development Organizations (SDO) (through the SDO's normal contribution process, or through liaison exchange).

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

3. Motivation

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

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

1. Internet access is no longer the bottleneck for many users.
2. Both speed and latency are important to user's satisfaction.
3. UDP's growing role in Transport, in areas where TCP once dominated.
4. Content and applications moving physically closer to users.
5. Less emphasis on ISP gateway measurements, possibly due to less traffic crossing ISP gateways in future.

4. General Parameters and Definitions

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

- o Src, the address of a host (such as the globally routable IP address).
- o Dst, the address of a host (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 host at Src to the host at Dst (such as the TTL or Hop Limit).
- o MaxHops, the maximum value of i used, (i=1,2,3,...MaxHops).
- o T0, the time at the start of measurement interval, when packets are first transmitted from the Source.
- o I, the duration of a measurement interval (default 10 sec)
- o dt, the duration of N equal sub-intervals in I (default 1 sec)
- o Tmax, a maximum waiting time for test packets to arrive at the destination, set sufficiently long to disambiguate packets with long delays from packets that are discarded (lost), such that the distribution of one-way delay is not truncated.
- o F, the number of different flows synthesized by the method (default 1 flow)
- 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 when routers have complied with [RFC6438] guidelines at the Tunnel End Points (TEP), and the source of the measurement is a TEP.
- o Type-P, the complete description of the packets for which this assessment applies (including the flow-defining fields). Note that the UDP transport layer is one requirement specified below. Type-P is a parallel concept to "population of interest" defined in ITU-T Rec. Y.1540.
- o PM, a list of fundamental metrics, such as loss, delay, and reordering, and corresponding Target performance threshold. At

least one fundamental metric and Target performance threshold MUST be supplied (such as One-way IP Packet Loss [RFC7680] equal to zero).

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

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

Note that time stamps, sequence numbers, etc. will be established by the test protocol.

5. IP-Layer Capacity Singleton Metric Definitions

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

5.1. Formal Name

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

Note that Type-P depends on the chosen method.

5.2. Parameters

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

No additional Parameters are needed.

5.3. Metric Definitions

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

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

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

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

Anticipating a Sample of Singletons, the interval dt SHOULD be set to a natural number m so that $T+I = T + m*dt$ with $dt_n - dt_{n-1} = dt$ and with $0 < n \leq m$.

Parameter PM represents other performance metrics [see section Related Round-Trip Delay and Loss Definitions below]; their measurement results SHALL be collected during measurement of IP-layer Capacity and associated with the corresponding dt_n for further evaluation and reporting.

Mathematically, this definition can be represented as:

$$C(T, I, PM) = \frac{(n0[dt_n-1, dt_n])}{dt}$$

Equation for IP-Layer Capacity

and:

- o $n0$ is the total number of IP-layer header and payload bits that can be transmitted in Standard Formed packets from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length, during the interval $[T, T+I]$,
- o $C(T, I, PM)$ the IP-Layer Capacity, corresponds to the value of $n0$ measured in any sub-interval ending at dt_n (meaning $T + n*dt$), divided by the length of sub-interval, dt .
- o all sub-intervals SHOULD be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.
- o The bit rate of the physical interface of the measurement device must be higher than that of the link whose $C(T, I, PM)$ is to be measured.

Measurements according to these definitions SHALL use UDP transport layer.

5.4. Related Round-Trip Delay and Loss Definitions

RTD[dtn-1,dtn] is defined as a sample of the [RFC2681] Round-trip Delay between the Src host and the Dst host over the interval [T,T+I]. The statistics used to to summarize RTD[dtn-1,dtn] MAY include the minimum, maximum, and mean.

RTL[dtn-1,dtn] is defined as a sample of the [RFC6673] Round-trip Loss between the Src host and the Dst host over the interval [T,T+I]. The statistics used to to summarize RTL[dtn-1,dtn] MAY include the lost packet count and the lost packet ratio.

5.5. Discussion

See the corresponding section for Maximum IP-Layer Capacity.

5.6. Reporting the Metric

The IP-Layer Capacity SHALL be reported with meaningful resolution, in units of Megabits per second (Mbps).

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

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

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

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

6.1. Formal Name

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

Note that Type-P depends on the chosen method.

6.2. Parameters

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

No additional Parameters or definitions are needed.

6.3. Metric Definitions

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

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

The interval dt SHOULD be set to a natural number m so that $T+I = T + m*dt$ with $dt_n - dt_{n-1} = dt$ and with $0 < n \leq m$.

Parameter PM represents the other performance metrics [see section Related Round-Trip Delay and Loss Definitions below] and their measurement results for the maximum IP-layer capacity. At least one target performance threshold (PM criterion) MUST be defined. If more than one target performance threshold is defined, then the sub-interval with maximum number of bits transmitted MUST meet all the target performance thresholds.

Mathematically, this definition can be represented as:

$$\text{Maximum_C}(T, I, PM) = \frac{\max_{[T, T+I]} (n0[dt_n-1, dt_n])}{dt}$$

where:

Equation for Maximum Capacity

and:

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

- o Maximum $_C(T,I,PM)$ the Maximum IP-Layer Capacity, corresponds to the maximum value of n_0 measured in any sub-interval ending at dtn (meaning $T + n*dt$), divided by the constant length of all sub-intervals, dt .
- o all sub-intervals SHOULD be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.
- o The bit rate of the physical interface of the measurement systems must be higher than that of the link whose Maximum $_C(T,I,PM)$ is to be measured (the bottleneck link).

In this definition, the m sub-intervals can be viewed as trials when the Src host varies the transmitted packet rate, searching for the maximum n_0 that meets the PM criteria measured at the Dst host in a test of duration, I . When the transmitted packet rate is held constant at the Src host, the m sub-intervals may also be viewed as trials to evaluate the stability of n_0 and metric(s) in the PM list over all dt -length intervals in I .

Measurements according to these definitions SHALL use UDP transport layer.

6.4. Related Round-Trip Delay and Loss Definitions

RTD[$dtn-1,dtn$] is defined as a sample of the [RFC2681] Round-trip Delay between the Src host and the Dst host over the interval $[T,T+I]$, and corresponds to the dt interval containing Maximum $_C(T,I,PM)$. The statistics used to to summarize RTD[$dtn-1,dtn$] MAY include the minimum, maximum, and mean.

RTL[$dtn-1,dtn$] is defined as a sample of the [RFC6673] Round-trip Loss between the Src host and the Dst host over the interval $[T,T+I]$ and corresponds to the dt interval containing Maximum $_C(T,I,PM)$. The statistics used to to summarize RTL[$dtn-1,dtn$] MAY include the lost packet count and the lost packet ratio.

6.5. Discussion

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

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

6.6. Reporting the Metric

The Maximum IP-Layer Capacity SHALL be reported with meaningful resolution, in units of Megabits per second.

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

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

7. IP-Layer Sender Bit Rate Singleton Metric Definitions

This section sets requirements for the following components to support the IP-layer Sender Bitrate Metric.

7.1. Formal Name

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

Note that Type-P depends on the chosen method.

7.2. Parameters

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

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

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

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

7.3. Metric Definition

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

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

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

7.4. Discussion

Both the Sender and Receiver or (source and destination) bit rates SHOULD be assessed as part of a measurement.

7.5. Reporting the Metric

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

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

8. Method of Measurement

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

Additional Test methods and configurations may be provided in this section, after review and further testing.

8.1. Load Rate Adjustment Algorithm (from udpst)

A table is pre-built defining all the offered load rates that will be supported ($R_1 - R_n$, in ascending order). Each rate is defined as datagrams of size S , sent as a burst of count C , every time interval T . While it is advantageous to use datagrams of as large a size as possible, it may be prudent to use a slightly smaller maximum that allows for secondary protocol headers and/or tunneling without resulting in IP-layer fragmentation.

At the beginning of a test, the sender begins sending at rate R_1 and the receiver starts a feedback timer at interval F (while awaiting inbound datagrams). As datagrams are received they are checked for sequence number anomalies (loss, out-of-order, duplication, etc.) and the delay variation is measured (one-way or round-trip). This information is accumulated until the feedback timer F expires and a status feedback message is sent from the receiver back to the sender, to communicate this information. The accumulated statistics are then reset by the receiver for the next feedback interval. As feedback messages are received back at the sender, they are evaluated to determine how to adjust the current offered load rate (R_x).

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

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

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

Lastly, the method for confirming congestion is that there were sequence number anomalies OR the delay variation was above the upper threshold for two consecutive feedback intervals.

8.2. Measurement Qualification or Verification

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

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

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

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

Here, as with any Active Capacity test, the test duration must be kept short. 10 second tests for each direction of transmission are common today. The default measurement interval specified here is I = 10 seconds). In combination with a fast search method and user-

network coordination, the concerns raised in RFC 6815[RFC6815] are alleviated. The method for assessing Max IP Capacity is different from classic [RFC2544] methods: they use short term load adjustment and are sensitive to loss and delay, like other congestion control algorithms used on the Internet every day.

8.3. Measurement Considerations

In general, the wide-spread measurements that this memo encourages will encounter wide-spread behaviors. The bimodal IP Capacity behavior is a good example.

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

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

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

1. the test hosts may be able to create higher load than with a single flow, or parallel test hosts may be used to generate 1 flow each.
2. there may be link aggregation present (flow-based load balancing) and multiple flows are need to occupy each member of the aggregate.

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

As testing continues, implementers should expect some evolution in the methods.

9. Reporting

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

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

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

Maximum IP-layer Capacity Results

Static and configuration parameters:

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

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

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

IP-layer Sender Bit Rate Results

Static and configuration parameters:

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

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

10. Security Considerations

Active metrics and measurements have a long history of security considerations [add references to LMAP Framework, etc.].

<There are certainly some new ones for Capacity testing>

11. IANA Considerations

This memo makes no requests of IANA.

12. Acknowledgements

Thanks to Joachim Fabini, Matt Mathis, and Ignacio Alvarez-Hamelin for their extensive comments on the memo and related topics.

13. References

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

- [RFC2889] Mandeville, R. and J. Perser, "Benchmarking Methodology for LAN Switching Devices", RFC 2889, DOI 10.17487/RFC2889, August 2000, <<https://www.rfc-editor.org/info/rfc2889>>.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, DOI 10.17487/RFC3148, July 2001, <<https://www.rfc-editor.org/info/rfc3148>>.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <<https://www.rfc-editor.org/info/rfc5136>>.
- [RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <<https://www.rfc-editor.org/info/rfc5180>>.
- [RFC6201] Asati, R., Pignataro, C., Calabria, F., and C. Olvera, "Device Reset Characterization", RFC 6201, DOI 10.17487/RFC6201, March 2011, <<https://www.rfc-editor.org/info/rfc6201>>.
- [RFC6412] Poretsky, S., Imhoff, B., and K. Michielsen, "Terminology for Benchmarking Link-State IGP Data-Plane Route Convergence", RFC 6412, DOI 10.17487/RFC6412, November 2011, <<https://www.rfc-editor.org/info/rfc6412>>.
- [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>>.
- [RFC6815] Bradner, S., Dubray, K., McQuaid, J., and A. Morton, "Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful", RFC 6815, DOI 10.17487/RFC6815, November 2012, <<https://www.rfc-editor.org/info/rfc6815>>.
- [RFC6985] Morton, A., "IMIX Genome: Specification of Variable Packet Sizes for Additional Testing", RFC 6985, DOI 10.17487/RFC6985, July 2013, <<https://www.rfc-editor.org/info/rfc6985>>.

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

13.2. Informative References

- [copycat] Edleine, K., Kuhlewind, K., Trammell, B., and B. Donnet, "copycat: Testing Differential Treatment of New Transport Protocols in the Wild (ANRW '17)", July 2017, <<https://irtf.org/anrw/2017/anrw17-final5.pdf>>.
- [RFC8239] Avramov, L. and J. Rapp, "Data Center Benchmarking Methodology", RFC 8239, DOI 10.17487/RFC8239, August 2017, <<https://www.rfc-editor.org/info/rfc8239>>.
- [TST009] Morton, R. A., "ETSI GS NFV-TST 009 V3.1.1 (2018-10), "Network Functions Virtualisation (NFV) Release 3; Testing; Specification of Networking Benchmarks and Measurement Methods for NFVI"", October 2018, <https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/009/03.01.01_60/gs_NFV-TST009v030101p.pdf>.
- [VSPERF-b2b] Morton, A., "Back2Back Testing Time Series (from CI)", June 2017, <[https://wiki.opnfv.org/display/vsperf/Traffic+Generator+Testing#TrafficGeneratorTesting-AppendixB:Back2BackTestingTimeSeries\(fromCI\)](https://wiki.opnfv.org/display/vsperf/Traffic+Generator+Testing#TrafficGeneratorTesting-AppendixB:Back2BackTestingTimeSeries(fromCI))>.

[VSPERF-BSLV]

Morton, A. and S. Rao, "Evolution of Repeatability in Benchmarking: Fraser Plugfest (Summary for IETF BMWG)", July 2018, <<https://datatracker.ietf.org/meeting/102/materials/slides-102-bmwg-evolution-of-repeatability-in-benchmarking-fraser-plugfest-summary-for-ietf-bmwg-00>>.

Authors' Addresses

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

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

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

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

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

Email: lencia@att.com

IPPM
Internet-Draft
Intended status: Informational
Expires: 13 November 2022

H. Song
Futurewei Technologies
G. Mirsky
Ericsson
C. Filsfils
A. Abdelsalam
Cisco Systems, Inc.
T. Zhou
Z. Li
Huawei
G. Mishra
Verizon Inc.
J. Shin
SK Telecom
K. Lee
LG U+
12 May 2022

In-Situ OAM Marking-based Direct Export
draft-song-ippm-postcard-based-telemetry-12

Abstract

The document describes a packet-marking variation of the IOAM DEX option, referred to as IOAM Marking. Similar to IOAM DEX, IOAM Marking does not carry the telemetry data in user packets but send the telemetry data through a dedicated packet. Unlike IOAM DEX, IOAM Marking does not require an extra instruction header. IOAM Marking raises some unique issues that need to be considered. This document formally describes the high level scheme and cover the common requirements and issues when applying IOAM Marking in different networks. IOAM Marking is complementary to the other on-path telemetry schemes such as IOAM trace and E2E options.

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 November 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. Motivation	3
2. IOAM Marking: Marking-based IOAM Direct Export	3
3. New Challenges	5
4. IOAM Marking Design Considerations	6
4.1. Packet Marking	6
4.2. Flow Path Discovery	7
4.3. Packet Identity for Export Data Correlation	7
4.4. Control the Load	8
5. Implementation Recommendation	8
5.1. Configuration	8
5.2. Postcard Format	8
5.3. Data Correlation	9
6. Use Cases	9
7. Security Considerations	10
8. IANA Considerations	10
9. Contributors	10
10. Acknowledgments	10
11. Informative References	10
Authors' Addresses	12

1. Motivation

To gain detailed data plane visibility to support effective network OAM, it is essential to be able to examine the trace of user packets along their forwarding paths. Such on-path flow data reflect the state and status of each user packet's real-time experience and provide valuable information for network monitoring, measurement, and diagnosis.

The telemetry data include but not limited to the detailed forwarding path, the timestamp/latency at each network node, and, in case of packet drop, the drop location, and the reason. The emerging programmable data plane devices allow user-defined data collection or conditional data collection based on trigger events. Such on-path flow data are from and about the live user traffic, which complements the data acquired through other passive and active OAM mechanisms such as IPFIX [RFC7011] and ICMP [RFC2925].

On-path telemetry was developed to cater to the need of collecting on-path flow data. There are two basic modes for on-path telemetry: the passport mode and the postcard mode. In the passport mode which is represented by IOAM trace option [I-D.ietf-ippm-ioam-data], each node on the path adds the telemetry data to the user packets (i.e., stamp the passport). The accumulated data-trace carried by user packets are exported at a configured end node. In the postcard mode which is represented by IOAM direct export option (DEX) [I-D.ietf-ippm-ioam-direct-export], each node directly exports the telemetry data using an independent packet (i.e., send a postcard) to avoid carrying the data with user packets. The postcard mode is complementary to the passport mode.

IOAM DEX uses an instruction header to explicitly instruct the telemetry data to be collected. This document describes another variation of the postcard mode on-path telemetry, IOAM Marking. Unlike IOAM DEX, IOAM Marking does not require a telemetry instruction header. However, IOAM Marking has unique issues that need to be considered. This document discusses the challenges and their solutions which are common to the high-level scheme of IOAM Marking.

2. IOAM Marking: Marking-based IOAM Direct Export

As the name suggests, IOAM Marking only needs a marking-bit in the existing headers of user packets to trigger the telemetry data collection and export. The sketch of IOAM Marking is as follows. If on-path data need to be collected, the user packet is marked at the path head node. At each IOAM Marking-aware node, if the mark is detected, a postcard (i.e., the dedicated OAM packet triggered by a

marked user packet) is generated and sent to a collector. The postcard contains the data requested by the management plane. The requested data are configured by the management plane. Once the collector receives all the postcards for a single user packet, it can infer the packet's forwarding path and analyze the data set. The path end node is configured to unmark the packets to its original format if necessary.

The overall architecture of IOAM Marking is depicted in Figure 1.

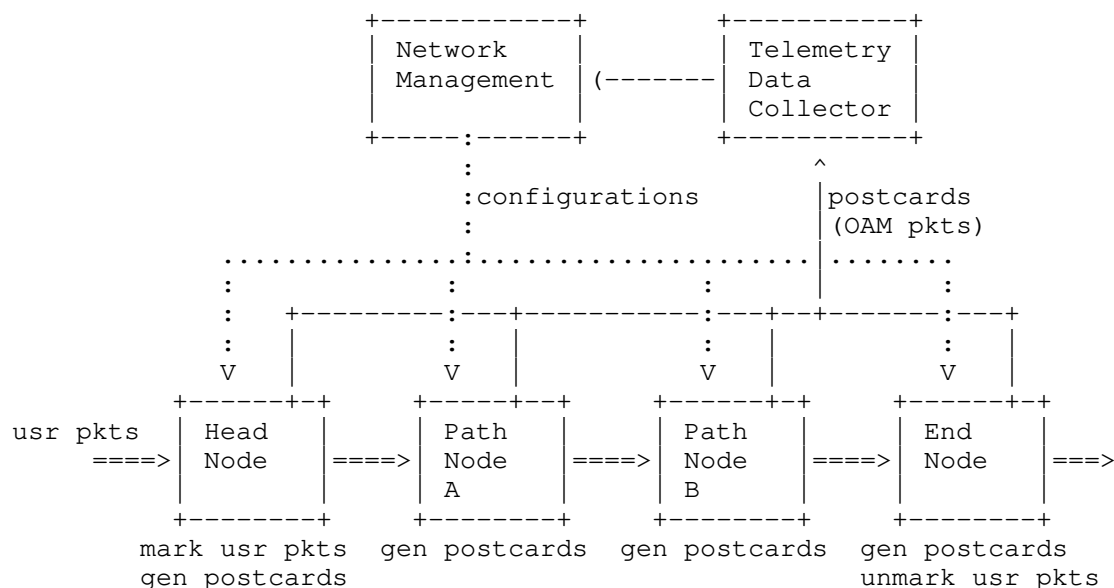


Figure 1: Architecture of IOAM Marking

The advantages of IOAM Marking are summarized as follows.

- * 1: IOAM Marking avoids augmenting user packets with new headers and the signaling for telemetry data collection remains in the data plane.
- * 2: IOAM Marking is extensible for collecting arbitrary new data to support possible future use cases. The data set to be collected can be configured through the management plane or control plane.

- * 3: IOAM Marking can avoid interfering with the normal forwarding. The collected data are free to be transported independently through in-band or out-of-band channels. The data collecting, processing, assembly, encapsulation, and transport are, therefore, decoupled from the forwarding of the corresponding user packets and can be performed in data-plane slow-path if necessary.
- * 4: For IOAM Marking, the types of data collected from each node can vary depending on application requirements and node capability.
- * 5: IOAM Marking makes it easy to secure the collected data without exposing it to unnecessary entities. For example, both the configuration and the telemetry data can be encrypted and/or authenticated before being transported, so passive eavesdropping and a man-in-the-middle attack can both be deterred.
- * 6: Even if a user packet under inspection is dropped at some node in the network, the postcards collected from the preceding nodes are still valid and can be used to diagnose the packet drop location and reason.

3. New Challenges

Although IOAM Marking has some unique features compared to the passport mode telemetry and the instruction-based IOAM DEX, it introduces a few new challenges.

- * Challenge 1 (Packet Marking): A user packet needs to be marked to trigger the path-associated data collection. Since IOAM Marking does not augment user packets with any new header fields, it needs to reserve or reuse bits from the existing header fields. This raises a similar issue as in the Alternate Marking Scheme [RFC8321]
- * Challenge 2 (Configuration): Since the packet header will not carry IOAM instructions anymore, the data plane devices need to be configured to know what data to collect. However, in general, the forwarding path of a flow packet (due to ECMP or dynamic routing) is unknown beforehand (note that there are some notable exceptions, such as segment routing). If the per-flow customized data collection is required, configuring the data set for each flow at all data plane devices might be expensive in terms of configuration load and data plane resources.
- * Challenge 3 (Data Correlation): Due to the variable transport latency, the dedicated postcard packets for a single packet may arrive at the collector out of order or be dropped in networks for

some reason. In order to infer the packet forwarding path, the collector needs some information from the postcard packets to identify the user packet affiliation and the order of path node traversal.

- * Challenge 4 (Load Overhead): Since each postcard packet has its header, the overall network bandwidth overhead of IOAM Marking can be high. A large number of postcards could add processing pressure on data collecting servers. That can be used as an attack vector for DoS.

4. IOAM Marking Design Considerations

To address the above challenges, we propose several design details of IOAM Marking.

4.1. Packet Marking

To trigger the path-associated data collection, usually, a single bit from some header field is sufficient. While no such bit is available, other packet-marking techniques are needed. We discuss several possible application scenarios.

- * IPv4. Alternate Marking (AM) [RFC8321] is an IP flow performance measurement framework that also requires a single bit for packet coloring. The difference is that AM does in-network measurement while IOAM Marking only collects and exports data at network nodes (i.e., the data analysis is done at the collector rather than in the network nodes). AM suggests to use some reserved bit of the Flag field or some unused bit of the TOS field. Actually, AM can be considered a sub-case of IOAM Marking, so that the same bit can be used for IOAM Marking. The management plane is responsible for configuring the actual operation mode.
- * SFC NSH. The OAM bit in the NSH header can be used to trigger the on-path data collection [RFC8300]. IOAM Marking does not add any other metadata to NSH.
- * MPLS. Instead of choosing a header bit, we take advantage of the synonymous flow label [I-D.bryant-mpls-synonymous-flow-labels] approach to mark the packets. A synonymous flow label indicates the on-path data should be collected and forwarded through a postcard.
- * SRv6: A flag bit in SRH can be reserved to trigger the on-path data collection [I-D.song-6man-srv6-pbt]. SRv6 OAM [I-D.ietf-6man-spring-srv6-oam] has adopted the O-bit in SRH flags as the marking bit to trigger the telemetry.

4.2. Flow Path Discovery

In case the path that a flow traverses is unknown in advance, all IOAM Marking-aware nodes should be configured to react to the marked packets by exporting some basic data, such as node ID and TTL before a data set template for that flow is configured. This way, the management plane can learn the flow path dynamically.

If the management plane wants to collect the on-path data for some flow, it configures the head node(s) with a probability or time interval for the flow packet marking. When the first marked packet is forwarded in the network, the IOAM Marking-aware nodes will export the basic data set to the collector. Hence, the flow path is identified. If other data types need to be collected, the management plane can further configure the data set's template to the target nodes on the flow's path. The IOAM Marking-aware nodes collect and export data accordingly if the packet is marked and a data set template is present.

If the flow path is changed for any reason, the new path can be quickly learned by the collector. Consequently, the management plane controller can be directed to configure the nodes on the new path. The outdated configuration can be automatically timed out or explicitly revoked by the management plane controller.

4.3. Packet Identity for Export Data Correlation

The collector needs to correlate all the postcard packets for a single user packet. Once this is done, the TTL (or the timestamp, if the network time is synchronized) can be used to infer the flow forwarding path. The key issue here is to correlate all the postcards for the same user packet.

The first possible approach includes the flow ID plus the user packet ID in the OAM packets. For example, the flow ID can be the 5-tuple IP header of the user traffic, and the user packet ID can be some unique information pertaining to a user packet (e.g., the sequence number of a TCP packet).

If the packet marking interval is large enough, the flow ID is enough to identify a user packet. As a result, it can be assumed that all the exported postcard packets for the same flow during a short time interval belong to the same user packet.

Alternatively, if the network is synchronized, then the flow ID plus the timestamp at each node can also infer the postcard affiliation. However, some errors may occur under some circumstances. For example, two consecutive user packets from the same flows are marked,

but one exported postcard from a node is lost. It is difficult for the collector to decide to which user packet the remaining postcard is related. In many cases, such a rare error has no catastrophic consequence. Therefore it is tolerable.

4.4. Control the Load

IOAM Marking should not be applied to all the packets all the time. It is better to be used in an interactive environment where the network telemetry applications dynamically decide which subset of traffic is under scrutiny. The network devices can limit the packet marking rate through sampling and metering. The postcard packets can be distributed to different servers to balance the processing load.

It is important to understand that the total amount of data exported by IOAM Marking is identical to that of IOAM trace option. The only extra overhead is the packet header of the postcards. In the case of IOAM trace option, it carries the data from each node throughout the path to the end node before exporting the aggregated data. On the other hand, IOAM Marking directly exports local data. The overall network bandwidth impact depends on the network topology and scale, and in some cases IOAM Marking could be more bandwidth efficient.

5. Implementation Recommendation

5.1. Configuration

The head node's ACL should be configured to filter out the target flows for telemetry data collection. Optionally, a flow packet sampling rate or probability could be configured to monitor a subset of the flow packets.

The telemetry data set that should be exported by postcards at each path node could be configured using the data set templates specified, for example, in IPFIX [RFC7011]. In future revisions, we will provide more details.

The IOAM Marking-aware path nodes could be configured to respond or ignore the marked packets.

5.2. Postcard Format

The postcard should use the same data export format as that used by IOAM. [I-D.spiegel-ippm-ioam-rawexport] proposes a raw format that can be interpreted by IPFIX. In future revisions, we will provide more details.

5.3. Data Correlation

Enough information should be included to help the collector to correlate and order the postcards for a single user packet. Section 4.3 provides several possible means. The application scenario and network protocol are important factors to determine the means to use. In future revisions, we will provide details for representative applications.

6. Use Cases

The MPLS Design Team has been investigating extensibility options for the MPLS data plane.

The challenge has been to continue to support existing MPLS architecture, backwards compatibility as well as not excessively increase the depth of the MPLS label stack with a variety of functional SPL labels and NAI indicators similar in concept to the MPLS Entropy label ELI, EL added to the label stack, as well as the MPLS extension headers being in Stack or post stack.

Reference Augmented Forwarding (RAF) [I-D.raszuk-mpls-raf-fwk] utilizes In Stack Data (ISD) with parity to Entropy Label stack {TL,RFI,RFV,AL} and control plane extension to distribute special network actions and forwarding behaviors.

Reference Augmented Forwarding (RAF) keeps the ISD and PSD stack depth in check by using an alternative means of carrying the IOAM data using IGP control plane extension TLV to carry the data to provide In-Situ IOAM on path telemetry using the postcard based telemetry.

The MPLS Design Team may come up with other alternatives to carry IOAM data such as the IGP extension mentioned and maybe other solutions, which will heavily rely on the the postcard based solution.

With Segment Routing SR-MPLS and SRv6 as Maximum SID Depth(MSD) as well as PMTU in SR Policy are critical issues for SR path instantiation by a controller, postcard based telemetry will become a critical solution to ensure that IOAM telemetry can be viable for operators by eliminating IOAM data from being carried in-situ in the SR-TE policy path.

This draft provides a critical optimization that fills the gaps with IOAM DEX related to packet marking triggers using existing mechanisms as well as flow path discovery mechanisms to avoid configuration of on path data plane node complexity and helps mitigate SR MSD and PMTU issues.

7. Security Considerations

Several security issues need to be considered.

- * Eavesdrop and tamper: the postcards can be encrypted and authenticated to avoid such security threats.
- * DoS attack: IOAM Marking can be limited to a single administrative domain. The mark must be removed at the egress domain edge. The node can rate-limit the extra traffic incurred by postcards.

8. IANA Considerations

No requirement for IANA is identified.

9. Contributors

We thank Alfred Morton who provided valuable suggestions and comments helping improve this draft.

10. Acknowledgments

TBD.

11. Informative References

[I-D.bryant-mpls-synonymous-flow-labels]

Bryant, S., Swallow, G., Sivabalan, S., Mirsky, G., Chen, M., and Z. Li, "RFC6374 Synonymous Flow Labels", Work in Progress, Internet-Draft, draft-bryant-mpls-synonymous-flow-labels-01, 4 July 2015, <<https://www.ietf.org/archive/id/draft-bryant-mpls-synonymous-flow-labels-01.txt>>.

[I-D.ietf-6man-spring-srv6-oam]

Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", Work in Progress, Internet-Draft, draft-ietf-6man-spring-srv6-oam-13, 23 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-6man-spring-srv6-oam-13.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-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.raszuk-mpls-raf-fwk]
Raszuk, R., "Framework of MPLS Reference Augmented Forwarding", Work in Progress, Internet-Draft, draft-raszuk-mpls-raf-fwk-00, 25 April 2022, <<https://www.ietf.org/archive/id/draft-raszuk-mpls-raf-fwk-00.txt>>.
- [I-D.song-6man-srv6-pbt]
Song, H., "Support Postcard-Based Telemetry for SRv6 OAM", Work in Progress, Internet-Draft, draft-song-6man-srv6-pbt-01, 14 October 2019, <<https://www.ietf.org/archive/id/draft-song-6man-srv6-pbt-01.txt>>.
- [I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", Work in Progress, Internet-Draft, draft-spiegel-ippm-ioam-rawexport-06, 21 February 2022, <<https://www.ietf.org/archive/id/draft-spiegel-ippm-ioam-rawexport-06.txt>>.
- [RFC2925] White, K., "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations", RFC 2925, DOI 10.17487/RFC2925, September 2000, <<https://www.rfc-editor.org/info/rfc2925>>.
- [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>>.

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

Authors' Addresses

Haoyu Song
Futurewei Technologies
2330 Central Expressway
Santa Clara, 95050,
United States of America
Email: hsong@futurewei.com

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium
Email: cfilsfil@cisco.com

Ahmed Abdelsalam
Cisco Systems, Inc.
Italy
Email: ahabdels@cisco.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China
Email: zhoutianran@huawei.com

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China
Email: lizhenbin@huawei.com

Gyan Mishra
Verizon Inc.
Email: hayabusagsm@gmail.com

Jongyoon Shin
SK Telecom
South Korea
Email: jongyoon.shin@sk.com

Kyungtae Lee
LG U+
South Korea
Email: coolee@lguplus.co.kr

MBONED
Internet-Draft
Intended status: Standards Track
Expires: July 23, 2021

H. Song
M. McBride
Futurewei Technologies
G. Mirsky
ZTE Corp.
G. Mishra
Verizon Inc.
January 19, 2021

Multicast On-path Telemetry Solutions
draft-song-multicast-telemetry-07

Abstract

This document discusses the requirement of on-path telemetry for multicast traffic. The existing solutions are examined and their issues are identified. Solution modifications are proposed to allow the original multicast tree to be correctly reconstructed without unnecessary replication of telemetry information.

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 July 23, 2021.

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
2. Requirements for Multicast Traffic Telemetry	3
3. Issues of Existing Techniques	4
4. Proposed Modifications to Existing Techniques	4
4.1. Per-hop postcard using IOAM DEX	5
4.2. Per-section postcard	7
5. Considerations for Different Multicast Protocols	8
5.1. Application in PIM	8
5.2. Application in P2MP	9
5.3. Application in BIER	9
6. Security Considerations	10
7. IANA Considerations	10
8. Contributors	10
9. Acknowledgments	10
10. References	10
10.1. Normative References	10
10.2. Informative References	11
Authors' Addresses	12

1. Introduction

Multicast traffic is an important traffic type in today's Internet. Multicast provides services that are often real time (e.g., online meeting) or have strict QoS requirements (e.g., IPTV, Market Data). Multicast packet drop and delay can severely affect the application performance and user experience.

It is important to monitor the performance of the multicast traffic. Existing OAM techniques cannot gain direct and accurate information about the multicast traffic. New on-path telemetry techniques such as In-situ OAM [I-D.ietf-ippm-ioam-data], Postcard-based Telemetry

[I-D.song-ippm-postcard-based-telemetry], and Hybrid Two-Step (HTS) [I-D.mirsky-ippm-hybrid-two-step] provide promising means to directly monitor the network experience of multicast traffic. However, multicast traffic has some unique characteristics which pose some challenges on efficiently applying such techniques.

When a network contains multicast (p2mp) trees there will be redundant data as data is replicated at branch points. The IP Multicast S,G data is identical from one branch to another on it's way to multiple receivers. When adding iOAM trace data, to multicast packets, we enlarge data packets thus consuming more network bandwidth. Instead of adding iOAM trace data, it could be more efficient to collect the telemetry information using solutions, such as iOAM postcard or HTS, to cut down on the redundant iOAM data. The problem is that a postcard type solution doesn't have a branch identifier.

This draft proposes a set of solutions to this iOAM data redundancy problem. The requirements for multicast traffic telemetry are discussed along with the issues of the existing on-path telemetry techniques. We propose modifications to make these techniques adapt to multicast in order for the original multicast tree to be correctly reconstructed while eliminating redundant data.

2. Requirements for Multicast Traffic Telemetry

Multicast traffic is forwarded through a multicast tree. With PIM and P2MP (MLDP, RSVP-TE) the forwarding tree is established and maintained by the multicast routing protocol. With BIER, no state is created in the network to establish a forwarding tree, instead, a bier header provides the necessary information for each packet to know the egress points. Multicast packets are only replicated at each tree branch node for efficiency.

There are several requirements for multicast traffic telemetry, a few of which are:

- o Reconstruct and visualize the multicast tree through data plane monitoring.
- o Gather the multicast packet delay and jitter performance.
- o Find the multicast packet drop location and reason.
- o Gather the VPN state and tunnel information in case of P2MP multicast.

In order to meet these requirements, we need the ability to directly monitor the multicast traffic and derive data from the multicast packets. The conventional OAM mechanisms, such as multicast ping and trace, may not be sufficient to meet these requirements.

3. Issues of Existing Techniques

On-path Telemetry techniques that directly retrieve data from multicast traffic's live network experience are ideal to address the above mentioned requirements. The representative techniques include In-situ OAM (IOAM) Trace option [I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX) option [I-D.ioamteam-ippm-ioam-direct-export], and Postcard-based Telemetry with Packet Marking (PBT-M) [I-D.song-ippm-postcard-based-telemetry]. However, unlike unicast, multicast poses some unique challenges to applying these techniques.

Multicast packets are replicated at each branch node in the corresponding multicast tree. Therefore, there are multiple copies of packets in the network.

If the IOAM trace option is used for on-path data collection, the partial trace data will also be replicated into multiple copies. The end result is that each copy of the multicast packet has a complete trace. Most of the data, however, is redundant. Data redundancy introduces unnecessary header overhead, wastes network bandwidth, and complicates the data processing. In case the multicast tree is large, and the path is long, the redundancy problem becomes severe.

The PBT solutions, including the IOAM DEX option and PBT-M, can be used to eliminate such data redundancy, because each node on the tree only sends a postcard covering local data. However, they cannot track the tree branches properly so it can bring confusion about the multicast tree topology. For example, Node A has two branches, one to Node B and the other to node D, and Node B leads to Node C and Node D leads to Node E. From the received postcards, one cannot tell whether or not Node C(E) is the next hop of Node B(D).

The fundamental reason for this problem is that there is not an identifier (either implicit or explicit) to correlate the data on each branch.

4. Proposed Modifications to Existing Techniques

Two solutions are proposed to address the above issues. One is built on PBT and requires augmentation or modification to the instruction header of the IOAM Direct Export Option; the other combines the IOAM trace option and PBT for an optimized solution.

4.1. Per-hop postcard using IOAM DEX

One way to mitigate PBT's multiple tree tracking weakness is to augment it with a branch identifier field. Note that this works for the IOAM DEX option but not for PBT-M because the IOAM DEX option uses an instruction header. To make the branch identifier globally unique, the branch node ID plus an index is used. For example, if Node A has two branches, one to Node B and one to Node C, Node A will use [A, 0] as the branch identifier for the branch to B, and [A, 1] for the branch to C. The identifier is unchanged for each multicast tree instance and carried with the multicast packet until the next branch node. Each postcard needs to include the branch identifier in the export data. The branch identifier, along with the other fields such as flow ID and sequence number, is sufficient for the data analyzer to reconstruct the topology of the multicast tree.

Figure 1 shows an example of this solution. "P" stands for the postcard packet. The square brackets contains the branch identifier. The curly brace contains the telemetry data about a specific node.

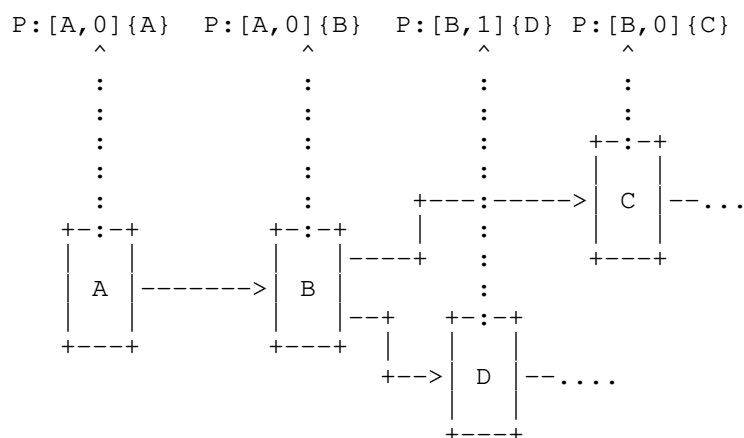


Figure 1: Per-hop Postcard

Each branch fork node need to generate the branch ID for each branch in its multicast tree instance and include it in the IOAM DEX option header so the downstream node can learn it. The branch ID contains two parts: the branch fork node ID and a unique branch index.

Figure 2 shows that the branch ID is carried as an optional field after the flow ID and sequence number optional fields in the IOAM DEX option header. A bit "M" in the Flags field is reserved to indicate

the presence of the branch index field. The "M" flag position will be determined later after the other flags are specified in [I-D.ioamteam-ippm-ioam-direct-export].

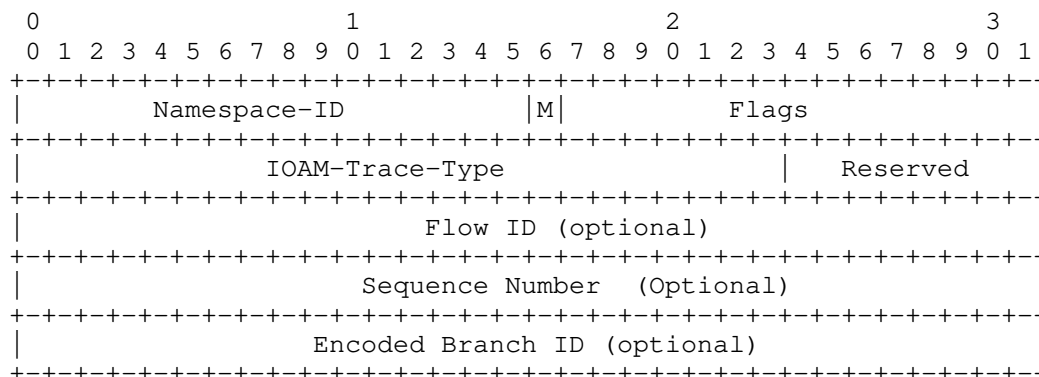


Figure 2: Carry Branch Index in IOAM DEX option header

To avoid introducing a new type of data field to the IOAM DEX option header, we can encode the branch identifier using the existing node ID data field as defined in [I-D.ietf-ippm-ioam-data]. Currently, the node ID field occupies three octets. A simple solution is to shorten the node ID field so a number of bits can be saved to encode the branch index, as shown in Figure 3.

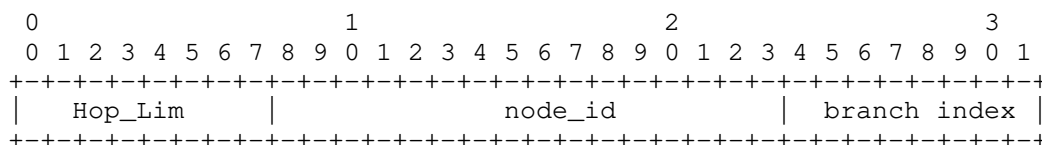


Figure 3: Encode Branch Index with Node ID Method 1

Another encoding method is to use the sum of the node ID and the branch index as the new node ID, as shown in Figure 4. As long as the node IDs are assigned with large enough gap, the telemetry data analyzer can still successfully recover the original node ID and branch index.

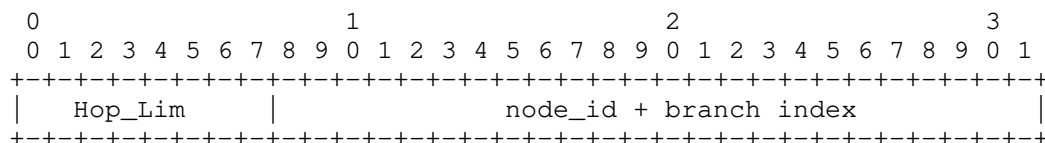


Figure 4: Encode Branch Index with Node ID Method 2

Once a node gets the branch ID information from the upstream, it **MUST** carry this information in its telemetry data export postcards, so the original multicast tree can be correctly reconstructed based on the postcards.

4.2. Per-section postcard

The second solution is a combination of the IOAM trace mode and PBT. To avoid data redundancy at each branch node, the trace data accumulated, to that point, is exported by a postcard before the packet is replicated. In this case, each branch still needs to maintain some identifier to help correlate the postcards for each tree section. The natural way to accomplish this is to simply carry the branch node's data (including its ID) in the trace of each branch. This is also necessary because each replicated multicast packet can have different telemetry data pertaining to this particular copy (e.g., node delay, egress timestamp, and egress interface). As a consequence, the local data exported by each branch node can only contain partial data (e.g., ingress interface and ingress timestamp).

Figure 5 shows an example in a segment of a multicast tree. Node B and D are two branch nodes and they will export a postcard covering the trace data for the previous section. The end node of each path will also need to export the data of the last section as a postcard.

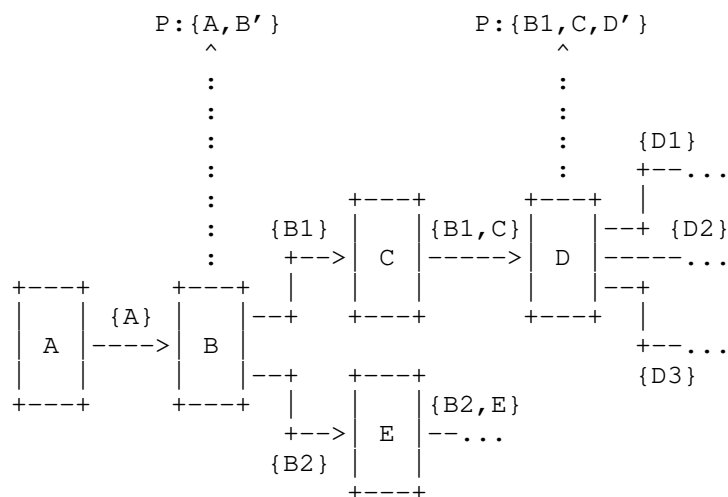


Figure 5: Per-section Postcard

There is no need to modify the IOAM trace mode header format. We just need to configure the branch node to export the postcard and refresh the IOAM header and data.

5. Considerations for Different Multicast Protocols

MTRACEv2 [RFC8487] provides an active probing approach for the tracing of an IP multicast routing path. Mtrace can also provide information such as the packet rates and losses, as well as other diagnostic information. New on-path telemetry techniques will enhance Mtrace, and other existing OAM solutions, with more granular and realtime network status data through direct measurements. There are various multicast protocols that are used to forward the multicast data. Each will require their own unique on-path telemetry solution.

5.1. Application in PIM

PIM-SM [RFC7761] is the most widely used multicast routing protocol deployed today. Of the various PIM modes (PIM-SM, PIM-DM, BIDIR-PIM, PIM-SSM), PIM-SSM is the preferred method due to its simplicity and removal of network source discovery complexity. With all PIM modes, control plane state is established in the network in order to forward multicast UDP data packets. But with PIM-SSM, the discovery of multicast sources is performed outside of the network via HTTP, SDN, etc. IP Multicast packets fall within the range of 224.0.0.0 through

239.255.255.255. The telemetry solution will need to work within this address range and provide telemetry data for this UDP traffic.

The proposed solutions for encapsulating the telemetry instruction header and metadata in IPv4/IPv6 UDP packets are described in [I-D.herbert-ipv4-udpencap-eh] and [I-D.ioametal-ippm-6man-ioam-ipv6-deployment].

5.2. Application in P2MP

Multicast Label Distribution Protocol (MLDP) and P2MP RSVP-TE are commonly used within a Multicast VPN (MVPN) environment. MLDP provides extensions to LDP to establish point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) in MPLS networks. P2MP RSVP-TE provides extensions to RSVP-TE for establish traffic-engineered P2MP LSPs in MPLS networks. The telemetry solution will need to be able to follow these P2MP paths. The telemetry instruction header and data should be encapsulated into MPLS packets on P2MP paths. A corresponding proposal is described in [I-D.song-mpls-extension-header].

5.3. Application in BIER

BIER [RFC8279] adds a new header to multicast packets and allows the multicast packets to be forwarded according to the header only. By eliminating the requirement of maintaining per multicast group state, BIER is more scalable than the traditional multicast solutions.

OAM Requirements for BIER [I-D.ietf-bier-oam-requirements] lists many of the requirements for OAM at the BIER layer which will help in the forming of on-path telemetry requirements as well.

There is also current work to provide solutions for BIER forwarding in ipv6 networks. For instance, a solution, BIER in Non-MPLS IPv6 Networks [I-D.xie-bier-ipv6-encapsulation], proposes a new bier Option Type codepoint from the "Destination Options and Hop-by-Hop Options" IPv6 sub-registry. This is similar to what IOAM proposes for IPv6 transport.

Depending on how the BIER header is encapsulated into packets with different transport protocols, the method to encapsulate the telemetry instruction header and metadata also varies. It is also possible to make the instruction header and metadata a part of the BIER header itself, such as in a TLV.

6. Security Considerations

No new security issues are identified other than those discovered by the IOAM and PBT drafts.

7. IANA Considerations

The document makes no request of IANA.

8. Contributors

TBD

9. Acknowledgments

The authors would like to thank Frank Brockners, Tianran Zhou for the comments and advice.

10. References

10.1. 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>>.
- [RFC4687] Yasukawa, S., Farrel, A., King, D., and T. Nadeau, "Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks", RFC 4687, DOI 10.17487/RFC4687, September 2006, <<https://www.rfc-editor.org/info/rfc4687>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [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>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.

[RFC8487] Asaeda, H., Meyer, K., and W. Lee. Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", RFC 8487, DOI 10.17487/RFC8487, October 2018, <<https://www.rfc-editor.org/info/rfc8487>>.

10.2. Informative References

- [I-D.herbert-ipv4-udpencap-eh]
Herbert, T., "IPv4 Extension Headers and UDP Encapsulated Extension Headers", draft-herbert-ipv4-udpencap-eh-01 (work in progress), March 2019.
- [I-D.ietf-bier-oam-requirements]
Mirsky, G., Nainar, N., Chen, M., and S. Pallagatti, "Operations, Administration and Maintenance (OAM) Requirements for Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-oam-requirements-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ioametal-ippm-6man-ioam-ipv6-deployment]
Bhandari, S., Brockners, F., Mizrahi, T., Kfir, A., Gafni, B., Spiegel, M., Krishnan, S., and M. Smith, "Deployment Considerations for In-situ OAM with IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-deployment-03 (work in progress), March 2020.
- [I-D.ioamteam-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.
- [I-D.mirsky-ippm-hybrid-two-step]
Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-07 (work in progress), December 2020.
- [I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., and L. Andersson, "MPLS Extension Header", draft-song-mpls-extension-header-02 (work in progress), February 2019.

[I-D.xie-bier-ipv6-encapsulation]

Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-09 (work in progress), January 2021.

Authors' Addresses

Haoyu Song
Futurewei Technologies
2330 Central Expressway
Santa Clara
USA

Email: hsong@futurewei.com

Mike McBride
Futurewei Technologies
2330 Central Expressway
Santa Clara
USA

Email: mmcbride@futurewei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com