

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

F. Brockners
Cisco
S. Bhandari
Thoughtspot
T. Mizrahi
Huawei
February 21, 2021

Integrity of In-situ OAM Data Fields
draft-brockners-ippm-ioam-data-integrity-01

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 is to assist the IPPM WG in designing a solution for those deployments where the integrity of IOAM data fields is a concern. This document proposes several methods to ensure the integrity of IOAM data fields.

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 25, 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. Conventions 4
- 3. Threat Analysis 4
 - 3.1. Modification: IOAM Data Fields 5
 - 3.2. Modification: IOAM Option-Type Headers 5
 - 3.3. Injection: IOAM Data Fields 6
 - 3.4. Injection: IOAM Option-Type Headers 6
 - 3.5. Replay 6
 - 3.6. Management and Exporting 7
 - 3.7. Delay 7
 - 3.8. Threat Summary 8
- 4. Methods of providing integrity to IOAM data fields 8
 - 4.1. Method 1: Using asymmetric keys for signing node trace data 9
 - 4.1.1. Overhead consideration for Method 1 10
 - 4.2. Method 2: Using symmetric keys for signing node trace data 10
 - 4.2.1. Overhead consideration for Method 2 11
 - 4.3. Method 3: Space optimized symmetric key based signing of trace data 11
 - 4.3.1. Overhead consideration for Method 3 12
 - 4.4. Method 4: Dynamic symmetric keys based signing of trace data 12
 - 4.4.1. Overhead consideration for Method 4 14
 - 4.5. Method 5: Leverage IP Authentication Header 14
 - 4.5.1. Overhead consideration for Method 5 15
- 5. IANA Considerations 15
- 6. Security Considerations 16
- 7. Acknowledgements 16
- 8. References 16
 - 8.1. Normative References 16
 - 8.2. Informative References 16
- Authors' Addresses 18

1. Introduction

"In-situ" Operations, Administration, and Maintenance (IOAM) 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 is being sent within packets specifically dedicated to OAM. IOAM is to

complement mechanisms such as Ping, Traceroute, or other active probing mechanisms. 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 1. 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 live 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 current [I-D.ietf-ippm-ioam-data] assumes that IOAM is deployed in specific network domains, where an operator has means to select, monitor, and control the access to all the networking devices, making the domain a trusted network. As such, IOAM tracing data is carried in the packets in clear and there are no protections against any node or middlebox tampering with the data. As a consequence, IOAM tracing data collected in an untrusted or semi-trusted environments cannot be trusted for critical operational decisions. Any rogue or unauthorized change to IOAM data fields in a user packet cannot be detected.

Recent discussions following the IETF last call on [I-D.ietf-ippm-ioam-data] revealed that there might be uses of IOAM where integrity protection of IOAM data fields is at least desirable, knowing that IOAM data fields integrity protection would incur extra effort in the data path of a device processing IOAM data fields. As such, the following additional considerations and requirements are to be taken into account in addition to addressing the problem of detectability of any integrity breach of the IOAM trace data collected:

1. IOAM trace data is processed by the data plane, hence viability of any method to prove integrity of the IOAM trace data must be feasible at data plane processing/forwarding rates (IOAM data might be applied to all traffic a router forwards).
2. IOAM trace data is carried within data packets. Additional space required to prove integrity of the data needs to be optimal, i.e. should not exceed the MTU or have adverse affect on packet processing.
3. Replay protection of older IOAM trace data should be possible. Without replay protection a rogue node can present the old IOAM

trace data masking any ongoing network issues/activity making the IOAM trace data collection useless.

This document is to assist the IPPM working group in designing and specifying a solution for those deployments where the integrity of IOAM data fields is a concern. This document proposes several methods to achieve integrity protection for IOAM data fields.

The discussion of the different methods to protect the integrity of IOAM data fields focuses mostly on protecting the integrity of IOAM trace data fields, though the outlined methods are not limited to protecting IOAM trace data fields only. The methods could be applied to other IOAM Option-Types, such as the E2E Option-Type or the DEX Option-Type. If methods only apply to a specific set of IOAM Option-Types, like e.g. the method 5 discussed below, those limits are discussed and explained explicitly.

2. Conventions

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].

Abbreviations used in this document:

Geneve: Generic Network Virtualization Encapsulation
[I-D.ietf-nvo3-geneve]

GRE Generic Routing Encapsulation

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

NSH: Network Service Header [RFC8300]

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

SFC: Service Function Chain

3. Threat Analysis

This section presents a threat analysis of integrity-related threats in the context of IOAM. The threats that are discussed are assumed to be independent of the lower layer protocols; it is assumed that

threats at other layers are handled by security mechanisms that are deployed at these layers.

This document is focused on integrity protection for IOAM data fields. Thus the threat analysis includes threats that are related to or result from compromising the integrity of IOAM data fields. Other security aspects such as confidentiality are not within the scope of this document.

Throughout the analysis there is a distinction between on-path and off-path attackers. As discussed in [I-D.ietf-detnet-security], on-path attackers are located in a position that allows interception and modification of in-flight protocol packets, whereas off-path attackers can only attack by generating protocol packets.

The analysis also includes the impact of each of the threats. Generally speaking, the impact of a successful attack on an OAM protocol [RFC7276] is a false illusion of nonexistent failures or preventing the detection of actual ones; in both cases, the attack may result in denial of service (DoS). Furthermore, creating the false illusion of a nonexistent issue may trigger unnecessary processing in some of the IOAM nodes along the path, and may cause more IOAM-related data to be exported to the management plane than is conventionally necessary. Beyond these general impacts, threat-specific impacts are discussed in each of the subsections below.

3.1. Modification: IOAM Data Fields

Threat

An attacker can maliciously modify the IOAM data fields of in-transit packets. The modification can either be applied to all packets or selectively applied to a subset of the en route packets. This threat is applicable to on-path attackers.

Impact

By systematically modifying the IOAM data fields of some or all of the in-transit packets an attacker can create a false picture of the paths in the network, the existence of faulty nodes and their location, and the network performance.

3.2. Modification: IOAM Option-Type Headers

Threat

An on-path attacker can modify IOAM data fields in one or more of the IOAM Option-Type headers in order to change or disrupt the behavior of nodes processing IOAM data fields along the path.

Impact

Changing the header of IOAM Option-Types may have several implications. An attacker can maliciously increase the processing overhead in nodes that process IOAM data fields and increase the on-the-wire overhead of IOAM data fields, for example by modifying the IOAM-Trace-Type field in the IOAM Trace-option header. An attacker can also prevent some of the nodes that process IOAM data fields from incorporating IOAM data fields by modifying the RemainingLen field.

3.3. Injection: IOAM Data Fields

Threat

An attacker can inject packets with IOAM Option-Types and IOAM data fields. This threat is applicable to both on-path and off-path attackers.

Impact

This attack and its impacts are similar to Section 3.1.

3.4. Injection: IOAM Option-Type Headers

Threat

An attacker can inject packets with IOAM Option-Type headers, thus manipulating other nodes that process IOAM data fields in the network. This threat is applicable to both on-path and off-path attackers.

Impact

This attack and its impacts are similar to Section 3.2.

3.5. Replay

Threat

An attacker can replay packets with IOAM data fields. Specifically, an attacker may replay a previously transmitted IOAM Option-Type with a new data packet, thus attaching old IOAM data

fields to a fresh user packet. This threat is applicable to both on-path and off-path attackers.

Impact

As with previous threats, this threat may create a false image of a nonexistent failure, or may overload nodes which process IOAM data fields with unnecessary processing.

3.6. Management and Exporting

Threat

Attacks that compromise the integrity of IOAM data fields can be applied at the management plane, e.g., by manipulating network management packets. Furthermore, the integrity of IOAM data fields that are exported to a receiving entity can also be compromised. Management plane attacks are not within the scope of this document; the network management protocol is expected to include inherent security capabilities. The integrity of exported data is also not within the scope of this document. It is expected that the specification of the export format will discuss the relevant security aspects.

Impact

Malicious manipulation of the management protocol can cause nodes that process IOAM data fields to malfunction, to be overloaded, or to incorporate unnecessary IOAM data fields into user packets. The impact of compromising the integrity of exported IOAM data fields is similar to the impacts of previous threats that were described in this section.

3.7. Delay

Threat

An on-path attacker may delay some or all of the in-transit packets that include IOAM data fields in order to create the false illusion of congestion. Delay attacks are well known in the context of deterministic networks [I-D.ietf-detnet-security] and synchronization [RFC7384], and may be somewhat mitigated in these environments by using redundant paths in a way that is resilient to an attack along one of the paths. This approach does not address the threat in the context of IOAM, as it does not meet the requirement to measure a specific path or to detect a problem along the path. It is noted that this threat is not within the

scope of the threats that are mitigated in the scope of this document.

Impact

Since IOAM can be applied to a fraction of the traffic, an attacker can detect and delay only the packets that include IOAM data fields, thus preventing the authenticity of delay and load measurements.

3.8. Threat Summary

Threat	In scope	Out of scope
Modification: IOAM Data Fields	+	
Modification: IOAM Option-Type Headers	+	
Injection: IOAM Data Fields	+	
Injection: IOAM Option-Type Headers	+	
Replay	+	
Management and Exporting		+
Delay		+

Figure 1: Threat Analysis Summary

4. Methods of providing integrity to IOAM data fields

This section outlines four different methods that are to provide integrity protection of IOAM data fields. As noted earlier, this document is to support the IPPM working group in designing and specifying a method for protecting the integrity of IOAM data fields. It isn't expected that all four methods would be chosen for a solution specification.

As already stated in the introduction, the discussion of the different methods focuses mostly on protecting the integrity of IOAM trace data fields, though the outlined methods are not limited to protecting IOAM trace data fields only. The methods could be applied to other IOAM Option-Types, such as the E2E Option-Type or the DEX Option-Type.

IOAM trace data can be embedded in a variety of protocols. There are specific drafts that cover the encapsulation of IOAM data into different protocols, like IPv6 [I-D.ietf-ippm-ioam-ipv6-options], NSH [I-D.ietf-sfc-ioam-nsh], Geneve [I-D.brockners-ippm-ioam-geneve], etc.

The IOAM Option-Types for tracing (Pre-allocated Trace-Option and Incremental Trace-Option) organize the collected data in an array, the "node data list". See [I-D.ietf-ippm-ioam-data] for further details).

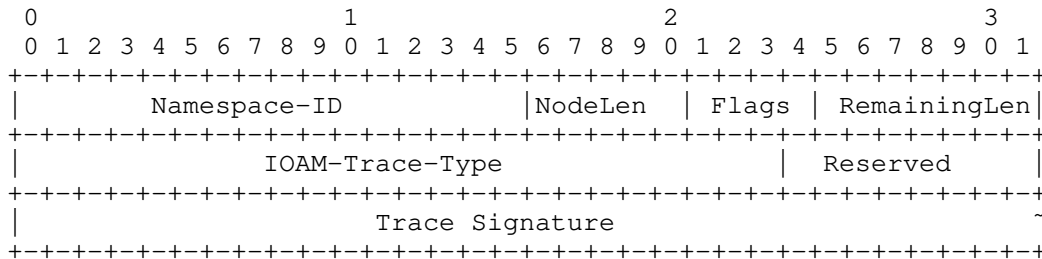
The basic idea is to introduce a new "signed node-data hash field" added by each node along with the node data to prove the integrity of the node data inserted.

The following sections describe different methods of how such a "signed node-data field" could be used and populated. The methods assume an IOAM-Domain containing IOAM-encapsulating nodes, IOAM-decapsulating nodes and IOAM-transit nodes. In addition, it is assumed that traffic also traverses a Validator node, which verifies the integrity of the IOAM data fields. In a typical deployment, the IOAM-decapsulating node would also serve as the Validator. The setup also includes a network management entity/controller which handles key distributions to the network nodes and also serves as a receiver for validation results provided by the Validator. Protocols and procedures for the exchange of keys and validation results between the network management entity/controller and the nodes are outside the scope of this document.

4.1. Method 1: Using asymmetric keys for signing node trace data

Method 1 uses asymmetric keys for signing node trace data. This is the procedure to be followed by each node:

1. Each IOAM capable node creates a key pair and shares the public key with the controller, the Validator and the network management system responsible for using the IOAM trace information in the network domain. The detailed mechanisms how keys are exchanged between nodes are outside the scope of this document. For optimal performance, use of algorithms like BLS [BLS] or ED25519 [EdDSA25519] are suggested, resulting in fast signing for small keys and limited overhead (see below for an overhead calculation).
2. Each node data list [x] field is extended with an additional "signed node-data" field: node_data_sign[x]. Node_data_sign includes a signature using the private key of the node over the hash of node data list[x] of the node and the previous node's



Method 3 assumes that symmetric keys have been distributed to the respective nodes as well as the Validator (the Validator receives all the keys). The details of the mechanisms of how keys are distributed are outside the scope of this document. The "Trace Signature" field is populated as follows:

1. The first node creates a seed and sign/HMAC over the hash of its node_data_list[x], the seed and its symmetric key. The seed can be included as a field in first node data or the seed can be the trailer to the trace option. The resulting HMAC/signature is included in the Trace Signature field.
2. Subsequent nodes will update the Trace Signature field by creating a signature/HMAC of data where the data is [Trace Signature || its node_data_list[x] hash] with its symmetric key.
3. The Validator will iteratively recreate the Trace Signature over the node data trace fields collected and matches the Trace Signature field to validate the trace data integrity.

4.3.1. Overhead consideration for Method 3

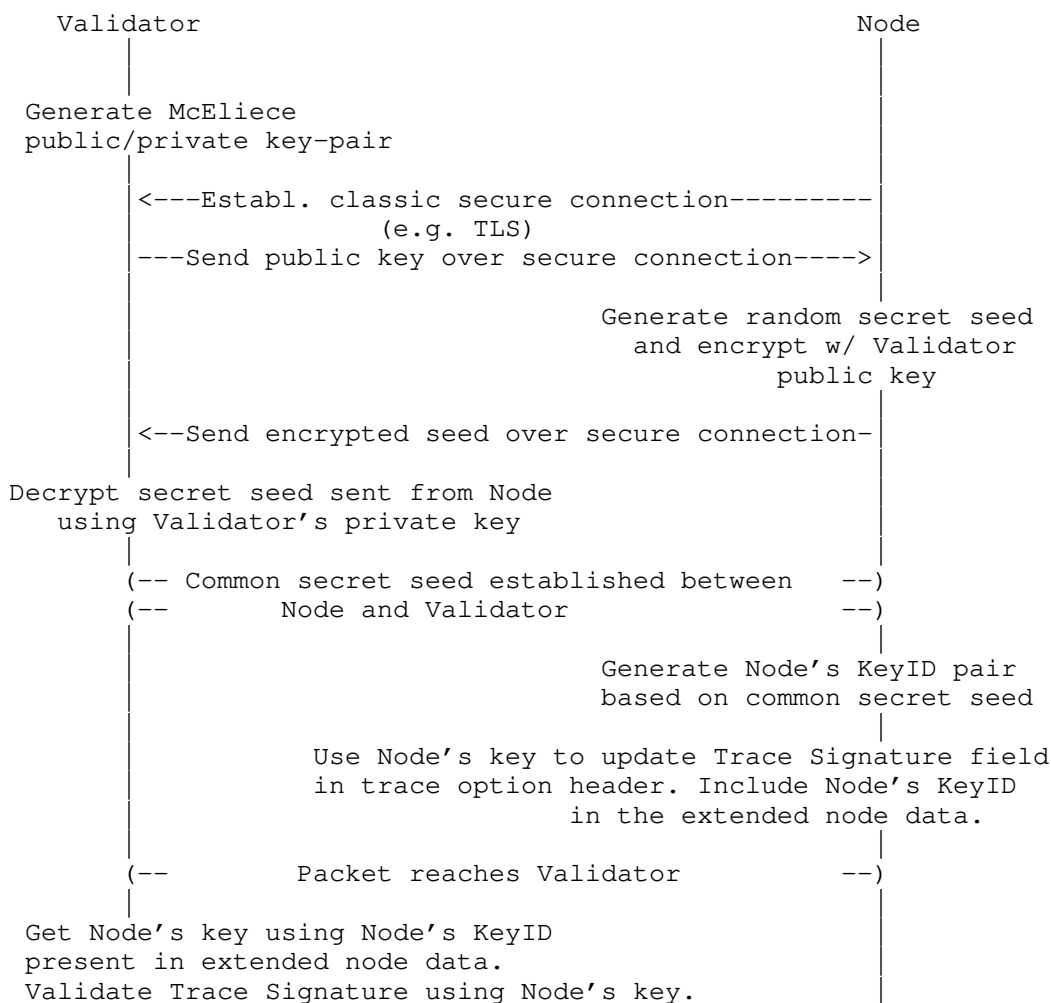
Much like method 2, the Trace Signature would consume 16 or 32 bytes - though with method 3, the Trace Signature is only carried once for the entire packet.

4.4. Method 4: Dynamic symmetric keys based signing of trace data

This method builds on top of Method 3 leverages Post-quantum Secure Pre-shared key distribution for deriving a dynamic symmetric key for every packet or a set of packets. The method utilizes the dynamic keys to provide for replay protection and does not require a seed to be added to the trace data to protect from replays because a private key is derived for each packet. The method relies on a local service that generates common Key/KeyID pairs for the participating Node and Validator (see the figure below). This common key generator uses

ratcheting cryptography to generate the next secret while forgetting about the previous one. A unique ID is paired with each secret generated. Given the same seed secret as input parameter, two implementations of the common key generator will generate the exact same key and associated ID. The common key generator can be queried for the next key or for a specific key ID.

The figure below illustrates the concept:



The main steps of method 4 are:

required to validate the integrity of the IOAM data. It is applicable to IOAM Option-Types as follows:

1. IOAM Edge-to-Edge Option-Type: The data for this IOAM Option-Type is immutable enroute and can be integrity checked using an IP AH between IOAM-encapsulating nodes and IOAM-decapsulating nodes.
2. The Direct Exporting (DEX) IOAM Option-Type ([I-D.ietf-ippm-ioam-direct-export]): The data for this IOAM Option-Type is immutable enroute and can be integrity checked using an IP AH between IOAM-encapsulating nodes and IOAM-decapsulating nodes. IOAM-transit nodes, that use the IOAM DEX Option-Type as a trigger to export IOAM data fields, and which are to validate the IOAM DEX Option-Type data fields may have to be configured with shared secrets, in case the AH negotiated integrity algorithm relies on shared secrets. These shared secrets correspond to those secrets that result from the IP AH integrity algorithm negotiated during SA setup.
3. IOAM Proof of Transit Option-Type: This IOAM Option-Type is mutable by IOAM-transit nodes enroute that participate in proof of transit operation. Hence the IP AH based integrity protection method does not apply.
4. IOAM Pre-allocated and Incremental Trace Option-Types: These IOAM Option-Types are mutable by IOAM-transit nodes encroute that process IOAM tracing Option-Types. Hence the IP AH based integrity protection method does not apply.

4.5.1. Overhead consideration for Method 5

This method involves overhead of setting up and maintaining SA for the AH IOAM-encapsulating nodes and IOAM-decapsulating nodes. The anti-replay mechanism supported by IP AH must be enabled for this method to effectively protect IOAM data fields whose integrity and freshness needs to be guaranteed. The anti-replay mechanism of IP AH has the overhead of maintaining and validating sequence numbers as part of the IP AH validation. In cases where IOAM-transit nodes have to validate IOAM Option-Types e.g. the DEX Option-Type, then there will be additional overhead to distribute shared secrets to the transit nodes when the integrity algorithm is based on shared secrets.

5. IANA Considerations

This document is to support the IPPM working group to design and specify a solution for protecting the integrity of IOAM data fields. It does not include any requests to IANA.

6. Security Considerations

This section will be completed in a future revision of this document.

7. Acknowledgements

The authors would like to thank Santhosh N, Rakesh Kandula, Saiprasad Muchala, Greg Mirsky, Benjamin Kaduk and Martin Duke for their comments and advice.

8. References

8.1. Normative References

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

8.2. Informative References

- [BLS] "BLS (Boneh-Lynn-Shacham) digital signature", 2021, <https://en.wikipedia.org/wiki/BLS_digital_signature>.
- [EdDSA25519] "Edwards-curve Digital Signature Algorithm (EdDSA)", 2021, <<https://en.wikipedia.org/wiki/EdDSA#Ed25519>>.
- [I-D.brockners-ippm-ioam-geneve] Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Nainar, N., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Lapukhov, P., Gafni, B., Kfir, A., and M. Spiegel, "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-05 (work in progress), November 2020.
- [I-D.ietf-detnet-security] Grossman, E., Mizrahi, T., and A. Hacker, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-13 (work in progress), December 2020.

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.
- [I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-16 (work in progress), March 2020.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-05 (work in progress), December 2020.
- [I-D.ietf-sfc-proof-of-transit]
Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S. Youell, "Proof of Transit", draft-ietf-sfc-proof-of-transit-08 (work in progress), November 2020.
- [McEliece]
McEliece, R., "A Public-Key Cryptosystem Based on Algebraic Coding Theory", 1978, <https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [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>>.
- [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>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

Authors' Addresses

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

Email: fbrockne@cisco.com

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

Email: shwetha.bhandari@thoughtspot.com

Tal Mizrahi
Huawei
8-2 Matam
Haifa 3190501
Israel

Email: tal.mizrahi.phd@gmail.com

opsawg
Internet-Draft
Intended status: Best Current Practice
Expires: March 11, 2021

F. Brockners
S. Bhandari
Cisco
D. Bernier
Bell Canada
September 7, 2020

In-situ OAM Deployment
draft-brockners-opsawg-ioam-deployment-02

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 provides a framework for IOAM deployment and provides best current practices.

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 March 11, 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	3
3.	IOAM Deployment: Domains And Nodes	4
4.	Types of IOAM	5
4.1.	Per-hop Tracing IOAM	6
4.2.	Proof of Transit IOAM	8
4.3.	Edge to Edge IOAM	8
4.4.	Direct Export IOAM	8
5.	IOAM Encapsulations	8
5.1.	IPv6	8
5.2.	NSH	9
5.3.	GRE	9
5.4.	Geneve	9
5.5.	Segment Routing	9
5.6.	Segment Routing for IPv6	10
5.7.	VXLAN-GPE	10
6.	IOAM Data Export	10
7.	IOAM Deployment Considerations	11
7.1.	IOAM Namespaces	11
7.2.	IOAM Layering	13
7.3.	IOAM Trace Option Types	14
7.4.	Traffic-sets That IOAM Is Applied To	16
7.5.	IOAM Loopback Mode	16
7.6.	IOAM Active Mode	16
7.7.	Brown Field Deployments: IOAM Unaware Nodes	17
8.	IOAM Manageability	17
9.	IANA Considerations	18
10.	Security Considerations	18
11.	Acknowledgements	19
12.	References	19
12.1.	Normative References	19
12.2.	Informative References	19
	Authors' Addresses	22

1. Introduction

"In-situ" Operations, Administration, and Maintenance (IOAM) 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 is being sent within packets specifically dedicated to OAM. IOAM is to complement mechanisms such as Ping, Traceroute, or other active probing mechanisms. 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 1. 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 live 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.

2. Conventions

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].

Abbreviations used in this document:

E2E	Edge to Edge
Geneve:	Generic Network Virtualization Encapsulation [I-D.ietf-nvo3-geneve]
GRE	Generic Routing Encapsulation
IOAM:	In-situ Operations, Administration, and Maintenance
MTU:	Maximum Transmit Unit
NSH:	Network Service Header [RFC8300]
OAM:	Operations, Administration, and Maintenance
POT:	Proof of Transit
SFC:	Service Function Chain
SID:	Segment Identifier
SR:	Segment Routing
VXLAN-GPE:	Virtual eXtensible Local Area Network, Generic Protocol Extension [I-D.ietf-nvo3-vxlan-gpe]

3. IOAM Deployment: Domains And Nodes

IOAM is a network domain specific feature, with "network domain" being a set of network devices or entities within a single administration. IOAM is not targeted for a deployment on the global Internet. The part of the network which employs IOAM is referred to as the "IOAM-Domain". For example, an IOAM-domain can include an enterprise campus using physical connections between devices or an overlay network using virtual connections / tunnels for connectivity between said devices. An IOAM-domain is defined by its perimeter or edge. The operator of an IOAM-domain is expected to put provisions in place to ensure that packets which contain IOAM data fields do not leak beyond the edge of an IOAM domain, e.g. 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.

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.

An "IOAM encapsulating node" incorporates one or more IOAM-Option-Types 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" updates 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 will update at most one of these Option-Types. A transit node does not add new IOAM-Option-Types to a packet, and does not change the IOAM-Data-Fields of an IOAM Edge-to-Edge Option-Type.

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

The role of an IOAM-encapsulating, IOAM-transit or IOAM-decapsulating node is always performed within a specific IOAM-Namespace. This means that an IOAM node which is e.g. an IOAM-decapsulating node for IOAM-Namespace "A" but not for IOAM-Namespace "B" will only remove the IOAM-Option-Types for IOAM-Namespace "A" from the packet. An IOAM decapsulating node situated at the edge of an IOAM domain

removes all IOAM-Option-Types and associated encapsulation headers for all IOAM-Namespaces from the packet.

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 7.1 for a discussion of IOAM-Namespaces.

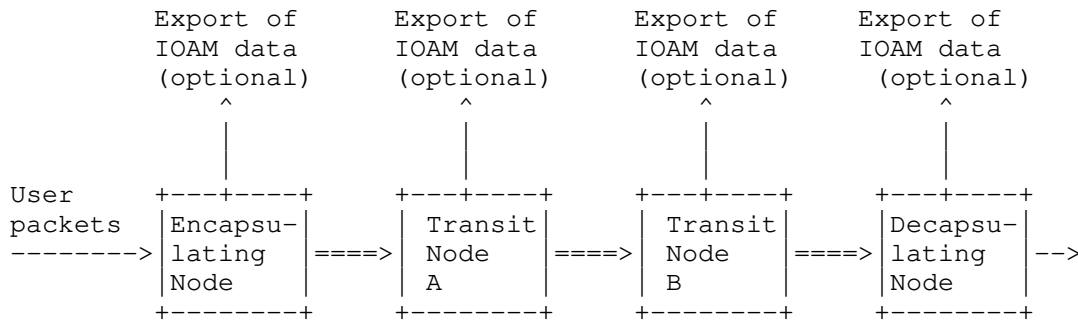


Figure 1: Roles of IOAM nodes

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.

4. Types of IOAM

IOAM supports different modes of operation, which are differentiated by the type of IOAM data fields being carried in the packet, the data being collected, the type of nodes which collect or update data as well as whether and how nodes export IOAM information.

- o Per-hop tracing: OAM information about each IOAM node a packet traverses is collected and stored within the user data packet as the packet progresses through the IOAM domain. Potential uses of IOAM per-hop tracing include:

- * Optimization: Understand the different paths different packets traverse between a source and a sink in a network that uses load balancing such as Equal Cost Load Balancing (ECMP). This information could be used to tune the algorithm for ECMP for optimized network resource usage.
- * Operations/Troubleshooting: Understand which path a particular packet or set of packets take as well as what amount of jitter and delay different nodes in the path contribute to the overall end-to-end delay and jitter.
- o Proof-of-transit: Information that a verifier node can use to verify whether a packet has traversed all nodes that is supposed to traverse is stored within the user data packet. Proof-of-transit could for example be used to verify that a packet indeed passes through all services of a service function chain (e.g. verify whether a packet indeed traversed the set of firewalls that it is expected to traverse), or whether a packet indeed took the expected path.
- o Edge-to-edge: OAM information which is specific to the edges of an IOAM domain is collected and stored within the user data packet. Edge-to-Edge OAM could be used to gather operational information about a particular network domain, such as the delay and jitter incurred by that network domain or the traffic matrix of the network domain.
- o Direct export: OAM information about each IOAM node a packet traverses is collected and immediately exported to a collector. Direct export could be used in situations where per-hop tracing information is desired, but gathering the information within the packet - as with per-hop tracing - is not feasible. Rather than automatically correlating the per-hop tracing information, as done with per-hop tracing, direct export requires a collector to correlate the information from the individual nodes. In addition, all nodes enabled for direct export need to be capable to export the IOAM information to the collector.

4.1. Per-hop Tracing IOAM

"IOAM tracing data" is expected to be collected at every IOAM transit node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM-Domain. I.e., 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 are IOAM capable, IOAM tracing information (i.e., node data, see below) will only be collected on those nodes which are IOAM capable. Nodes which are not

IOAM capable 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.

IOAM offers two different trace Option-Types, the "incremental" Option-Type as well as the "pre-allocated" Option-Type. For a discussion which of the two option types is the most suitable for an implementation and/or deployment, see Section 7.3.

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. All IOAM-Data-Fields 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 in-situ OAM 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 should 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.
- o Data that relates to how the packet traversed a node (transit delay, buffer occupancy in case the packet was buffered, queue depth in case the packet was queued)

The Option-Types of incremental tracing and pre-allocated tracing are defined in [I-D.ietf-ippm-ioam-data].

4.2. Proof of Transit IOAM

IOAM Proof of Transit Option-Type is to support path or service function chain [RFC7665] verification use cases. Proof-of-transit uses methods like nested hashing or nested encryption of the IOAM data or mechanisms such as Shamir's Secret Sharing Schema (SSSS).

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". For details see [I-D.ietf-ippm-ioam-data].

4.3. Edge to Edge IOAM

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". For details see [I-D.ietf-ippm-ioam-data].

4.4. Direct Export IOAM

Direct Export is an IOAM mode of operation within which IOAM data to be directly exported to a collector rather than being collected within the data packets. The IOAM Direct Export Option-Type consist of a fixed size "IOAM direct export option header". Direct Export for IOAM is defined in [I-D.ioamteam-ippm-ioam-direct-export].

5. IOAM Encapsulations

IOAM data fields and associated data types for in-situ OAM are defined in [I-D.ietf-ippm-ioam-data]. The in-situ OAM data field can be transported by a variety of transport protocols, including NSH, Segment Routing, Geneve, IPv6, etc.

5.1. IPv6

IOAM encapsulation for IPv6 is defined in [I-D.ietf-ippm-ioam-ipv6-options]. IOAM deployment considerations for IPv6 networks are found in [I-D.ioametal-ippm-6man-ioam-ipv6-deployment].

5.2. NSH

IOAM encapsulation for NSH is defined in [I-D.ietf-sfc-ioam-nsh].

5.3. GRE

IOAM encapsulation for GRE is outlined as part of the "EtherType Protocol Identification of In-situ OAM Data" in [I-D.weis-ippm-ioam-eth], though no example protocol header stacks are provided in the document. When used with GRE, the IOAM-Option-Types (the below diagram uses "IOAM" as shorthand for IOAM-Option-Types) are sequenced in behind the GRE header that follows the "outer" IP header. Figure 2 shows two example protocol header stacks that use GRE along with IOAM.

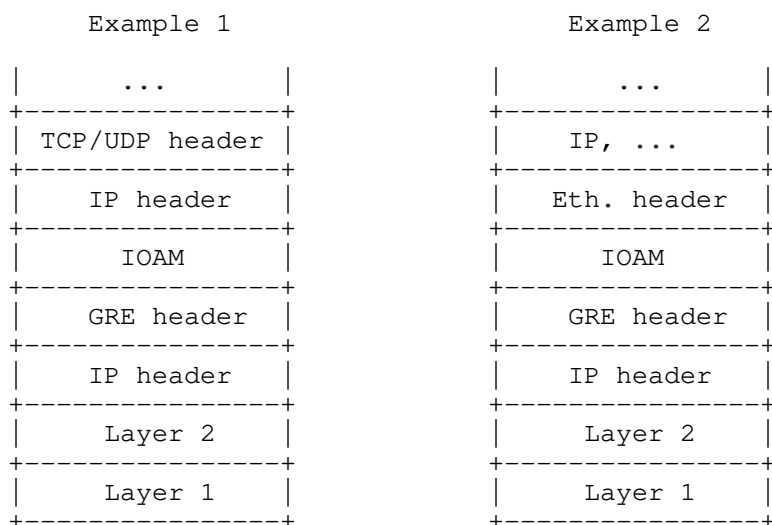


Figure 2: GRE with IOAM examples

5.4. Geneve

IOAM encapsulation for Geneve is defined in [I-D.brockners-ippm-ioam-geneve].

5.5. Segment Routing

IOAM encapsulation for Segment Routing is defined in [I-D.gandhi-spring-ioam-sr-mp].

5.6. Segment Routing for IPv6

IOAM encapsulation for Segment Routing over IPv6 is defined in [I-D.ali-spring-ioam-srv6].

5.7. VXLAN-GPE

IOAM encapsulation for VXLAN-GPE is defined in [I-D.brockners-ippm-ioam-vxlan-gpe].

6. 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.

Raw data export of IOAM data using IPFIX is discussed in [I-D.spiegel-ippm-ioam-rawexport]. "Raw export of IOAM data" refers to a mode of operation where a node exports the IOAM data as it is received in the packet. The exporting node neither interprets, aggregates nor reformats the IOAM data before it is exported. Raw export of IOAM data is to support an operational model where the processing and interpretation of IOAM data is decoupled from the operation of encapsulating/updating/decapsulating IOAM data, which is also referred to as IOAM data-plane operation. The figure below shows the separation of concerns for IOAM export: Exporting IOAM data is performed by the "IOAM node" which performs IOAM data-plane operation, whereas the interpretation of IOAM data is performed by one or several IOAM data processing systems. The separation of concerns is to off-load interpretation, aggregation and formatting of IOAM data from the node which performs data-plane operations. In other words, a node which is focused on data-plane operations, i.e. forwarding of packets and handling IOAM data will not be tasked to also interpret the IOAM data, but can leave this task to another system or a set of systems. For scalability reasons, a single IOAM node could choose to export IOAM data to several IOAM data processing systems. Similarly, there several monitoring systems or analytics systems can be used to further process the data received from the IOAM preprocessing systems. Figure 3 shows an overview of IOAM export, including IOAM data processing systems and monitoring/analytics systems.

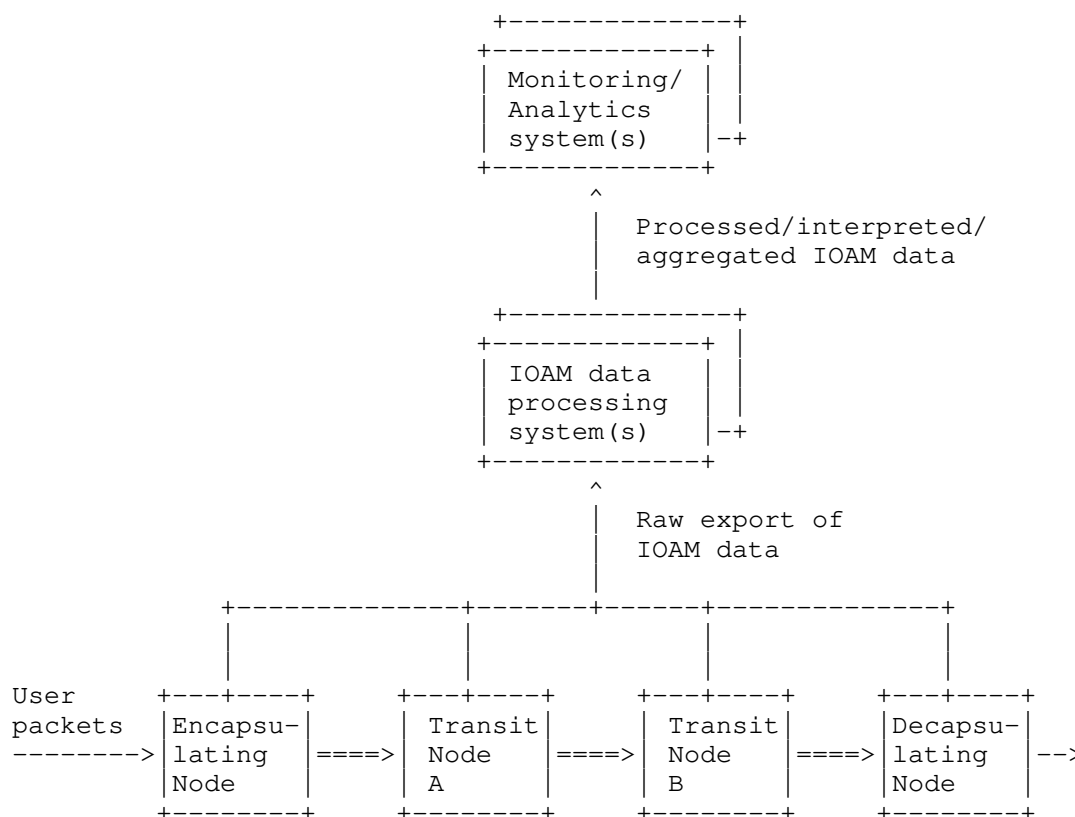


Figure 3: IOAM framework with data export

7. IOAM Deployment Considerations

This section discusses several aspects of an IOAM deployment, including IOAM Namespaces, IOAM Layering, traffic-sets that IOAM is applied to and IOAM loopback mode.

7.1. IOAM Namespaces

IOAM-Namespaces add further context to IOAM-Option-Types and associated IOAM-Data-Fields. IOAM-Namespaces support several different uses:

- o IOAM-Namespaces can be used by an operator to distinguish different operational domains. Devices at domain edges can filter on Namespace-IDs to provide for proper IOAM-Domain isolation.

- o IOAM-Namespaces provide additional context for IOAM-Data-Fields and thus ensure that IOAM-Data-Fields are unique and can be interpreted properly by management stations or network controllers. While, for example, the node identifier field does not need to be unique in a deployment (e.g. an operator may wish 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 combination of node_id and Namespace-ID will always be unique. 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.
- * Assigning different IOAM Namespace-IDs to different sets of nodes or network partitions and using the Namespace-ID as a selector at the IOAM encapsulating node, a full trace for a flow could be collected and constructed via partial traces in different packets of the same flow. Example: An operator could choose to group the devices of a domain into two IOAM-Namespaces, in a way that at average, only every second hop would be recorded by any device. To retrieve a full view of the deployment, the captured IOAM-Data-Fields of the two IOAM-Namespaces need to be correlated.
- * 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.

7.2. IOAM Layering

If several encapsulation protocols (e.g., in case of tunneling) are stacked on top of each other, IOAM-Data-Fields could be present in different protocol fields at different layers. Layering allows operators to instrument the protocol layer they want to measure. 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. Or in other words: Even though the term "layering" often implies some form of hierarchy and relationship, in IOAM, layers are independent from each other and don't assume any relationship among them. The different layers could, but do not have to share the same IOAM encapsulation mechanisms. Similarly, the semantics of the IOAM-Data-Fields, can, do not have to be associated to across different layers. For example, a node which inserts node-id information into two different layers could use "node-id=10" for one layer and "node-id=1000" for the second layer.

Figure 4 shows an example of IOAM layering. The figure shows a Geneve tunnel carried over IPv6 which starts at node A and ends at node D. IOAM information is encapsulated in IPv6 as well as in Geneve. At the IPv6 layer, node A is IOAM encapsulating node (into IPv6), node D is the IOAM decapsulating node and node B and node C are IOAM transit nodes. At the Geneve layer, node A is IOAM encapsulating node (into Geneve) and node D is IOAM decapsulating node (from Geneve). The use of IOAM at both layers as shown in the example here could be used to reveal which nodes of an underlay (here the IPv6 network) are traversed by tunneled packet in an overlay (here the Geneve network) - which assumes that the IOAM information encapsulated by nodes A and D into Geneve and IPv6 is associated to each other.

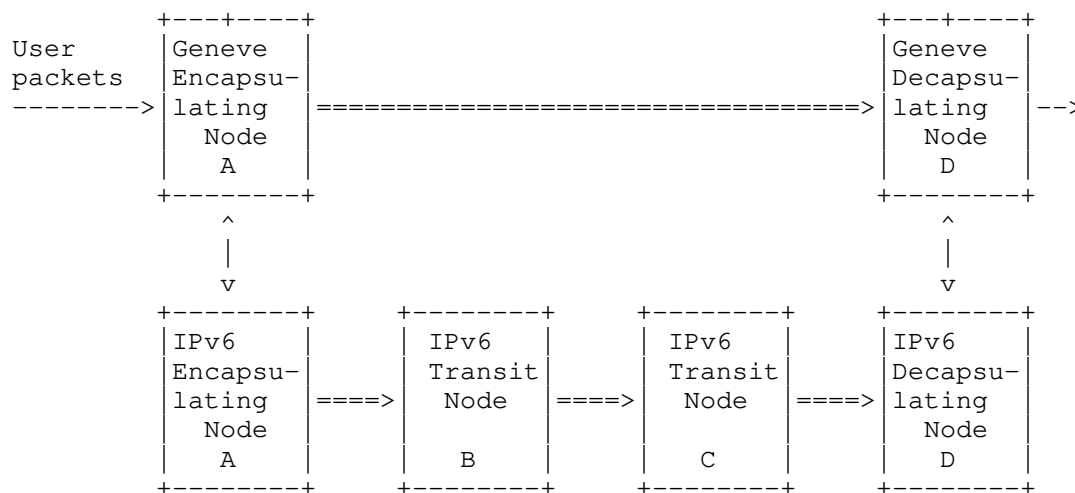


Figure 4: IOAM layering example

7.3. IOAM Trace Option Types

IOAM offers two different IOAM Option-Types for tracing: "Incremental" Trace-Option-Type and "Pre-allocated" Trace-Option-Type. "Incremental" refers to a mode of operation where the packet is expanded at every IOAM node that adds IOAM-Data-Fields. "Pre-Allocated" describes a mode of operation where the IOAM encapsulating node allocates room for all IOAM-Data-Fields in the entire IOAM domain. More specifically:

Pre-allocated Trace-Option: This trace option is defined as a container of node data fields 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).

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.

A deployment can choose to configure and support one or both of the IOAM Trace-Option-Types. The operator decides by means of configuration which Trace-Option-Type(s) will be used for a particular domain. Deployments can mix devices which include either the Incremental Trace-Option-Type or the Pre-allocated Trace-Option-Type, e.g. in case different types of packet forwarders and

associated different types of IOAM implementations exist in a deployment. As a result, both Option-Types can be present in a packet. IOAM decapsulating nodes remove both types of Trace-Option-Types from the packet.

The two different Option-Types cater to different packet forwarding infrastructures and are to allow an optimized implementation of IOAM tracing:

Pre-allocated Trace-Option: For some implementations of packet forwarders it is efficient to allocate the space for the maximum number of nodes that IOAM Trace Data-Fields should be collected from and insert/update information in the packet at flexible locations, based on a pointer retrieved from a field in the packet. The IOAM encapsulating node allocates an array of the size of the maximum number of nodes that IOAM Trace Data-Fields should be collected from. IOAM transit nodes index into the array to populate the data during transit. Software forwarders often fall into this class of packet forwarder implementations. The maximum number of nodes that IOAM information could be collected from is configured by the operator on the IOAM encapsulating node. The operator has to ensure that the packet with the pre-allocated array that carries the IOAM Data-Fields does not exceed the MTU of any of the links in the IOAM domain.

Incremental Trace-Option: Looking up a pointer contained in the packet and inserting/updating information at a flexible location in the packet as a result of the pointer lookup is costly for some forwarding infrastructures. Hardware-based packet forwarding infrastructures often fall into this category. Consequently, hardware-based packet forwarders could choose to support the incremental IOAM-Trace-Option-Type. The incremental IOAM-Trace-Option-Type eliminates the need for the IOAM transit nodes to read the full array in the Trace-Option-Type and allows packets to grow to the size of the MTU of the IOAM domain. IOAM transit nodes will expand the packet and insert the IOAM-Data-Fields as long as there is space available in the packet, i.e. as long as the size of the packet stays within the bounds of the MTU of any of the links in the IOAM domain. There is no need for the operator to configure the IOAM encapsulation node with the maximum number of nodes that IOAM information could be collected from. The operator has to ensure that the minimum MTU of any of the links in the IOAM domain is known to all IOAM transit nodes.

7.4. Traffic-sets That IOAM Is Applied To

IOAM can be deployed on all or only on subsets of the live user traffic, e.g. per interface, based on an access control list or flow specification defining a specific set of traffic, etc.

7.5. IOAM Loopback Mode

IOAM Loopback is used to trigger each transit device along the path of a packet to send a copy of the data packet back to the source. 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 enabled by the encapsulating node setting the loopback flag. Looped-back packets use the source address of the original packet as destination address and the address of the node which performs the loopback operation as source address. Nodes which loop back a packet clear the loopback flag before sending the copy back towards the source. Loopback applies to IOAM deployments where the encapsulating node is either a host or the start of a tunnel: For details on IOAM loopback, please refer to [I-D.ietf-ippm-ioam-flags].

7.6. IOAM Active Mode

The IOAM active mode flag indicates that a packet is an active OAM packet as opposed to regular user data traffic. Active mode is expected to be used for active measurement using IOAM. Example use-cases include:

- o Endpoint detailed active measurement: Synthetic probe packets are sent between the source and destination, traversing the IOAM domain. These probe packets include a Trace Option-Type (i.e., either incremental or pre-allocated). 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. The encapsulating node can choose to set the Active flag, providing an explicit indication that these probe packets are meant for telemetry collection.
- o IOAM active measurement using probe packets: Probe packets are generated and transmitted by the IOAM encapsulating node, and are expected to be terminated by the decapsulating node. Probe packets include a Trace Option-Type (i.e., either incremental or pre-allocated) 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-Type that has its Active flag set, indicating that the decapsulating node should terminate it.

For details on IOAM active mode, please refer to [I-D.ietf-ippm-ioam-flags].

7.7. Brown Field Deployments: IOAM Unaware Nodes

A network can consist of a mix of IOAM aware and IOAM unaware nodes. The encapsulation of IOAM-Data-Fields into different protocols (see also Section 5) are defined such that data packets that include IOAM-Data-Fields do not get dropped by IOAM unaware nodes. For example, packets which contain the IOAM-Trace-Option-Types in IPv6 Hop by Hop extension headers are defined with bits to indicate "00 - skip over this option and continue processing the header". This will ensure that when a node that is IOAM unaware receives a packet with IOAM-Data-Fields included, does not drop the packet.

Deployments which leverage the IOAM-Trace-Option-Type(s) could benefit from the ability to detect the presence of IOAM unaware nodes, i.e. nodes which forward the packet but do not update/add IOAM-Data-Fields in IOAM-Trace-Option-Type(s). The node data that is defined as part of the IOAM-Trace-Option-Type(s) includes a Hop_Lim field associated to the node identifier to detect missed nodes, i.e. "holes" in the trace. Monitoring/Analytics system(s) could utilize this information to account for the presence of IOAM unaware nodes in the network.

8. IOAM Manageability

The YANG model for configuring IOAM in network nodes which support IOAM is defined in [I-D.zhou-ippm-ioam-yang].

A deployment can leverage IOAM profiles is to limit the scope of IOAM features, allowing simpler implementation, verification, and interoperability testing in the context of specific use cases that do not require the full functionality of IOAM. An IOAM profile defines a use case or a set of use cases for IOAM, and an associated set of rules that restrict the scope and features of the IOAM specification, thereby limiting it to a subset of the full functionality. IOAM profiles are defined in [I-D.mizrahi-ippm-ioam-profile].

9. IANA Considerations

This document does not request any IANA actions.

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.

The Proof of Transit Option-Type (Section Section 4.2) is used for verifying the path of data packets. The security considerations of POT are further discussed in [I-D.ietf-sfc-proof-of-transit].

Security considerations related to the use of IOAM flags, in particular the loopback flag are found in [I-D.ietf-ippm-ioam-flags].

The data elements of IOAM can be used for network reconnaissance, allowing attackers to collect information about network paths, performance, queue states, buffer occupancy and other information. Note that in case IOAM is used in "immediate export" mode (reference to be added in a future revision), the IOAM related trace information would not be available in the customer data packets, but would trigger export of packet related IOAM information at every node. IOAM data export and securing IOAM data export is outside the scope of this document.

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 collectors that 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.

Since IOAM options may 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 may be implemented by misconfiguring or by maliciously configuring IOAM-enabled nodes in a way that enables other attacks. Thus, IOAM configuration should be secured in a way that authenticates authorized users and verifies the integrity of configuration procedures.

Notably, IOAM is expected to be deployed in specific network domains, thus confining the potential attack vectors to within the network domain. Indeed, in order to limit the scope of threats to within the current network 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. Note that the Immediate Export mode (reference to be added in a future revision) can mitigate the potential threat of IOAM data leaking through data packets.

11. Acknowledgements

The authors would like to thank Tal Mizrahi, Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Barak Gafni, Karthik Babu Harichandra Babu, Akshaya Nadahalli, LJ Wobker, Erik Nordmark, Vengada Prasad Govindan, Andrew Yourtchenko, Aviv Kfir, Tianran Zhou, Zhenbin (Robin), Joe Clarke, Al Morton, Tom Herbet, Haoyu song, and Mickey Spiegel for the comments and advice on IOAM.

12. References

12.1. 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>>.
- [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>>.

12.2. Informative References

- [I-D.ali-spring-ioam-srv6] Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.
- [I-D.brockners-ippm-ioam-geneve] Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Lapukhov, P., Gafni, B., Kfir, A., and M. Spiegel, "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-04 (work in progress), May 2020.

- [I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-vxlan-gpe-03 (work in progress), November 2019.
- [I-D.gandhi-spring-ioam-sr-mpls]
Gandhi, R., Ali, Z., Filmsfils, C., Brockners, F., Wen, B., and V. Kozak, "Segment Routing with MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-spring-ioam-sr-mpls-02 (work in progress), August 2019.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.
- [I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-ietf-ippm-ioam-flags-02 (work in progress), July 2020.
- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., and R. Asati, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-02 (work in progress), July 2020.
- [I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-16 (work in progress), March 2020.
- [I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN (VXLAN-GPE)", draft-ietf-nvo3-vxlan-gpe-10 (work in progress), July 2020.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-04 (work in progress), June 2020.

- [I-D.ietf-sfc-proof-of-transit]
Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S. Youell, "Proof of Transit", draft-ietf-sfc-proof-of-transit-06 (work in progress), June 2020.
- [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.mizrahi-ippm-ioam-profile]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., Zhou, T., and J. Lemon, "In Situ OAM Profiles", draft-mizrahi-ippm-ioam-profile-03 (work in progress), August 2020.
- [I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-03 (work in progress), March 2020.
- [I-D.weis-ippm-ioam-eth]
Weis, B., Brockners, F., Hill, C., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "EtherType Protocol Identification of In-situ OAM Data", draft-weis-ippm-ioam-eth-04 (work in progress), May 2020.
- [I-D.zhou-ippm-ioam-yang]
Zhou, T., Guichard, J., Brockners, F., and S. Raghavan, "A YANG Data Model for In-Situ OAM", draft-zhou-ippm-ioam-yang-08 (work in progress), July 2020.
- [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>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

Authors' Addresses

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

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 31, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
M. Chen
Huawei
B. Janssens
Colt
R. Foote
Nokia
April 29, 2021

Simple TWAMP (STAMP) Extensions for Segment Routing Networks
draft-gandhi-ippm-stamp-srpm-03

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) extensions for SR networks, for both SR-MPLS and SRv6 data planes by augmenting the optional extensions defined in RFC 8972.

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 October 31, 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. Conventions Used in This Document	3
2.1. Requirements Language	3
2.2. Abbreviations	3
2.3. Reference Topology	3
3. Destination Node Address TLV	4
4. Return Path TLV	5
4.1. Return Path Sub-TLVs	6
4.1.1. Return Path Control Code Sub-TLV	6
4.1.2. Return Address Sub-TLV	7
4.1.3. Return Segment List Sub-TLVs	8
5. Security Considerations	8
6. IANA Considerations	9
7. References	10
7.1. Normative References	10
7.2. Informative References	10
Acknowledgments	11
Authors' Addresses	11

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762] without the use of a control channel to pre-signal session parameters. [RFC8972] defines optional extensions for STAMP.

The STAMP test packets are transmitted along an IP path between a Session-Sender and a Session-Reflector to measure performance delay and packet loss along that IP path. It may be desired in SR networks that the same path (same set of links and nodes) between the Session-Sender and Session-Reflector is used for the STAMP test packets in both directions. This is achieved by using the STAMP [RFC8762] extensions for SR-MPLS and SRv6 networks specified in this document by augmenting the optional extensions defined in [RFC8972].

2. Conventions Used in This Document

2.1. Requirements Language

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

2.2. Abbreviations

MPLS: Multiprotocol Label Switching.

PM: Performance Measurement.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

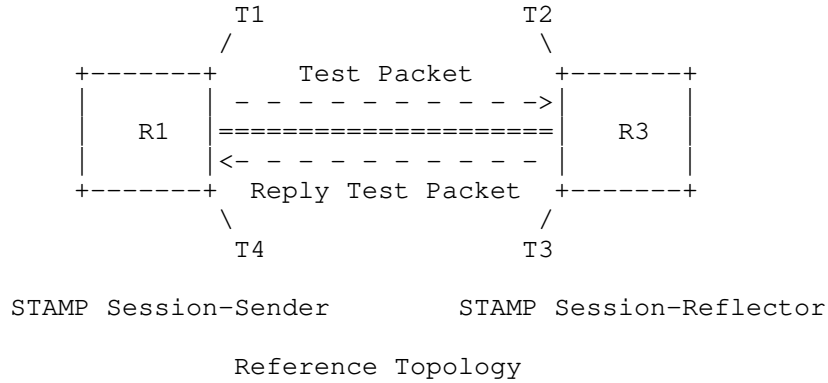
SSID: STAMP Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology shown below, the STAMP Session-Sender R1 initiates a STAMP test packet and the STAMP Session-Reflector R3 transmits a reply test packet. The reply test packet may be transmitted to the STAMP Session-Sender R1 on the same path (same set of links and nodes) or a different path in the reverse direction from the path taken towards the Session-Reflector.

The nodes R1 and R3 may be connected via a link or an SR path [RFC8402]. The link may be a physical interface, virtual link, or Link Aggregation Group (LAG) [IEEE802.1AX], or LAG member link. The SR path may be an SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R3 (called tail-end).



3. Destination Node Address TLV

The STAMP Session-Sender may need to transmit test packets to the STAMP Session-Reflector with a different destination address not matching an address on the Session-Reflector e.g. when the STAMP test packet is encapsulated by a tunneling protocol or an MPLS Segment List with IPv4 address from 127/8 range. In an error condition, the STAMP test packet may not reach the intended STAMP Session-Reflector, an un-intended node may transmit reply test packets resulting in reporting of invalid measurement metrics.

[RFC8972] defines STAMP test packets that can include one or more optional TLVs. In this document, Destination Node Address TLV (Type TBA1) is defined for STAMP test packet [RFC8972] and has the following format shown in Figure 1:

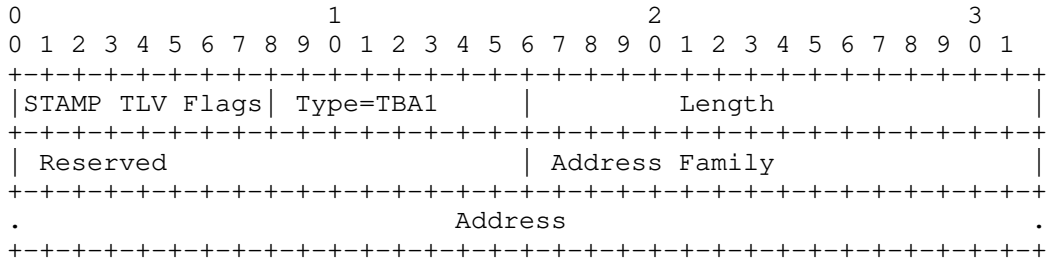


Figure 1: Destination Node Address TLV Format

The Address Family field indicates the type of the address, and it SHALL be set to one of the assigned values in the "IANA Address Family Numbers" registry.

The STAMP TLV Flags are set using the procedures described in [RFC8972].

The Destination Node Address TLV is optional. The Destination Node Address TLV indicates the address of the intended Session-Reflector node of the test packet. The STAMP Session-Reflector that supports this TLV, MUST NOT transmit reply test packet if it is not the intended destination node of the received Session-Sender test packet.

4. Return Path TLV

For end-to-end SR paths, the STAMP Session-Reflector may need to transmit the reply test packet on a specific return path. The STAMP Session-Sender can request this in the test packet to the STAMP Session-Reflector using a Return Path TLV. With this TLV carried in the STAMP Session-Sender test packet, signaling and maintaining dynamic SR network state for the STAMP sessions on the Session-Reflector are avoided.

For links, the STAMP Session-Reflector may need to transmit the reply test packet on the same incoming link in the reverse direction. The STAMP Session-Sender can request this in the test packet to the STAMP Session-Reflector using a Return Path TLV.

[RFC8972] defines STAMP test packets that can include one or more optional TLVs. In this document, the TLV Type (value TBA2) is defined for the Return Path TLV that carries the return path for the STAMP Session-Sender test packet. The format of the Return Path TLV is shown in Figure 2:

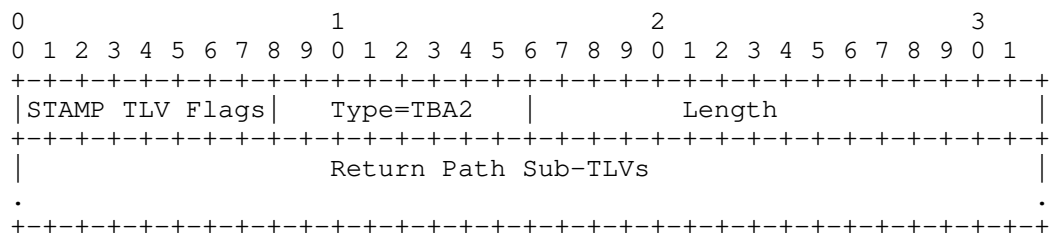


Figure 2: Return Path TLV

The STAMP TLV Flags are set using the procedures described in [RFC8972].

0x0: No Reply Requested.

0x1: Reply Requested on the Same Link.

When Control Code flag is set to 0x0 in the STAMP Session-Sender test packet, the Session-Reflector does not transmit reply test packet to the Session-Sender and terminates the STAMP test packet. Optionally, the Session-Reflector may locally stream performance metrics via telemetry using the information from the received test packet. All other Return Path Sub-TLVs are ignored in this case.

When Control Code flag is set to 0x1 in the STAMP Session-Sender test packet, the Session-Reflector transmits the reply test packet over the same incoming link where the test packet is received in the reverse direction towards the Session-Sender.

4.1.2. Return Address Sub-TLV

The STAMP reply test packet may be transmitted to a different node than the Session-Sender (e.g. to a controller for telemetry use-cases). For this, the Session-Sender can specify in the test packet the receiving destination node address for the Session-Reflector reply test packet.

The format of the Return Address Sub-TLV is shown in Figure 4. The Address Family field indicates the type of the address, and it SHALL be set to one of the assigned values in the "IANA Address Family Numbers" registry. The Type of the Return Address Sub-TLV is defined as following:

- o Type (value 2): Return Address. Destination node address of the STAMP Session-Reflector reply test packet different than the Source Address in the Session-Sender test packet.

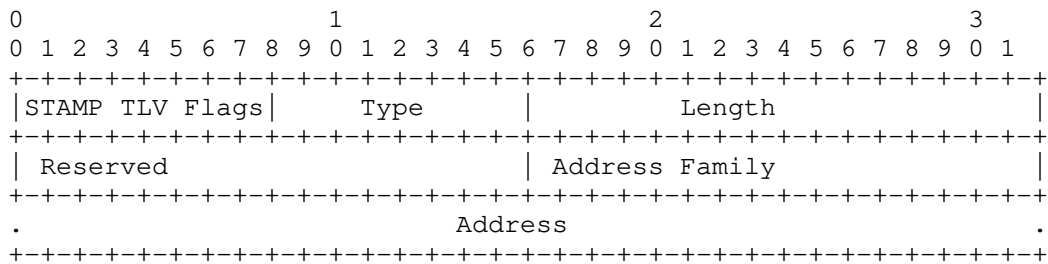


Figure 4: Return Address Sub-TLV in Return Path TLV

4.1.3. Return Segment List Sub-TLVs

The format of the Segment List Sub-TLVs in the Return Path TLV is shown in Figure 5. The segment entries MUST be in network order. The Segment List Sub-TLV can be one of the following Types:

- o Type (value 3): SR-MPLS Label Stack of the Return Path
- o Type (value 4): SRv6 Segment List of the Return Path

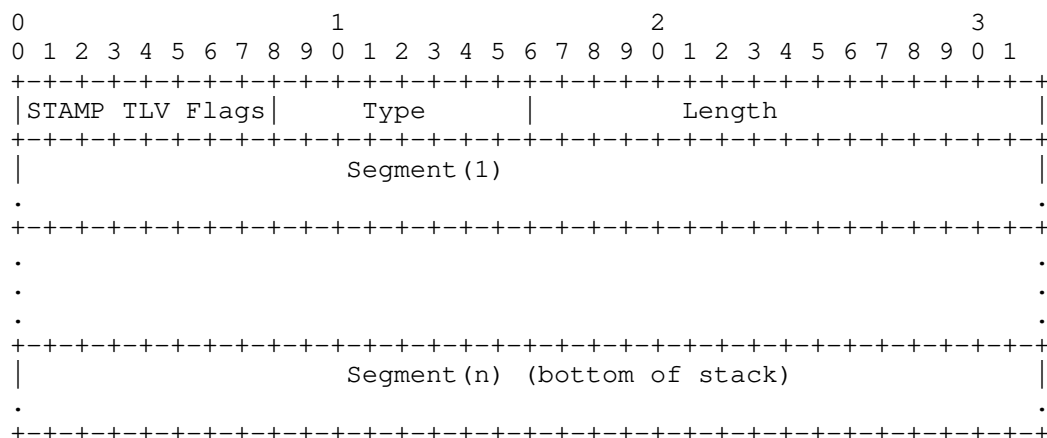


Figure 5: Segment List Sub-TLV in Return Path TLV

An SR-MPLS Label Stack Sub-TLV may carry only Binding SID [I-D.ietf-pce-binding-label-sid] of the Return SR-MPLS Policy.

An SRv6 Segment List Sub-TLV may carry only Binding SID [I-D.ietf-pce-binding-label-sid] of the Return SRv6 Policy.

The STAMP Session-Sender MUST only insert one Segment List Return Path Sub-TLV in the test packet. The STAMP Session-Reflector MUST only process the first Segment List Return Path Sub-TLV in the test packet and ignore other Segment List Return Path Sub-TLVs if present.

5. Security Considerations

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

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

The security considerations specified in [RFC8762] and [RFC8972] also apply to the extensions defined in this document.

6. IANA Considerations

IANA will create a "STAMP TLV Type" registry for [RFC8972]. IANA is requested to allocate a value for the following Destination Address TLV Type from the IETF Review TLV range of this registry. This TLV is to be carried in the STAMP test packets.

- o Type TBA1: Destination Node Address TLV

IANA is also requested to allocate a value for the following Return Path TLV Type from the IETF Review TLV range of the same registry. This TLV is to be carried in the STAMP test packets.

- o Type TBA2: Return Path TLV

IANA is requested to create a sub-registry for "Return Path Sub-TLV Type". 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]. 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: Return Path Sub-TLV Type Registry

IANA is requested to allocate the values for the following Sub-TLV Types from this registry.

- o Type (value 1): Return Path Control Code
- o Type (value 2): Return Address
- o Type (value 3): SR-MPLS Label Stack of the Return Path
- o Type (value 4): SRv6 Segment List of the Return Path

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>>.
- [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>>.
- [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

- [RFC8402] Filtsfils, 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>>.
- [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>>.

[I-D.ietf-spring-segment-routing-policy]

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

[I-D.ietf-pce-binding-label-sid]

Sivabalan, S., Filsfils, C., Tantsura, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment Identifier in PCE-based Networks.", draft-ietf-pce-binding-label-sid-08 (work in progress), April 2021.

[IEEE802.1AX]

IEEE Std. 802.1AX, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", November 2008.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky, Mike Koldychev, Gyan Mishra, Tianran Zhou, and Cheng Li for providing comments and suggestions.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach (Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

Richard Foote
Nokia

Email: footer.foote@nokia.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 28, 2021

A. Morton
AT&T Labs
R. Geib
Deutsche Telekom
L. Ciavattone
AT&T Labs
April 26, 2021

Metrics and Methods for One-way IP Capacity
draft-ietf-ippm-capacity-metric-method-10

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on October 28, 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	3
1.1. Requirements Language	4
2. Scope, Goals, and Applicability	4
3. Motivation	5
4. General Parameters and Definitions	6
5. IP-Layer Capacity Singleton Metric Definitions	7
5.1. Formal Name	7
5.2. Parameters	7
5.3. Metric Definitions	8
5.4. Related Round-Trip Delay and One-way Loss Definitions	9
5.5. Discussion	10
5.6. Reporting the Metric	10
6. Maximum IP-Layer Capacity Metric Definitions (Statistic)	10
6.1. Formal Name	10
6.2. Parameters	10
6.3. Metric Definitions	10
6.4. Related Round-Trip Delay and One-way Loss Definitions	12
6.5. Discussion	12
6.6. Reporting the Metric	13
7. IP-Layer Sender Bit Rate Singleton Metric Definitions	13
7.1. Formal Name	13
7.2. Parameters	14
7.3. Metric Definition	14
7.4. Discussion	15
7.5. Reporting the Metric	15
8. Method of Measurement	15
8.1. Load Rate Adjustment Algorithm	15
8.2. Measurement Qualification or Verification	20
8.3. Measurement Considerations	22
8.4. Running Code	23
9. Reporting Formats	24
9.1. Configuration and Reporting Data Formats	26
10. Security Considerations	26
11. IANA Considerations	27
12. Acknowledgments	27
13. Appendix A - Load Rate Adjustment Pseudo Code	27
14. Appendix B - RFC 8085 UDP Guidelines Check	28
14.1. Assessment of Mandatory Requirements	28
14.2. Assessment of Recommendations	30
15. References	33
15.1. Normative References	33
15.2. Informative References	34
Authors' Addresses	36

1. Introduction

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

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

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

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

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

Appendix A describes the load rate adjustment algorithm in pseudo-code. Appendix B discusses the algorithm's compliance with [RFC8085].

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, Goals, and Applicability

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.

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

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

The load rate adjustment algorithm's scope is limited to helping determine the Maximum IP-Layer Capacity in the context of an infrequent, diagnostic, short term measurement. It is RECOMMENDED to discontinue non-measurement traffic that shares a subscriber's dedicated resources while testing: measurements may not be accurate and throughput of competing elastic traffic may be greatly reduced.

The primary application of the metric and method of measurement described here is the same as in Section 2 of [RFC7497] where:

- o The access portion of the network is the focus of this problem statement. The user typically subscribes to a service with bidirectional access partly described by rates in bits per second.

In addition, the use of the load rate adjustment algorithm described in section 8.1 has the following additional applicability limitations:

- MUST only be used in the application of diagnostic and operations measurements as described in this memo
- MUST only be used in circumstances consistent with Section 10, Security Considerations
- If a network operator is certain of the access capacity to be validated, then testing MAY start with a fixed rate test at the access capacity and avoid activating the load adjustment algorithm. However, the stimulus for a diagnostic test (such as a subscriber request) strongly implies that there is no certainty and the load adjustment algorithm will be needed.

Further, the metric and method of measurement are intended for use where specific exact path information is unknown within a range of possible values:

- the subscriber's exact Maximum IP-Layer Capacity is unknown (which is sometimes the case; service rates can be increased due to upgrades without a subscriber's request, or to provide a surplus to compensate for possible underestimates of TCP-based testing.
- the size of the access bottleneck buffer is unknown.

Finally, the measurement system's load rate adjustment algorithm SHALL NOT be provided with the exact capacity value to be validated a priori. This restriction fosters a fair result, and removes an opportunity for bad actors to operate with knowledge of the "right answer".

3. Motivation

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

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

1. Internet access is no longer the bottleneck for many users.
2. Both transfer rate and latency are important to user's satisfaction.
3. UDP's growing role in Transport, in areas where TCP once dominated.

4. Content and applications are moving physically closer to users.
5. There is 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 MaxHops, the limit on the number of Hops a specific packet may visit as it traverses from the host at Src to the host at Dst (implemented in the TTL or Hop Limit).
- o T0, the time at the start of measurement interval, when packets are first transmitted from the Source.
- o I, the nominal duration of a measurement interval at the destination (default 10 sec)
- o dt, the nominal duration of m equal sub-intervals in I at the destination (default 1 sec)
- o dtn, the beginning boundary of a specific sub-interval, n, one of m sub-intervals in I
- o FT, the feedback time interval between status feedback messages communicating measurement results, sent from the receiver to control the sender. The results are evaluated throughout the test to determine how to adjust the current offered load rate at the sender (default 50ms)
- 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.

- o Type-P, the complete description of the test packets for which this assessment applies (including the flow-defining fields). Note that the UDP transport layer is one requirement for test packets specified below. Type-P is a parallel concept to "population of interest" defined in clause 6.1.1 of[Y.1540].
- o PM, a list of fundamental metrics, such as loss, delay, and reordering, and corresponding target performance threshold. At least one fundamental metric and target performance threshold MUST be supplied (such as One-way IP Packet Loss [RFC7680] equal to zero).

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

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

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

5. IP-Layer Capacity Singleton Metric Definitions

This section sets requirements for the singleton metric that supports the Maximum IP-Layer Capacity Metric definition in Section 6.

5.1. Formal Name

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

Note that Type-P depends on the chosen method.

5.2. Parameters

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

No additional Parameters are needed.

5.3. Metric Definitions

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

Define the IP-Layer Capacity, $C(T,dt,PM)$, to be the number of IP-Layer bits (including header and data fields) in packets that can be transmitted from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length. The IP-Layer Capacity depends on the Src and Dst hosts, the host addresses, and the path between the hosts.

The number of these IP-Layer bits is designated $n0[dt_n,dt_{n+1}]$ for a specific dt .

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

Anticipating a Sample of Singletons, the number of sub-intervals with duration dt MUST be set to a natural number m , so that $T+I = T + m*dt$ with $dt_{n+1} - dt_n = dt$ for $1 \leq n \leq m$.

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

Mathematically, this definition is represented as (for each n):

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

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 [RFC8468] from the Src host and correctly received by the Dst host during one contiguous sub-interval, dt in length, during the interval $[T, T+I]$,

- o $C(T,dt,PM)$ the IP-Layer Capacity, corresponds to the value of n_0 measured in any sub-interval beginning at dtn , divided by the length of sub-interval, dt .
- o PM represents other performance metrics [see section 5.4 below]; their measurement results SHALL be collected during measurement of IP-Layer Capacity and associated with the corresponding dtn for further evaluation and reporting.
- o all sub-intervals MUST be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.
- o The bit rate of the physical interface of the measurement devices MUST be higher than the smallest of the links on the path whose $C(T,I,PM)$ is to be measured (the bottleneck link).

Measurements according to these definitions SHALL use the UDP transport layer. Standard-formed packets are specified in Section 5 of [RFC8468]. The measurement SHOULD use a randomized Source port or equivalent technique, and SHOULD send responses from the Source address matching the test packet destination address.

Some compression affects on measurement are discussed in Section 6 of [RFC8468].

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

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

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

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

5.5. Discussion

See the corresponding section for Maximum IP-Layer Capacity.

5.6. Reporting the Metric

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

The related One-way Loss metric and Round Trip Delay measurements for the same Singleton SHALL be reported, also with meaningful resolution for the values measured.

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

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

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

6.1. Formal Name

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

Note that Type-P depends on the chosen method.

6.2. Parameters

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

No additional Parameters or definitions are needed.

6.3. Metric Definitions

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

Define the Maximum IP-Layer Capacity, $Maximum_C(T,I,PM)$, to be the maximum number of IP-Layer bits $n0[dt_n, dt_n+1]$ divided by dt 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

- o all sub-intervals MUST be of equal duration. Choosing dt as non-overlapping consecutive time intervals allows for a simple implementation.
- o The bit rate of the physical interface of the measurement systems MUST be higher than than the smallest of the links on the path whose Maximum_C(T,I,PM) is to be measured (the bottleneck link).

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

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

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

RTD[dt_n,dt_{n+1}] and OWL[dt_n,dt_{n+1}] are defined in Section 5.4. Here, the test intervals are increased to match the capacity Samples, RTD[T,I] and OWL[T,I].

The interval dt_n,dt_{n+1} where Maximum_C[T,I,PM] occurs is the reporting sub-interval within RTD[T,I] and OWL[T,I].

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

6.5. Discussion

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

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

6.6. Reporting the Metric

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

The related One-way Loss metric and Round Trip Delay measurements for the same Singleton SHALL be reported, also with meaningful resolution for the values measured.

When there are demonstrated and repeatable Capacity modes in the Sample, then the Maximum IP-Layer Capacity SHALL be reported for each mode, along with the relative time from the beginning of the stream that the mode was observed to be present. Bimodal Maximum IP-Layer Capacities have been observed with some services, sometimes called a "turbo mode" intending to deliver short transfers more quickly, or reduce the initial buffering time for some video streams. Note that modes lasting less than dt duration will not be detected.

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

7. IP-Layer Sender Bit Rate Singleton Metric Definitions

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

7.1. Formal Name

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

Note that Type-P depends on the chosen method.

7.2. Parameters

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

- o S , the duration of the measurement interval at the Source
- o st , the nominal duration of N sub-intervals in S (default $st = 0.05$ seconds)
- o stn , the beginning boundary of a specific sub-interval, n , one of N sub-intervals in S

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

st SHOULD be much smaller than the sub-interval dt and on the same order as FT , otherwise the rate measurement will include many rate adjustments and include more time smoothing, thus missing the Maximum IP-Layer Capacity. 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 with address pair Src and Dst 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, $[stn, stn+1]$.

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

7.4. Discussion

Both the Sender and Receiver or (Source and Destination) bit rates SHOULD be assessed as part of an IP-Layer Capacity measurement. Otherwise, an unexpected sending rate limitation could produce an erroneous Maximum IP-Layer Capacity measurement.

7.5. Reporting the Metric

The IP-Layer Sender Bit Rate SHALL be reported with meaningful resolution, in units of Megabits per second (which is 1,000,000 bits per second to avoid any confusion).

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

8. Method of Measurement

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

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

8.1. Load Rate Adjustment Algorithm

The algorithm described in this section MUST NOT be used as a general Congestion Control Algorithm (CCA). As stated in the Scope Section 2, the load rate adjustment algorithm's goal is to help determine the Maximum IP-Layer Capacity in the context of an infrequent, diagnostic, short term measurement. There is a tradeoff between test duration (also the test data volume) and algorithm aggressiveness (speed of ramp-up and down to the Maximum IP-Layer Capacity). The parameter values chosen below strike a well-tested balance among these factors.

A table SHALL be pre-built defining all the offered load rates that will be supported (R1 through Rn, in ascending order, corresponding to indexed rows in the table). It is RECOMMENDED that rates begin with 0.5 Mbps at index zero, use 1 Mbps at index one, and then continue in 1 Mbps increments to 1 Gbps. Above 1 Gbps, and up to 10 Gbps, it is RECOMMENDED that 100 Mbps increments be used. Above 10 Gbps, increments of 1 Gbps are RECOMMENDED. A higher initial IP-Layer Sender Bitrate might be configured when the test operator is certain that the Maximum IP-Layer Capacity is well-above the initial

IP-Layer Sender Bitrate and factors such as test duration and total test traffic play an important role.

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

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

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

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

If the feedback indicates that no sequence number anomalies were detected AND the delay range was below the lower threshold, the offered load rate is increased. If congestion has not been confirmed up to this point, the offered load rate is increased by more than one rate (e.g., *Rx+10*). This allows the offered load to quickly reach a near-maximum rate. Conversely, if congestion has been previously confirmed, the offered load rate is only increased by one (*Rx+1*). However, if a rate threshold between high and very high sending rates (such as 1 Gbps) is exceeded, the offered load rate is only increased by one (*Rx+1*) above the rate threshold in any congestion state.

If the feedback indicates that sequence number anomalies were detected OR the delay range was above the upper threshold, the offered load rate is decreased. The RECOMMENDED values are 0 for sequence number gaps and 30 ms for lower and 90 ms for upper delay thresholds, respectively. Also, if congestion is now confirmed for the first time by the current feedback message being processed, then the offered load rate is decreased by more than one rate (e.g., *Rx-30*). This one-time reduction is intended to compensate for the fast

initial ramp-up. In all other cases, the offered load rate is only decreased by one (Rx-1).

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

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

The load rate adjustment algorithm MUST include timers that stop the test when received packet streams cease unexpectedly. The timeout thresholds are provided in the table below, along with values for all other parameters and variables described in this section. Operation of non-obvious parameters appear below:

load packet timeout Operation: The load packet timeout SHALL be reset to the configured value each time a load packet received. If the timeout expires, the receiver SHALL be closed and no further feedback sent.

feedback message timeout Operation: The feedback message timeout SHALL be reset to the configured value each time a feedback message is received. If the timeout expires, the sender SHALL be closed and no further load packets sent.

Parameter	Default	Tested Range or values	Expected Safe Range (not entirely tested, other values NOT RECOMMENDED)
FT, feedback time interval	50ms	20ms, 50ms, 100ms	20ms <= FT <= 250ms Larger values may slow the rate increase and fail to find the max
Feedback message timeout	L*FT, L=20 (1sec with FT=50ms)	L=100 with FT=50ms (5sec)	0.5sec <= L*FT <= 30sec Upper limit for very unreliable test

(stop test)			paths only
load packet timeout (stop test)	1sec	5sec	0.250sec - 30sec Upper limit for very unreliable test paths only
table index 0	0.5Mbps	0.5Mbps	when testing <=10Gbps
table index 1	1Mbps	1Mbps	when testing <=10Gbps
table index (step) size	1Mbps	1Mbps<=rate<=1Gbps	same as tested
table index (step) size, rate>1Gbps	100Mbps	1Gbps<=rate<=10Gbps	same as tested
table index (step) size, rate>10Gbps	1Gbps	untested	>10Gbps
ss, UDP payload size, bytes	none	<=1222	Recommend max at largest value that avoids fragmentation; use of too-small payload size might result in unexpected sender limitations.
cc, burst count	none	1<=cc<= 100	same as tested. Vary cc as needed to create the desired maximum sending rate. Sender buffer size may limit cc in implementation.
tt, burst interval	100microsec	100microsec, 1msec	available range of "tick" values (HZ param)
low delay range	30ms	5ms, 30ms	same as tested

threshold			
high delay range threshold	90ms	10ms, 90ms	same as tested
sequence error threshold	0	0, 100	same as tested
consecutive errored status report threshold	2	2	Use values >1 to avoid misinterpreting transient loss
Fast mode increase, in table index steps	10	10	2 <= steps <= 30
Fast mode decrease, in table index steps	3 * Fast mode increase	3 * Fast mode increase	same as tested

Parameters for Load Rate Adjustment Algorithm

As a consequence of default parameterization, the Number of table steps in total for rates <10Gbps is 2000 (excluding index 0).

A related sender backoff response to network conditions occurs when one or more status feedback messages fail to arrive at the sender.

If no status feedback messages arrive at the sender for the interval greater than the Lost Status Backoff timeout:

$$\text{UDRT} + (2+w)*\text{FT} = \text{Lost Status Backoff timeout}$$

where:

UDRT = upper delay range threshold (default 90ms)

FT = feedback time interval (default 50ms)

w = number of repeated timeouts (w=0 initially, w++ on each timeout, and reset to 0 when a message is received)

beginning when the last message (of any type) was successfully received at the sender:

Then the offered load SHALL be decreased, following the same process as when the feedback indicates presence of one or more sequence number anomalies OR the delay range was above the upper threshold (as described above), with the same load rate adjustment algorithm variables in their current state. This means that rate reduction and congestion confirmation can result from a three-way OR that includes lost status feedback messages, sequence errors, or delay variation.

The RECOMMENDED initial value for w is 0, taking Round Trip Time (RTT) less than FT into account. A test with RTT longer than FT is a valid reason to increase the initial value of w appropriately. Variable w SHALL be incremented by 1 whenever the Lost Status Backoff timeout is exceeded. So with FT = 50ms and UDRT = 90ms, a status feedback message loss would be declared at 190ms following a successful message, again at 50ms after that (240ms total), and so on.

Also, if congestion is now confirmed for the first time by a Lost Status Backoff timeout, then the offered load rate is decreased by more than one rate (e.g., Rx-30). This one-time reduction is intended to compensate for the fast initial ramp-up. In all other cases, the offered load rate is only decreased by one (Rx-1).

Appendix B discusses compliance with the applicable mandatory requirements of [RFC8085], consistent with the goals of the IP-Layer Capacity Metric and Method, including the load rate adjustment algorithm described in this section.

8.2. Measurement Qualification or Verification

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

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

There are many ways that the Method of Measurement could handle this false-max issue. The default value for measurement of singletons (dt = 1 second) has proven to 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 Maximum 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 Maximum IP-Layer Capacity for I, or an indefinite period. The same Maximum 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. The combination of a fast and congestion-aware search method and user-network coordination make a unique contribution to production testing. The Maximum IP Capacity metric and method for assessing performance is very different from classic [RFC2544] Throughput metric and methods : it uses near-real-time load adjustments that are sensitive to loss and delay, similar to other congestion control algorithms used on the Internet every day, along with limited duration. On the other hand, [RFC2544] Throughput measurements can produce sustained overload conditions for extended periods of time. Individual trials in a test governed by a binary search can last 60 seconds for each step, and the final confirmation trial may be even longer. This is very different from "normal" traffic levels, but overload conditions are not a concern in the isolated test environment. The concerns raised in [RFC6815] were that [RFC2544] methods would be let loose on production networks, and instead the authors challenged the standards community to develop metrics and methods like those described in this memo.

8.3. Measurement Considerations

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

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

The path measured may be state-full based on many factors, and the Parameter "Time of day" when a test starts may not be enough information. Repeatable testing may require the time from the beginning of a measured flow, and how the flow is constructed including how much traffic has already been sent on that flow when a state-change is observed, because the state-change may be based on time or bytes sent or both. Both load packets and status feedback messages MUST contain sequence numbers, which helps with measurements based on those packets.

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

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

1. Congestion of an interconnection or backbone interface may appear as packet losses distributed over time in the test stream, due to much higher rate interfaces in the backbone.
2. Packet loss due to use of Random Early Detection (RED) or other active queue management may or may not affect the measurement flow if competing background traffic (other flows) are simultaneously present.
3. There may be only small delay variation independent of sending rate under these conditions, too.

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

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

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

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

1. the test hosts may be able to create higher load than with a single flow, or parallel test hosts may be used to generate 1 flow each.
2. there may be link aggregation present (flow-based load balancing) and multiple flows are needed to occupy each member of the aggregate.
3. access policies may limit the IP-Layer Capacity depending on the Type-P of packets, possibly reserving capacity for various stream types.

Each flow would be controlled using its own implementation of the load rate adjustment (search) algorithm.

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

8.4. Running Code

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

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

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

9. Reporting Formats

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

- o timestamp (especially the time when the maximum was observed in dtn)
- o Source and Destination (by IP or other meaningful ID)
- o other inner parameters of the test case (Section 4)
- o outer parameters, such as "test conducted in motion" or other factors belonging to the context of the measurement
- o result validity (indicating cases where the process was somehow interrupted or the attempt failed)
- o a field where unusual circumstances could be documented, and another one for "ignore/mask out" purposes in further processing

The Maximum IP-Layer Capacity results SHOULD be reported in the format of a table with a row for each of the test Phases and Number

of Flows. There SHOULD be columns for the phases with number of flows, and for the resultant Maximum IP-Layer Capacity results for the aggregate and each flow tested.

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

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

Maximum IP-layer Capacity Results

Static and configuration parameters:

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

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

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

Phase, Flow or Aggregate	st, sec	Sender Bitrate, Mbps
Search,1	0.00 - 0.05	345
Search,2	0.00 - 0.05	289
Search,Agg	0.00 - 0.05	634

IP-layer Sender Bit Rate Results

Static and configuration parameters:

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

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

9.1. Configuration and Reporting Data Formats

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

10. Security Considerations

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

When considering privacy of those involved in measurement or those whose traffic is measured, the sensitive information available to potential observers is greatly reduced when using active techniques which are within this scope of work. Passive observations of user traffic for measurement purposes raise many privacy issues. We refer the reader to the privacy considerations described in the Large Scale Measurement of Broadband Performance (LMAP) Framework [RFC7594], which covers active and passive techniques.

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

1. Cooperating Source and Destination hosts and agreements to test the path between the hosts are REQUIRED. Hosts perform in either the Src or Dst roles.
2. It is REQUIRED to have a user client-initiated setup handshake between cooperating hosts that allows firewalls to control inbound unsolicited UDP traffic which either goes to a control port [expected and w/authentication] or to ephemeral ports that

are only created as needed. Firewalls protecting each host can both continue to do their job normally.

3. Client-server authentication and integrity protection for feedback messages conveying measurements is RECOMMENDED.
4. Hosts MUST limit the number of simultaneous tests to avoid resource exhaustion and inaccurate results.
5. Senders MUST be rate-limited. This can be accomplished using a pre-built table defining all the offered load rates that will be supported (Section 8.1). The recommended load-control search algorithm results in "ramp-up" from the lowest rate in the table.
6. Service subscribers with limited data volumes who conduct extensive capacity testing might experience the effects of Service Provider controls on their service. Testing with the Service Provider's measurement hosts SHOULD be limited in frequency and/or overall volume of test traffic (for example, the range of I duration values SHOULD be limited).

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

11. IANA Considerations

This memo makes no requests of IANA.

12. Acknowledgments

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

13. Appendix A - Load Rate Adjustment Pseudo Code

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

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

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

14. Appendix B - RFC 8085 UDP Guidelines Check

The BCP on UDP usage guidelines [RFC8085] focuses primarily on congestion control in section 3.1. The Guidelines appear in mandatory (MUST) and recommendation (SHOULD) categories.

14.1. Assessment of Mandatory Requirements

The mandatory requirements in Section 3 of [RFC8085] include:

Internet paths can have widely varying characteristics, ... Consequently, applications that may be used on the Internet MUST NOT make assumptions about specific path characteristics. They MUST instead use mechanisms that let them operate safely under very different path conditions. Typically, this requires conservatively probing the current conditions of the Internet path they communicate over to establish a transmission behavior that it can sustain and that is reasonably fair to other traffic sharing the path.

The purpose of the load rate adjustment algorithm in Section 8.1 is to probe the network and enable Maximum IP-Layer Capacity measurements with as few assumptions about the measured path as possible, and within the range application described in Section 2. The degree of probing conservatism is in tension with the need to minimize both the traffic dedicated to testing (especially with Gigabit rate measurements) and the duration of the test (which is one contributing factor to the overall algorithm fairness).

The text of Section 3 of [RFC8085] goes on to recommend alternatives to UDP to meet the mandatory requirements, but none are suitable for the scope and purpose of the metrics and methods in this memo. In fact, ad hoc TCP-based methods fail to achieve the measurement accuracy repeatedly proven in comparison measurements with the running code [LS-SG12-A] [LS-SG12-B] [Y.Sup60]. Also, the UDP aspect of these methods is present primarily to support modern Internet transmission where a transport protocol is required [copycat]; the metric is based on the IP-Layer and UDP allows simple correlation to the IP-Layer.

Section 3.1.1 of [RFC8085] discusses protocol timer guidelines:

Latency samples MUST NOT be derived from ambiguous transactions. The canonical example is in a protocol that retransmits data, but subsequently cannot determine which copy is being acknowledged.

Both load packets and status feedback messages MUST contain sequence numbers, which helps with measurements based on those packets, and there are no retransmissions needed.

When a latency estimate is used to arm a timer that provides loss detection -- with or without retransmission -- expiry of the timer MUST be interpreted as an indication of congestion in the network, causing the sending rate to be adapted to a safe conservative rate...

The method described in this memo uses timers for sending rate backoff when status feedback messages are lost (Lost Status Backoff

timeout), and for stopping a test when connectivity is lost for an longer interval (Feedback message or load packet timeouts).

There is no specific benefit foreseen by using Explicit Congestion Notification (ECN) in this memo.

Section 3.2 of [RFC8085] discusses message size guidelines:

To determine an appropriate UDP payload size, applications MUST subtract the size of the IP header (which includes any IPv4 optional headers or IPv6 extension headers) as well as the length of the UDP header (8 bytes) from the PMTU size.

The method uses a sending rate table with a maximum UDP payload size that anticipates significant header overhead and avoids fragmentation.

Section 3.3 of [RFC8085] provides reliability guidelines:

Applications that do require reliable message delivery MUST implement an appropriate mechanism themselves.

The IP-Layer Capacity Metric and Method do not require reliable delivery.

Applications that require ordered delivery MUST reestablish datagram ordering themselves.

The IP-Layer Capacity Metric and Method does not need to reestablish packet order; it is preferred to measure packet reordering if it occurs [RFC4737].

14.2. Assessment of Recommendations

The load rate adjustment algorithm's goal is to determine the Maximum IP-Layer Capacity in the context of an infrequent, diagnostic, short term measurement. This goal is a global exception to many [RFC8085] SHOULD-level requirements, of which many are intended for long-lived flows that must coexist with other traffic in more-or-less fair way. However, the algorithm (as specified in Section 8.1 and Appendix A above) reacts to indications of congestion in clearly defined ways.

A specific recommendation is provided as an example. Section 3.1.5 of [RFC8085] on implications of RTT and Loss Measurements on Congestion Control says:

A congestion control designed for UDP SHOULD respond as quickly as possible when it experiences congestion, and it SHOULD take into

account both the loss rate and the response time when choosing a new rate.

The load rate adjustment algorithm responds to loss and RTT measurements with a clear and concise rate reduction when warranted, and the response makes use of direct measurements (more exact than can be inferred from TCP ACKs).

Section 3.1.5 of [RFC8085] goes on to specify:

The implemented congestion control scheme SHOULD result in bandwidth (capacity) use that is comparable to that of TCP within an order of magnitude, so that it does not starve other flows sharing a common bottleneck.

This is a requirement for coexistent streams, and not for diagnostic and infrequent measurements using short durations. The rate oscillations during short tests allow other packets to pass, and don't starve other flows.

Ironically, ad hoc TCP-based measurements of "Internet Speed" are also designed to work around this SHOULD-level requirement, by launching many flows (9, for example) to increase the outstanding data dedicated to testing.

The load rate adjustment algorithm cannot become a TCP-like congestion control, or it will have the same weaknesses of TCP when trying to make a Maximum IP-Layer Capacity measurement, and will not achieve the goal. The results of the referenced testing [LS-SG12-A] [LS-SG12-B] [Y.Sup60] supported this statement hundreds of times, with comparisons to multi-connection TCP-based measurements.

A brief review of some of the other SHOULD-level requirements follows (Yes or Not applicable = NA) :

Y?	RFC 8085 Recommendation	Section
Yes	MUST tolerate a wide range of Internet path conditions	3
NA	SHOULD use a full-featured transport (e.g., TCP)	
Yes	SHOULD control rate of transmission	3.1
NA	SHOULD perform congestion control over all traffic	
	for bulk transfers,	3.1.2
NA	SHOULD consider implementing TFRC	
NA	else, SHOULD in other ways use bandwidth similar to TCP	

	for non-bulk transfers,	3.1.3
NA	SHOULD measure RTT and transmit max. 1 datagram/RTT	3.1.1
NA	else, SHOULD send at most 1 datagram every 3 seconds	
NA	SHOULD back-off retransmission timers following loss	
Yes	SHOULD provide mechanisms to regulate the bursts of transmission	3.1.6
NA	MAY implement ECN; a specific set of application mechanisms are REQUIRED if ECN is used.	3.1.7
Yes	for DiffServ, SHOULD NOT rely on implementation of PHBs	3.1.8
Yes	for QoS-enabled paths, MAY choose not to use CC	3.1.9
Yes	SHOULD NOT rely solely on QoS for their capacity non-CC controlled flows SHOULD implement a transport circuit breaker MAY implement a circuit breaker for other applications	3.1.10
	for tunnels carrying IP traffic,	3.1.11
NA	SHOULD NOT perform congestion control	
NA	MUST correctly process the IP ECN field	
	for non-IP tunnels or rate not determined by traffic,	3.1.11
NA	SHOULD perform CC or use circuit breaker	
NA	SHOULD restrict types of traffic transported by the tunnel	
Yes	SHOULD NOT send datagrams that exceed the PMTU, i.e.,	3.2
Yes	SHOULD discover PMTU or send datagrams < minimum PMTU;	
NA	Specific application mechanisms are REQUIRED if PLPMTUD is used.	
Yes	SHOULD handle datagram loss, duplication, reordering	3.3
NA	SHOULD be robust to delivery delays up to 2 minutes	
Yes	SHOULD enable IPv4 UDP checksum	3.4
Yes	SHOULD enable IPv6 UDP checksum; Specific application mechanisms are REQUIRED if a zero IPv6 UDP checksum is used.	3.4.1
NA	SHOULD provide protection from off-path attacks	5.1
	else, MAY use UDP-Lite with suitable checksum coverage	3.4.2
NA	SHOULD NOT always send middlebox keep-alive messages	3.5
NA	MAY use keep-alives when needed (min. interval 15 sec)	

Yes	Applications specified for use in limited use (or controlled environments) SHOULD identify equivalent mechanisms and describe their use case.	3.6
NA	Bulk-multicast apps SHOULD implement congestion control	4.1.1
NA	Low volume multicast apps SHOULD implement congestion control	4.1.2
NA	Multicast apps SHOULD use a safe PMTU	4.2
Yes	SHOULD avoid using multiple ports	5.1.2
Yes	MUST check received IP source address	
NA	SHOULD validate payload in ICMP messages	5.2
Yes	SHOULD use a randomized source port or equivalent technique, and, for client/server applications, SHOULD send responses from source address matching request	6
NA	SHOULD use standard IETF security protocols when needed	6

15. References

15.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <<https://www.rfc-editor.org/info/rfc4656>>.

- [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>>.
- [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>>.
- [RFC7497] Morton, A., "Rate Measurement Test Protocol Problem Statement and Requirements", RFC 7497, DOI 10.17487/RFC7497, April 2015, <<https://www.rfc-editor.org/info/rfc7497>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8468] Morton, A., Fabini, J., Elkins, N., Ackermann, M., and V. Hegde, "IPv4, IPv6, and IPv4-IPv6 Coexistence: Updates for the IP Performance Metrics (IPPM) Framework", RFC 8468, DOI 10.17487/RFC8468, November 2018, <<https://www.rfc-editor.org/info/rfc8468>>.

15.2. Informative References

- [copycat] Edleine, K., Kuhlewind, K., Trammell, B., and B. Donnet, "copycat: Testing Differential Treatment of New Transport Protocols in the Wild (ANRW '17)", July 2017, <<https://irtf.org/anrw/2017/anrw17-final5.pdf>>.
- [LS-SG12-A] 12, I. S., "LS - Harmonization of IP Capacity and Latency Parameters: Revision of Draft Rec. Y.1540 on IP packet transfer performance parameters and New Annex A with Lab Evaluation Plan", May 2019, <<https://datatracker.ietf.org/liaison/1632/>>.

- [LS-SG12-B] 12, I. S., "LS on harmonization of IP Capacity and Latency Parameters: Consent of Draft Rec. Y.1540 on IP packet transfer performance parameters and New Annex A with Lab & Field Evaluation Plans", March 2019, <<https://datatracker.ietf.org/liaison/1645/>>.
- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/info/rfc2544>>.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC 3148, DOI 10.17487/RFC3148, July 2001, <<https://www.rfc-editor.org/info/rfc3148>>.
- [RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <<https://www.rfc-editor.org/info/rfc5136>>.
- [RFC6815] Bradner, S., Dubray, K., McQuaid, J., and A. Morton, "Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful", RFC 6815, DOI 10.17487/RFC6815, November 2012, <<https://www.rfc-editor.org/info/rfc6815>>.
- [RFC7312] Fabini, J. and A. Morton, "Advanced Stream and Sampling Framework for IP Performance Metrics (IPPM)", RFC 7312, DOI 10.17487/RFC7312, August 2014, <<https://www.rfc-editor.org/info/rfc7312>>.
- [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<https://www.rfc-editor.org/info/rfc7594>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.

- [RFC8337] Mathis, M. and A. Morton, "Model-Based Metrics for Bulk Transport Capacity", RFC 8337, DOI 10.17487/RFC8337, March 2018, <<https://www.rfc-editor.org/info/rfc8337>>.
- [TR-471] Morton, A., "Broadband Forum TR-471: IP Layer Capacity Metrics and Measurement", July 2020, <<https://www.broadband-forum.org/technical/download/TR-471.pdf>>.
- [udpst] udpst Project Collaborators, "UDP Speed Test Open Broadband project", December 2020, <<https://github.com/BroadbandForum/obudpst>>.
- [Y.1540] Y.1540, I. R., "Internet protocol data communication service - IP packet transfer and availability performance parameters", December 2019, <<https://www.itu.int/rec/T-REC-Y.1540-201912-I/en>>.
- [Y.Sup60] Morton, A., "Recommendation Y.Sup60, (09/20) Interpreting ITU-T Y.1540 maximum IP-layer capacity measurements, and Errata", September 2020, <<https://www.itu.int/rec/T-REC-Y.Sup60/en>>.

Authors' Addresses

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

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

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

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

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

Email: lencia@att.com

ippm
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

R. Geib, Ed.
Deutsche Telekom
February 22, 2021

A Connectivity Monitoring Metric for IPPM
draft-ietf-ippm-connectivity-monitoring-01

Abstract

Within a Segment Routing domain, segment routed measurement packets can be sent along pre-determined paths. This enables new kinds of measurements. Connectivity monitoring allows to supervise the state and performance of a connection or a (sub)path from one or a few central monitoring systems. This document specifies a suitable type-P connectivity monitoring metric.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 26, 2021.

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
1.1.	Requirements Language	5
2.	A brief segment routing connectivity monitoring framework . .	5
3.	Topology and measurement loop set up requirements	9
3.1.	General network topology requirements	9
3.2.	Sub-path Monitoring measurement loop routing requirements	10
3.3.	Path	11
4.	Generic Type-P-SR-Path-Periodic-* metric	11
4.1.	Metric Name	12
4.2.	Generic Metric Parameters	12
4.3.	Metric Units	12
5.	Singleton Definition for Type-P-SR-Path-Periodic-Delay . . .	12
5.1.	Metric Name	12
5.2.	Metric Parameters	12
5.3.	Delay Metric Units	12
5.4.	Definition	13
5.5.	Discussion	13
5.6.	Methodologies	13
5.7.	Errors and Uncertainties	13
5.8.	Reporting the metric	13
6.	Definition of Samples for Type-P-SR-Path-Periodic-Delay . . .	13
6.1.	Generic Type-P-SR-Path-Periodic-Delay-* metric	13
6.1.1.	Metric Name	14
6.1.2.	Metric Parameters	14
6.1.3.	Metric Units	14
6.1.4.	Metric Definition	14
6.1.5.	Discussion	14
6.1.6.	Errors and uncertainties	14
6.2.	Definition of Type-P-SR-Path-Periodic-Delay-Stream . . .	14
6.2.1.	Metric Name	15
6.3.	Definition of Type-P-SR-Path-Periodic-Delay-Variation . .	15
6.3.1.	Metric Name	15
6.3.2.	Methodologies	15
6.3.3.	Discussion of SRDV	15
6.3.4.	Errors and uncertainties	15
6.4.	Definition of Type-P-SR-Path-Periodic-Delay-Variation- Stream	15
6.4.1.	Metric Name	15
6.4.2.	Metric Definition	16
7.	Statistic Definitions for SR-Path-Periodic-*--Stream samples .	16
7.1.	SR-Path-Periodic-*--Mean	16
7.2.	SR-Path-Periodic-*--Std	16
8.	Sub-Path monitoring metrics derived from samples captured	

along the measurement loops	17
8.1. Baseline measurement	17
8.2. Discussion of the baseline measurement	18
8.3. Definition of SR-Path-Sub-Path-RTD-Estimate	18
8.4. Definition of SR-Path-Sub-Path-*-Changepoint	19
8.5. Discussion of SR-Path-Sub-Path-*-Changepoint	19
8.6. Definition of SR-Path-Sub-Path-Congestion-Location	20
8.7. Discussion of SR-Path-Sub-Path-*-Location	21
9. Discussion of Temporal Resolution	21
10. IANA Considerations	22
11. Security Considerations	22
12. References	22
12.1. Normative References	22
12.2. Informative References	23
Author's Address	24

1. Introduction

Within a Segment Routing domain, measurement packets can be sent along pre-determined segment routed paths [RFC8402]. A segment routed path may consist of pre-determined sub paths, specific router-interfaces or a combination of both. A measurement path may also consist of sub paths spanning multiple routers, given that all segments to address a desired path are available and known at the SR domain edge interface.

A Path Monitoring System or PMS (see [RFC8403]) is a dedicated central Segment Routing (SR) domain monitoring device (as compared to a distributed monitoring approach based on router-data and -functions only). Monitoring individual sub-paths or point-to-point connections is executed for different purposes. IGP exchanges hello messages between neighbors to keep alive routing and swiftly adapt routing to topology changes. Network Operators may be interested in monitoring connectivity and congestion of interfaces or sub-paths at a timescale of seconds, minutes or hours. In both cases, the periodicity is significantly smaller than commodity interface monitoring based on router counters, which may be collected on a minute timescale to keep the processor- or monitoring data-load low.

The IPPM architecture was a first step to that direction [RFC2330]. Commodity IPPM solutions require dedicated measurement systems, a large number of measurement agents and synchronised clocks. Monitoring a domain from edge to edge by commodity IPPM solutions increases scalability of the monitoring system. But localising the site of a detected change in network behaviour may then require network tomography methods.

The IPPM Metrics for Measuring Connectivity offer generic connectivity metrics [RFC2678]. These metrics allow to measure connectivity between end nodes without making any assumption on the paths between them. The metric and the type-p packet specified by this document follow a different approach: they are designed to monitor connectivity and performance of a specific single link or a path segment. The underlying definition of connectivity is partially the same: a packet not reaching a destination indicates a loss of connectivity. An IGP re-route may indicate a loss of a link, while it might not cause loss of connectivity between end systems. The metric specified here detects a link-loss, if the change in end-to-end delay along a new route is differing from that of the original path.

A Segment Routing PMS is part of an SR domain. The PMS is IGP topology aware, covering the IP and (if present) the MPLS layer topology [RFC8402]. This allows to steer PMS measurement packets along arbitrary pre-determined concatenated sub-paths, identified by suitable Segment IDs. Basically, the SR connectivity metric as specified by this document requires set up of a number of constrained, overlaid measurement loops (or measurement paths). The delay of the packets sent along each of these measurement loops is measured. A single congested interface or a single loss of connectivity of a monitored sub-path cause a delay change on several measurement paths. Any single evnet of that type on one of the monitored sub-paths changes delays of a unique subset of measurement loops. The number of measurement loops may be limited to one per sub-path (or connection) to be monitored, if a hub-and-spoke like sub-path topology as described below is monitored. In addition to information revealed by a commodity ICMP ping measurement, the metrics and methods specified here identify the location of a congested interface. To do so, tomography assumptions and methods are combined to first plan the overlaid SR measurement loop set up and later on to evaluate the captured delay measurements.

There's another difference as compared with commodity ping: the measurement loop packets remain in the data plane of passed routers. These need to forward the measurement packets without additional processing apart from that.

It is recommended to consider automated measurement loop set-ups. The methods proposed here are error-prone if the topology and measurement loop design isn't followed properly. While details of an automated set-up are not within scope of this document, some formal definitions of constraints to respected are given.

This document specifies a type-p metric determining properties of an SR path which allows to monitor connectivity and congestion of

interfaces and further allows to locate the path or interface which caused a change in the reported type-p metric. This document is limited to the Segment Routing MPLS layer, but the methodology may be applied within SR domains or MPLS domains in general.

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 [RFC2119].

2. A brief segment routing connectivity monitoring framework

The Segment Routing IGP topology information consists of the IP and (if present) the MPLS layer topology. The minimum SR topology information consists of Node-Segment-Identifiers (Node-SID), identifying an SR router. The IGP exchange of Adjacency-SIDs [RFC8667], which identify local interfaces to adjacent nodes, is optional. It is RECOMMENDED to distribute Adj-SIDs in a domain operating a PMS to monitor connectivity as specified below. If Adj-SIDs aren't available, [RFC8287] provides methods how to steer packets along desired paths by the proper choice of an MPLS Echo-request IP-destination address. A detailed description of [RFC8287] methods as a replacement of Adj-SIDs is out of scope of this document.

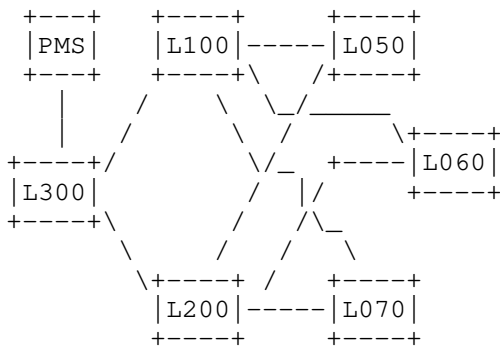
An active round trip measurement between two adjacent nodes is a simple method to monitor connectivity of a connecting link. If multiple links are operational between two adjacent nodes and only a single one fails, a single plain round trip measurement may fail to notice that or identify which link has failed. A round trip measurement also fails to identify which interface is congested, even if only a single link connects two adjacent nodes.

Segment Routing enables the set-up of extended measurement loops. Several different measurement loops can be set up to form a partial overlay. If done properly, any network change impacts more than a single measurement loop's round trip delay (or causes drops of packets of more than one loop). Randomly chosen measurement loop paths including the interfaces or paths to be monitored may fail to produce the desired unique result patterns, hence commodity network tomography methods aren't applicable here [CommodityTomography]. The approach pursued here uses a pre-specified measurement loop overlay design.

A centralised monitoring approach doesn't require report collection and result correlation from two (or more) receivers (the measured delays of different measurement loops still need to be correlated).

An additional property of the measurement path set-up specified below is that it allows to estimate the packet round trip and the one way delay of a monitored sub-path. The delay along a single link is not perfectly symmetric. Packet processing causes small delay differences per interface and direction. These cause an error, which can't be quantified or removed by the specified method. Quantifying this error requires a different measurement set-up. As this will introduce additional measurements loops, packets and evaluations, the cost in terms of reduced scalability is not felt to be worth the benefit in measurement accuracy. IPPM metrics prefer precision to accuracy and the mentioned processing differences are relatively stable, resulting in relatively precise delay estimates for each monitored sub-path.

An example hub and spoke network, operated as SR domain, is shown below. The included PMS shown is supposed to monitor the connectivity of all 6 links (a very generic kind of sub-path) attaching the spoke-nodes L050, L060 and L070 to the hub-nodes L100 and L200.



Hub and spoke connectivity verification with a PMS

Figure 1

The SID values are picked for convenient reading only. Node-SID: 100 identifies L100, Node-SID: 300 identifies L300 and so on. Adj-SID 10050: Adjacency L100 to L050, Adj-SID 10060: Adjacency L100 to L060, Adj-SID 60200: Adjacency L60 to L200 and so on (note that the Adj-SID are locally assigned per node interface, meaning two per link).

Monitoring the 6 links between hub nodes Ln00 (where n=1,2) and spoke nodes L0m0 (where m=5,6,7) requires 6 measurement loops, which have the following properties:

- o Each measurement loop follows a single round trip from one hub Ln00 to one spoke L0m0 (e.g., between L100 and L050).
- o Each measurement loop passes two more links: one between the same hub Ln00 and another spoke L0m0 and from there to the alternate hub Ln00 (e.g., between L100 and L060 and then from L060 to L200)
- o Every monitored link is passed by a single round trip measurement loop only once and further only once unidirectional by two other loops. These unidirectional measurement loop sections forward packets in opposing direction along the monitored link. In the end, three measurement loops pass each single monitored link (sub-path). In figure 1, e.g., one measurement loop having a round trip L100 to L050 and back (M1, see below), a second loop passing L100 to L050 only (M3) and a third loop passing L050 to L100 only (M6).

Note that any 6 links connecting two to five nodes can be monitored that way too. Further note that the measurement loop overlay chosen is optimised for 6 links and a hub and spoke topology of two to five nodes. The 'one measurement loop per measured sub-path' paradigm only works under these conditions.

The above overlay scheme results in 6 measurement loops for the given example. The start and end of each measurement loop is PMS to L300 to L100 or L200 and a similar sub-path on the return leg. These parts of the measurement loops are omitted here for brevity (some discussion may be found below). The following delays are measured along the SR paths of each measurement loop:

1. M1 is the delay along L100 -> L050 -> L100 -> L060 -> L200
2. M2 is the delay along L100 -> L060 -> L100 -> L070 -> L200
3. M3 is the delay along L100 -> L070 -> L100 -> L050 -> L200
4. M4 is the delay along L200 -> L050 -> L200 -> L060 -> L100
5. M5 is the delay along L200 -> L060 -> L200 -> L070 -> L100
6. M6 is the delay along L200 -> L070 -> L200 -> L050 -> L100

An example for a stack of a loop consisting of Node-SID segments allowing to capture M1 is (top to bottom): 100 | 050 | 100 | 060 | 200 | PMS.

An example for a stack of Adj-SID segments the loop resulting in M1 is (top to bottom): 100 | 10050 | 50100 | 10060 | 60200 | PMS. As

can be seen, the Node-SIDs 100 and PMS are present at top and bottom of the segment stack. Their purpose is to transport the packet from the PMS to the start of the measurement loop at L100 and return it to the PMS from its end.

The Evaluation of the measurement loop Round Trip Delays M1 - M6 allow to detect the following state-changes of the monitored sub-paths:

- o If the loops are set up using Node-SIDs only, any single complete loss of connectivity caused by a failing single link between any Ln00 and any L0m0 node briefly disturbs (and changes the measured delay) of three loops. The traffic to the Node-SIDs is rerouted (in the case of a single link loss, no node is completely disconnected in the example network).
- o If the loops are set up using Adj-SIDs only, any single complete loss of connectivity caused by a failing single link between any Ln00 and any L0m0 node terminates the traffic along three measurement loops. The packets of all three loops will be dropped, until the link gets back into service. Traffic to Adj-SIDs is not rerouted. Note that Node-SIDs may be used to forward the measurement packets from the PMS to the hub node, where the first sub-path to be monitored begins and from the hub node, receiving the measurement from the last monitored sub path, to the PMS.
- o Any congested single interface between any Ln00 and any L0m0 node only impacts the measured delay of two measurement loops.
- o As an example, the formula for a single link (sub-path) Round Trip Delay (RTD) is shown here $4 * RTD_{L100-L050-L100} = 3 * M1 + M3 + M6 - M2 - M4 - M5$. This formula is reproducible for all other links: sum up $3*RTD$ measured along the loop passing the monitored link of interest in round trip fashion, and add the RTDs of the two measurement loops passing the link of interest only in a single direction. From this sum subtract the RTD measured on all loops not passing the monitored link of interest to get four times the RTD of the monitored link of interest.

A closer look reveals that each single event of interest for the proposed metric, which are a loss of connectivity or a case of congestion, uniquely only impacts a single set of measurement loops which can be determined a-priori. If, e.g., connectivity is lost between L200 and L050, measurement loops (3), (4) and (6) indicate a change in the measured delay.

As a second example: if the interface L070 to L100 is congested, measurement loops (3) and (5) indicate a change in the measured delay. Without listing all events, all cases of single losses of connectivity or single events of congestion influence only delay measurements of a unique set of measurement loops.

Assume that the measurement loops are set up while there's no congestion. In that case, the congestion free RTDs of all monitored links can be calculated as shown above. A single congestion event adds queuing delay to the RTD measured by two specific measurement loops. The two measurement loops impacted allow to distinct the congested interface and calculation of the queue-depth in terms of seconds. As an example, assume a queue of an average depth of 20 ms to build up at interface L200 to L070 after the uncongested measurement interval T0. The measurement loops M5 and M6 are the only ones passing the interface in that direction. Both indicate a congestion M5 and M6 of + 20 ms during measurement interval T1, while M1-4 indicate no change. The location of the congested interface is determined by the combination of the two (and only two) measurement loops M5 and M6 showing an increased delay. The average queue depth = (M5[T1] - M5[T0] + M6[T1] - M6[T0])/2.

As mentioned there's a constant delay added for each measurement loop, which is the delay of the path transversed from PMS -> L100 + L200 -> PMS. Please note, that this added delay is appearing twice in the formula resulting in the monitored link delay estimate of the example network. Then it is the RTD PMS -> L100 + RTD L200 -> PMS. Both RTDs can be directly measured by two additional measurements Cor1 = RTD (PMS -> L100 -> PMS) and Cor2 = RTD (PMS -> L200 -> PMS). The monitored link RTD formula was $linkRTD_{uncor} = 3 * M_x + M_y + M_z - M_s - M_t - M_u$. The correct $4 * linkRTD_x = 4 * linkRTD_{xuncor} - Cor1 - Cor2$.

If the interface between PMS and L100/L200 is congested, all measurements loops M1-M6 as well as Cor1 and Cor2 will see a change. A congested interface of a monitored link doesn't impact the RTDs captured by Cor1 and Cor2.

The measurement loops may also be set up between hub nodes L100 and L200, if that's preferred and supported by the nodes. In that case, the above formulas apply without correction.

3. Topology and measurement loop set up requirements

3.1. General network topology requirements

The metric and methods specified below can be applied to monitor networks or sub-paths forming a hub and spoke topology. A single sub-path status change of type loss of connectivity or congestion can

be detected. The nodes don't have to act as hubs or spokes, this terminology is only chosen to describe a topology requirement. In detail, the topology to be monitored MUST meet the following constraints:

- o The SR domain sub-paths to be monitored create a hub and spoke topology with a PMS connected to all hub nodes. The PMS may reside in a hub.
- o Exactly 6 (six) sub-paths are monitored.
- o The monitored sub-paths connect at least two and no more than 5 nodes.
- o Every spoke node MUST have at least one path to every hub node.
- o Every spoke node MUST at least be connected to one (or more) hub node(s) by two monitored sub-paths.
- o Sub-paths between spokes can't be monitored and therefore are out of scope (the overlay measurement loops can't be set up as desired).

Shared resources, like a Shared Risk Link Group (e.g., a single fiber bundle) or a shared queue passed by several logical links need to be considered during set up. Shared resources may either be desired or to be avoided. As an example, if a set of logical links share one parental scheduler queue, it is sufficient to monitor a single logical connection to monitor the state of that parental scheduler.

3.2. Sub-path Monitoring measurement loop routing requirements

The methodologies specified by this document REQUIRE a measurement loop path overlay of all path delay measurement streams F_i , i in $[1, 2...6]$ as defined in this section. In the following, a path delay measurement stream F_i is called measurement (loop) F_i for brevity.

- o Define the segment routed Sub-paths SP_i , i in $[1, 2...6]$ to be monitored. The Sub-paths SP_i SHOULD not share resources, if the operator isn't aware of the impact of the shared resources on the measurement loops F_i and the methodologies defined below. The Sub-path SP_i topology SHOULD respect the general network topology requirements as specified above.
- o Set up $i = 1, 2...6$ measurement loops F_i thus that measurement F_i passes SP_i and only SP_i bidirectional (or by a round-trip) from Hub to Spoke and back. Note that the correspondance of SP_i and F_i

isn't strictly required. Measurement F_i thus however appears in all methodologies calculating a metric related to SP_i .

- o Set up the SR path per measurement loops F_j and F_k thus that SP_i is passed by exactly one other measurement loop F_j unidirectional in direction Hub to Spoke and by exactly one other measurement loop F_k unidirectional in the opposite direction (Spoke to Hub). The measurement loop $F_i \neq F_j \neq F_k$. As a description, one measurement loop F_j pass SP_i in "downstream" direction from Hub to Spoke, whereas measurement loop F_k passes SP_i in "upstream" direction from Spoke to Hub.
- o Set up each segment routed measurement loop path F_i thus that it passes SP_i bidirectional as specified above, SP_j unidirectional from Hub to Spoke and SP_k unidirectional from Spoke to Hub. The monitored Sub-path SP_i MUST NOT be equal to SP_j and MUST NOT be equal to SP_k .
- o The measurement loop set up to monitor all Sub-paths SP_i is completed, if:
 - o
 - * Each Sub-path SP_i is passed by exactly three measurements loops F_i , F_j and F_k as specified above.
 - * Each segment routed measurement loop path F_i passes exactly three concatenated Sub-paths SP_i , SP_m and SP_n as specified above (indices m and n are chosen here only to avoid misconceptions which may result from picking indices j and k already appearing before - equality of j and k with either m and n is neither excluded nor required).

3.3. Path

This document specifies sub-path monitoring within a closed domain by a controlled and pre-designed measurement loop set-up. The path traversed by the packet SHOULD be reported, as detecting data plane forwarding in line with the desired measurement loop set-up is essential for the metric to enable and verify accurate evaluation. See [RFC8287] for SR MPLS OAM and [ID.draft-ietf-6man-spring-srv6-oam] for SRv6 OAM.

4. Generic Type-P-SR-Path-Periodic-* metric

To reduce the redundant information presented in the detailed metrics sections that follow, this section presents the specifications that are common to two or more metrics. The section is organized using

the same subsections as the individual metrics, to simplify comparisons.

4.1. Metric Name

All metrics use the Type-P convention as described in [RFC2330]. The rest of the name is unique to each metric.

4.2. Generic Metric Parameters

Refer to section 3.2. Metric Parameters: Type-P-* of [RFC6673]. The following parameters are added, enhanced or removed:

Dst SHOULD be a diagnostic IP address as specified by [RFC8287] and [RFC8029], if MPLS OAM is operated to capture the metric.

Fi, where i in [1, 2...6], a selection function defining unambiguously a packet of one particular stream i forming part of the monitoring overlay measurement loop set up.

L, a packet length in bits. The packets of all Type-P-SR-Path-Delay-Periodic-Streams Fi SHOULD all be of the same length.

MLAi, a stack of Segment IDs determining a monitoring loop Fi. The Segment-IDs MUST be chosen so that a singleton type-p packet of selection function Fi passes the sub-path i to be monitored.

No support: lambda (Poisson Streams remain ffs.)

4.3. Metric Units

Refer to section 3.4. Metric Units: Type-P-* of [RFC6673].

5. Singleton Definition for Type-P-SR-Path-Periodic-Delay

5.1. Metric Name

Type-P-SR-Path-Periodic-Delay

5.2. Metric Parameters

See section Section 4.2.

5.3. Delay Metric Units

A sequence of consecutive time values. The value of a Type-P-SR-Path-Periodic-Delay is either a real number or an undefined

(informally, infinite) number of seconds per singleton of each stream F_i .

5.4. Definition

Section 3.4 of [RFC7679] applies per singleton of each stream F_i . The additional information related to singletons of section 4.2.4 of [RFC3432] applies too.

5.5. Discussion

See section 3.5 of [RFC7679]. One generalisation seems appropriate: a global satellite navigation system affords one way to achieve synchronization within usec.

5.6. Methodologies

Section 3.6 of [RFC7679] applies per stream F_i with one exception: at the Src host, select Src and Dst IP addresses, if IP-routing is applied, or select the proper functional IP-destination address if an [RFC8287] SR MPLS OAM packet format is applied. Further add the appropriate stack of Segment IDs MLA_i determining the monitoring loop F_i and form a test packet of Type-P with these addresses and the segment stack.

5.7. Errors and Uncertainties

See section 3.7 of [RFC7679] and section 4.6 of [RFC3432].

5.8. Reporting the metric

See section 3.8 of [RFC7679].

6. Definition of Samples for Type-P-SR-Path-Periodic-Delay

This sections defines metric samples and metrics derived from samples.

6.1. Generic Type-P-SR-Path-Periodic-Delay-* metric

To reduce the redundant information presented in the detailed metrics sections that follow, this section presents the specifications that are common to two or more metrics. The section is organized using the same subsections as the individual metrics, to simplify comparisons.

6.1.1. Metric Name

Type-P-SR-Path-Periodic-Delay-*

6.1.2. Metric Parameters

Src, the IP address of a host

Dst, the IP address of a host

MLAi, a stack of Segment IDs

T0, a time

Tf, a time

incT, a time

6.1.3. Metric Units

See section Section 5.3.

6.1.4. Metric Defintion

Given T0 and Tf and nominal inter-packet interval incT, those time values greater than or equal to T0 and less than or equal to Tf are then selected. At each of the selected times in this process, we obtain one value of Type-P-SR-Path-Periodic-Delay. The value of the sample is the sequence made up of the resulting [time, delay] pairs. If there are no such pairs, the sequence is of length zero and the sample is said to be empty.

6.1.5. Discussion

See section 4.4 of [RFC3432].

6.1.6. Errors and uncertainties

See section 4.6 of [RFC3432].

6.2. Definition of Type-P-SR-Path-Periodic-Delay-Stream

The only definition required for this metric is a unique metric name.

6.2.1. Metric Name

Type-P-SR-Path-Periodic-Delay-Stream

6.3. Definition of Type-P-SR-Path-Periodic-Delay-Variation

The smallest sample Type-P-SR-Path-Periodic-Delay-Stream is one of two consecutively received values. These may be used to calculate a Segment Routed Path Delay-Variation (SRDV) singleton, defined below.

6.3.1. Metric Name

Type-P-SR-Path-Periodic-Delay-Variation

6.3.2. Methodologies

SRDV[i, j], for each sample of packets j and j-1 of stream Fi, j > 1, the delay variation between successive packets is calculated as:

$$\text{SRDV}[i, j] = \text{Delay}[i, j] - \text{Delay}[i, j-1],$$

j in [2,3...N] and N the total number of packets received at Dst. If one or more of the M packets sent by Src are lost, they are ignored for the metric, as no reasonable metric value is defined here. If N > 1, the metric is calculated for every valid packet received and the preceding one.

6.3.3. Discussion of SRDV

Evaluation statistics of differential SRDV metric samples may help to identify issues.

6.3.4. Errors and uncertainties

See section 2.7 of [RFC3393].

6.4. Definition of Type-P-SR-Path-Periodic-Delay-Variation-Stream

The only definition required for this metric is a unique metric name.

6.4.1. Metric Name

Type-P-SR-Path-Periodic-Delay-Variation-Stream

6.4.2. Metric Defintion

Given T_0 and T_f , those time values greater than or equal to T_0 and less than or equal to T_f are then selected. At each of the selected times in this process, we obtain one value of Type-P-SR-Path-Periodic-Delay. The value of the sample is the sequence made up of the resulting [time, delay-variation] pairs with time being set to the Dst timestamp of the Delay-Variation singleton, for which a valid singleton is calculated. If there are no such pairs, the sequence is of length zero and the sample is said to be empty. If N Delay singletons are captured and sampled $N-1$ Delay-Variation singletons are sampled during the same interval

7. Statistic Definitions for SR-Path-Periodic-*--Stream samples

Change point detection requires statistical defintions. These are provided below. The names of the statistics contain an "*" placeholder, which may be replaced by "Delay" or "Delay-Variation" [Editor note: a "Loss" metric remains tbd].

7.1. SR-Path-Periodic-*--Mean

For a type-p metric, the mean is specified by:

$$SR-*Mean = (1/N) * \text{Sum}(\text{from } i=1 \text{ to } N, \text{ value}[i])$$

- o N sample size
- o value sample value of a sampled [time, value] pair

7.2. SR-Path-Periodic-*--Std

For a type-p metric, the Standard-Deviation Std is specified by:

$$SR-*Std = [1/(N-1)] * \text{Sum}(\text{from } i=1 \text{ to } N, [SR-*Mean - \text{value}[i]]^2)$$

- o N sample size
- o value sample value of a sampled [time, value] pair
- o $SR-*Mean$ sample mean of the same metric as defined above

The definition as given above requires a two-pass calculation per sample. Algorithms estimating the standard-deviation by one-pass calculation have been published and might be preferable, if metric singletons and samples aren't buffered or calculations need to be fast.

8. Sub-Path monitoring metrics derived from samples captured along the measurement loops

To produce meaningful sub-path monitoring values, the measurement loop metrics are captured during a phase with stable networking conditions. In a backbone network domain, the absence of congestion often is a sufficient condition (frequent traffic shifts due to changes in routing and traffic engineering aren't expected). This may be different in a network based on a shared medium. It may be outright difficult in networks with frequently changing traffic management- and routing-policies.

In the following, the index CS indicates a statistic captured during a measurement interval with stable routing and no congestion.

8.1. Baseline measurement

Capture a sample of delay values Type-P-SR-Path-Periodic-Delay-Stream of sample size N for each measurement loop F_i . As a rule of thumb choose N in [30, 100].

For each measurement loop F_i , calculate the following metrics characterising the monitored Sub-Paths during stable and congestion free network conditions:

- o SR-Path-Delay-MeanCS $_i$, the mean delay captured along measurement loop F_i
- o SR-Path-Delay-StdCS $_i$, the standard-deviation of the delay captured along measurement loop F_i
- o SR-Path-Delay-Variation-MeanCS $_i$, the mean delay variation captured along measurement loop F_i
- o SR-Path-Delay-Variation-StdCS $_i$, the standard-deviation of the delay variation captured along measurement loop F_i

A stable and uncongested network should produce rather constant delays, resulting in low standard-deviation values and almost zero mean delay variation.

Example data was captured in a lightly loaded Gigabit network. 11 routers are passed per measurement loop. The sample size is 30 packets, more than 200 samples were captured per measurement loop. The loops are set up for a different purpose than specified here, they are picked due to a high number of passed routers. Note that SR-DV-Mean here refers to an abs(SR-DV-Mean) sample, thus small, positive, non-zero means result. The time unit is microseconds.

Metric	Quantile	SR-D-Mean	SR-D-Std	SR-DV-Mean	SR-DV-Std
Loop1	95%	34510	40	28	49
Loop2	95%	35104	45	34	49
Loop1	50%	34495	17	15	13
Loop2	50%	35088	15	14	12
Loop1	5%	34504	10	12	11
Loop2	5%	35080	13	12	9

Example baseline metrics for an 11 hop measurement loop

Figure 2

8.2. Discussion of the baseline measurement

Delay outliers may occur at any time in any communication network, and the measurement system packet processing itself may also produce some. It is fair to expect only single outliers in a stable, not congested network. It may be worth to capture several consecutive SR-Path-Periodic*-Stream samples and compare their statistics, before picking reasonable baseline metric values. Samples showing higher standard deviations (compare the 95% quantile values in the above figure to the 50% quantile values) may benefit from removing the maximum singleton value from the sample. This will smooth the mean and standard-deviation, and if the result then is closer to those of the majority of the samples, foster confidence in determining the baseline metrics. Depending on the preferred method of data-processing and storing, this may require capturing the sample maximum as a separate metric.

8.3. Definition of SR-Path-Sub-Path-RTD-Estimate

Within a single evaluation interval of identical Time T_0 and T_f , SR-Path-Delay-MeanCS_i (from now on DMeanCS_i) is the mean delay of the measurement loop passing the monitored Sub-Path SP_i by a round trip. Let's keep the index i applied above, then F_j and F_k with captured mean delays DMeanCS_j and DMeanCS_k pass SP_i unidirectional. Further, 3 measurement loops F_x , F_y and F_z don't pass Sub-Path at all. The corresponding mean delays are DMeanCS_s, DMeanCS_t and DMeanCS_u.

The the SR-Path-Sub-Path-RTD-Estimate of the Round Trip Delay along the monitored Sub-Path F_i , RTD_{F_i} , is

$$RTD_{F_i} = (3 * DMeanCS_i + DMeanCS_j + DMeanCS_k - DMeanCS_x - DMeanCS_y - DMeanCS_z) / 4$$

8.4. Definition of SR-Path-Sub-Path-*--Changepoint

The asterisk stands for "Interface" as well as "Connectivity" (the latter may be indicated by packet drops, but also a change in sub-path routes with a change in measurement loop delay may be applied to detect and locate this event).

Network changes are often characterised by a change in the mean delay of a monitoring measurement. CUSUM (cumulative sum) charts have been shown to be efficient in detecting shifts in the mean of a process [NIST]. The upper bound CUSUM is defined as:

$$Sup(t)_{-F_i-Delay} = \max(0, Sup(t-1) + x_t - SR-Path-*--MeanCS_i - k_i)$$

with $Sh_0 = 0$, $k_i = \Delta * SR-Path-*--StdCS_i$ (Δ is a dimensionless integer number), $x_t = Type-P-SR-Path-Periodic-*$ singleton for measurement loop F_i at time t .

The actual SR-Path-Delay-Mean of Measurement Loop F_i is decided to be significantly above $SR-Path-*--MeanCS_i$, if:

$$Sup(t)_{-F_i-Delay} > h_{SP}, \text{ with } h_{SP} = d * k_i \text{ (d is a dimensionless integer number).}$$

An analogous CUSUM controls changes to a lower mean delay (which may be caused by a re-routing event):

$$Slo(t)_{-F_i-Delay} = \max(0, Slo(t-1) + SR-Path-*--MeanCS_i - x_j - k)$$

The actual SR-Path-Delay-Mean of F_i is decided to be significantly below $SR-Path-*--MeanCS_i$, if:

$$Slo(t)_{-F_i-Delay} > h_{SP}$$

8.5. Discussion of SR-Path-Sub-Path-*--Changepoint

CUSUM chart based changepoint detection is sensible even to small changes in the mean. CUSUM charts offer a limited protection against single, isolated outliers. A cumulated sum only grows, if the controlled process consistently changes its mean (or standard deviation, respectively). Assuming constant physical minimum delays to characterise wireline communication networks, a change in standard

deviation not affecting the mean delay doesn't seem to be caused by a change in networking conditions.

The measured delays will change once a Sub-Path route has changed, or once persistent congestion starts to fill a queue. Both indicate changes in the network. As the Sub-Pathes SP_i form an overlay with designed properties, every network change affecting a sub-path creates correlated SR-Path-* metric changes. As the correspondance of network changes to Sub-Path metrics is known a-priori, detecting correlated SR-Path-* metric changes allows to locate the change.

In the absence of packet re-routing, packet loss is characterising a loss of connectivity. Packet loss requires a time threshold when to decide that an active measurement packet was lost, and consecutive loss requires receiver awareness, that packets have been sent (this argues for the sender to be the receiver, unless both communicate fast and reliable out of band).

The preferred CUSUM parametrisation will depend on the kind of events to detected and on the outlier characteristics.

$k_i = \Delta * SR\text{-}Path\text{-}*\text{-}StdCS_i$ may be set to a value relevant high enough to exclude single outliers to trigger an alert, but low enough to indicate persistent changes in delay. The same holds for the to be picked for d .

A broader discussion on CUSUM parametrisation may be found in literature. Networking skills are required to parametrise CUSUM, as well as to interpret the results (notably to differ re-routing from congestion).

8.6. Definition of SR-Path-Sub-Path-Congestion-Location

An interface along a single monitored Sub-Path SP_i whose queue is persistently filled adds latency to measurement loop F_i and one of the two unidirectional measurement loops F_j and F_k passing Sub-Path SP_i . F_j has been defined to pass SP_i from Hub to Spoke and F_k pass SP_i in opposite direction.

```
IF Sup(t)_SPi_Periodic-Delay + Sup(t)_SPj_Periodic-Delay > h_SP
AND h_SP > Sup(t)_SPk_Periodic-Delay
AND h_SP > Sup(t)_SPx_Periodic-Delay
AND h_SP > Sup(t)_SPy_Periodic-Delay
AND h_SP > Sup(t)_SPz_Periodic-Delay
```

Then Sub-Path SPi faces congestion in direction "Hub to Spoke".

IF $\text{Sup}(t)_{\text{SPi_Periodic-Delay}} + \text{Sup}(t)_{\text{SPk_Periodic-Delay}} > h_{\text{SP}}$

AND $h_{\text{SP}} > \text{Sup}(t)_{\text{SPj_Periodic-Delay}}$

AND $h_{\text{SP}} > \text{Sup}(t)_{\text{SPx_Periodic-Delay}}$

AND $h_{\text{SP}} > \text{Sup}(t)_{\text{SPy_Periodic-Delay}}$

AND $h_{\text{SP}} > \text{Sup}(t)_{\text{SPz_Periodic-Delay}}$

Then Sub-Path SPi faces congestion in direction "Spoke to Hub".

Here, h_{SP} is a universal threshold in unit time to indicate a filling queue or a significant change in delay due to a Sub-Path reroute or another persistent change in topology (like e.g. automated Layer 1 / Layer 2 topology changes). SPx, SPy and SPz don't pass

8.7. Discussion of SR-Path-Sub-Path-* -Location

[Editor Note: Discussion and a suitable connectivity monitoring metric Definition+Discussion to be added.]

9. Discussion of Temporal Resolution

[Editor Note: requires a review..] The metric reports loss of connectivity of monitored sub-path or congestion of an interface and identifies the sub-path and the direction of traffic in the case of congestion.

The temporal resolution of the detected events depends on the spacing interval of packets transmitted per measurement path. An identical sending interval is chosen for every measurement path. As a rule of thumb, an event is reliably detected if a sample consists of at least 5 probes indicating the same underlying change in behavior. Depending on the underlying event either two or three measurement paths are impacted. At least two consecutively received measurement packets per measurement path should suffice to indicate a change. The values chosen for an operational network will have to reflect scalability constraints of a PMS measurement interface. As an example, a PMS may work reliable if no more than one measurement packet is transmitted per millisecond. Further, measurement is configured so that the measurement packets return to the sender interface. Assume always groups of 6 links to be monitored as described above by 6 measurements paths. If one packet is sent per measurement path within 500 ms, up to 498 links can be monitored with

a reliable temporal resolution of roughly one second per detected event.

Note that per group measurement packet spacing, measurement loop delay difference and latency caused by congestion impact the reporting interval. If each measurement path of a single 6 link monitoring group is addressed in consecutive milliseconds (within the 500 ms interval) and the sum of maximum physical delay of the per group measurement paths and latency possibly added by congestion is below 490 ms, the one second reports reliably capture 4 packets of two different measurement paths, if two measurement paths are congested, or 6 packets of three different measurement paths, if a link is lost.

A variety of reporting options exist, if scalability issues and network properties are respected.

10. IANA Considerations

If standardised, the metric will require an entry in the IPPM metric registry.

11. Security Considerations

This draft specifies how to use methods specified or described within [RFC8402] and [RFC8403]. It does not introduce new or additional SR features. The security considerations of both references apply here too.

12. References

12.1. 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>>.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <<https://www.rfc-editor.org/info/rfc2678>>.
- [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>>.
- [RFC6673] Morton, A., "Round-Trip Packet Loss Metrics", RFC 6673, DOI 10.17487/RFC6673, August 2012, <<https://www.rfc-editor.org/info/rfc6673>>.
- [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>>.
- [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>>.
- [RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <<https://www.rfc-editor.org/info/rfc8287>>.
- [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>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

12.2. Informative References

[CommodityTomography]

Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, ED., and N. Taft, "Structural analysis of network traffic flows", 2004, <https://www.cc.gatech.edu/classes/AY2007/cs7260_spring/papers/odflows-sigm04.pdf>.

[ID.draft-ietf-6man-spring-srv6-oam]

Zafar, A., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", 2021.

[NIST]

NIST, "NIST/SEMATECH e-Handbook of Statistical Methods, section CUSUM Control Charts", 2021, <<http://www.itl.nist.gov/div898/handbook/>>.

[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>>.

[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>>.

Author's Address

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

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

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

H. Song
Futurewei
B. Gafni
Nvidia
T. Zhou
Z. Li
Huawei
F. Brockners
Cisco
S. Bhandari, Ed.
Thoughtspot
R. Sivakolundu
Cisco
T. Mizrahi, Ed.
Huawei
February 17, 2021

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

Abstract

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

Status of This Memo

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

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

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

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

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. 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	6
6. Security Considerations	7
7. References	7
7.1. Normative References	8
7.2. Informative References	8
Appendix A. Hop Limit and Hop Count in Direct Exporting	8
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 receiving entity

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

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

2. Conventions

2.1. Requirement Language

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

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

DEX: Direct EXporting

3. The Direct Exporting (DEX) IOAM Option Type

3.1. Overview

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

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

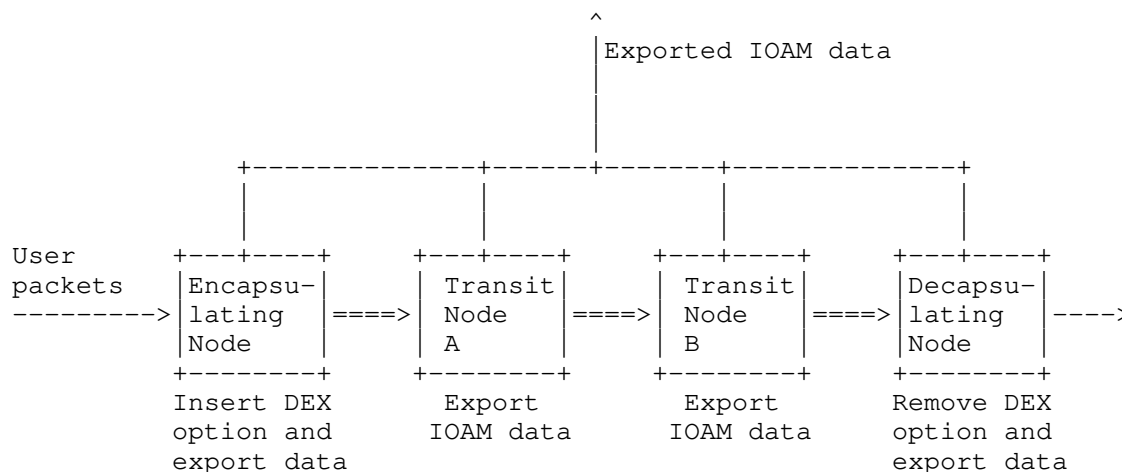


Figure 1: DEX Architecture

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

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

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

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

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

3.2. The DEX Option Format

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

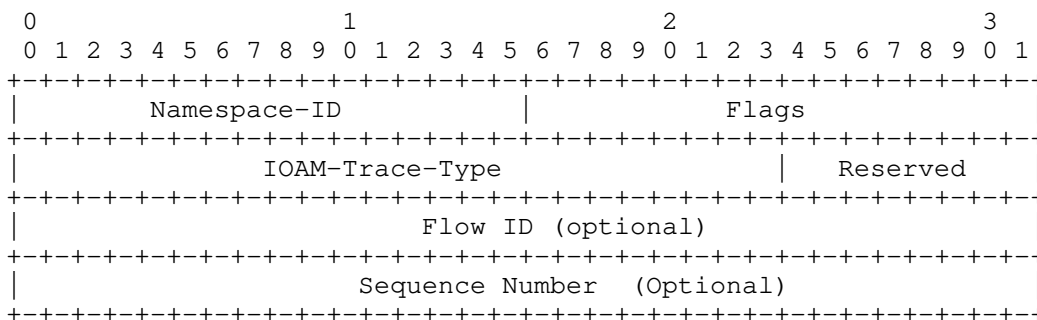


Figure 2: DEX Option Format

- Namespace-ID A 16-bit identifier of the IOAM namespace, as defined in [I-D.ietf-ippm-ioam-data].
- Flags A 16-bit field, comprised of 16 one-bit subfields. Flags are allocated by IANA, as defined in Section 4.2.
- IOAM-Trace-Type A 24-bit identifier which specifies which data fields should be exported. The format of this field is as defined in [I-D.ietf-ippm-ioam-data]. Specifically, bit 23, which corresponds to the Checksum Complement data field, should be assigned to be zero by the IOAM encapsulating node, and ignored by transit and decapsulating nodes. The reason for this is that the Checksum Complement is intended for in-flight packet modifications and is not relevant for direct exporting.

Reserved This field SHOULD be ignored by the receiver.

Flow ID A 32-bit flow identifier. If the actual Flow ID is shorter than 32 bits, it is zero padded in its most significant bits. The field is set at the encapsulating node. The Flow ID can be uniformly assigned by a central controller or algorithmically generated by the encapsulating node. The latter approach cannot guarantee the uniqueness of Flow ID, yet the conflict probability is small due to the large Flow ID space. The Flow ID can be used to correlate the exported data of the same flow from multiple nodes and from multiple packets.

Sequence Number A 32-bit sequence number starting from 0 and increasing by 1 for each following monitored packet from the same flow at the encapsulating node. The Sequence Number, when combined with the Flow ID, provides a convenient approach to correlate the exported data from the same user packet.

4. IANA Considerations

4.1. IOAM Type

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

TBD-type IOAM Direct Export (DEX) Option Type

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

4.2. IOAM DEX Flags

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

5. Performance Considerations

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

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

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

6. Security Considerations

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

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

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

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

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

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

7. References

7.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

7.2. Informative References

- [I-D.ietf-ippm-ioam-flags]
Mizrahi, T., Brockners, F., Bhandari, S., Sivakolundu, R., Pignataro, C., Kfir, A., Gafni, B., Spiegel, M., and J. Lemon, "In-situ OAM Flags", draft-ietf-ippm-ioam-flags-03 (work in progress), October 2020.
- [I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.
- [I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-04 (work in progress), November 2020.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

Appendix A. Hop Limit and Hop Count in Direct Exporting

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

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

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

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

Authors' Addresses

Haoyu Song
Futurewei
2330 Central Expressway
Santa Clara 95050
USA

Email: haoyu.song@huawei.com

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

Email: gbarak@nvidia.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Zhenbin Li
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

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

Email: fbrockne@cisco.com

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

Email: shwetha.bhandari@thoughtspot.com

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

Email: sramesh@cisco.com

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

Email: tal.mizrahi.phd@gmail.com

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

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

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

Abstract

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

Status of This Memo

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

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

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

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

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
5. Active Measurement with IOAM	5
6. IANA Considerations	7
7. Performance Considerations	7
8. Security Considerations	7
9. References	9
9.1. Normative References	9
9.2. Informative References	9
Authors' Addresses	10

1. Introduction

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

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

2. Conventions

2.1. Requirements Language

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

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

3. New IOAM Trace Option Flags

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

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

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

4. Loopback in IOAM

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

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

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

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

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

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

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

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

Note that an IOAM encapsulating node may either be an endpoint (such as an IPv6 host), or a switch/router that pushes a tunnel encapsulation onto data packets. In both cases, the functionality that was described above avoids IOAM data leaks from the IOAM domain. 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 the measurement. This section presents use cases of active measurement using the IOAM Active flag.

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

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

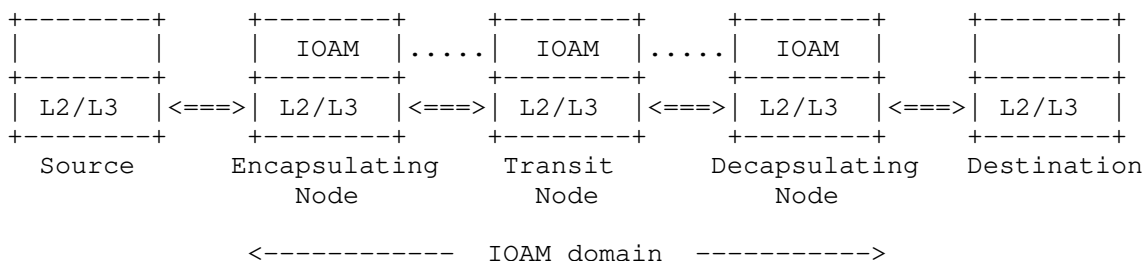


Figure 1: Network using IOAM.

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

- 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.

6. IANA Considerations

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

Bit 1 "Loopback" (L-bit)

Bit 2 "Active" (A-bit)

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

7. Performance Considerations

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

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

8. Security Considerations

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

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

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

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

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

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

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

9. References

9.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-05 (work in progress), December 2020.
- [I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-04 (work in progress), November 2020.

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

Authors' Addresses

Tal Mizrahi
Huawei
Israel

Email: tal.mizrahi.phd@gmail.com

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

Email: fbrockne@cisco.com

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

Email: shwetha.bhandari@thoughtspot.com

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

Email: sramesh@cisco.com

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

Email: cpignata@cisco.com

Aviv Kfir
Nvidia

Email: avivk@nvidia.com

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

Email: gbarak@nvidia.com

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

Email: mspiegel@barefootnetworks.com

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

Email: jennifer.lemon@broadcom.com

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

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

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

Abstract

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

Status of This Memo

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

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

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 25, 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	3
2. Conventions	3
2.1. Requirements Language	3
2.2. Abbreviations	3
3. In-situ OAM Metadata Transport in IPv6	3
4. IOAM Deployment In IPv6 Networks	6
4.1. Considerations for IOAM deployment in IPv6 networks . . .	6
4.2. IOAM domains bounded by hosts	7
4.3. IOAM domains bounded by network devices	7
4.4. Deployment options	8
4.4.1. IPv6-in-IPv6 encapsulation	8
4.4.2. IP-in-IPv6 encapsulation with ULA	8
4.4.3. x-in-IPv6 Encapsulation that is used Independently .	9
5. Security Considerations	9
6. IANA Considerations	10
7. Acknowledgements	10
8. References	10
8.1. Normative References	10
8.2. Informative References	11
Authors' Addresses	11

1. Introduction

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

2. Conventions

2.1. Requirements Language

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

2.2. Abbreviations

Abbreviations used in this document:

E2E: Edge-to-Edge

IOAM: In-situ Operations, Administration, and Maintenance

ION: IOAM Overlay Network

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

3. In-situ OAM Metadata Transport in IPv6

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

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

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. xxxxxx=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.

4. IOAM Deployment In IPv6 Networks

4.1. Considerations for IOAM deployment in IPv6 networks

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

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

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

- C3 Packets with IOAM data or associated ICMP errors, should not arrive at destinations that have no knowledge of IOAM. For 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.

4.2. IOAM domains bounded by hosts

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

4.3. IOAM domains bounded by network devices

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

4.4. Deployment options

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

4.4.1. IPv6-in-IPv6 encapsulation

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

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

1. The source address of the outer IPv6 header is that of the IOAM encapsulating node. The destination address of the outer IPv6 header is the same as the inner IPv6 destination address, i.e., the destination address of the packet does not change.
2. To simplify debugging in case of leaked IOAM data fields, consider a new IOAM E2E destination option to identify the Source IOAM domain (AS, v6 prefix). Insert this option into the IOAM destination options EH attached to the outer IPv6 header. This additional information would allow for easy identification of an AS operator that is the source of packets with leaked IOAM information. Note that leaked packets with IOAM data fields would only occur in case a router would be misconfigured.
3. All the IOAM options are defined with type "00" - skip over this option and continue processing the header. Presence of these options must not cause packet drops in network elements that do not understand the option. In addition, [I-D.ietf-6man-hbh-header-handling] should be considered.

4.4.2. IP-in-IPv6 encapsulation with ULA

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

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

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

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

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

5. Security Considerations

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

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

6. IANA Considerations

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

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

Hex Value	Binary Value			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]

7. Acknowledgements

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

8. References

8.1. Normative References

- [I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-direct-export] Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

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

8.2. Informative References

- [I-D.ietf-6man-hbh-header-handling] Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options Extension Header", March 2016.
- [I-D.kitamura-ipv6-record-route] Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.
- [RFC1772] Rekhter, Y. and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, DOI 10.17487/RFC1772, March 1995, <<https://www.rfc-editor.org/info/rfc1772>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <<https://www.rfc-editor.org/info/rfc8250>>.

Authors' Addresses

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

Email: shwetha.bhandari@thoughtspot.com

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

Email: fbrockne@cisco.com

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

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

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

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

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

Email: avivk@mellanox.com

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

Email: gbarak@mellanox.com

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

Email: petr@fb.com

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

Email: mickey.spiegel@intel.com

Suresh Krishnan
Kaloom

Email: suresh@kaloom.com

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

Email: rajiva@cisco.com

Mark Smith
PO BOX 521
HEIDELBERG, VIC 3084
AU

Email: markzzzsmith+id@gmail.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: July 16, 2021

T. Zhou, Ed.
Huawei
J. Guichard
Futurewei
F. Brockners
S. Raghavan
Cisco Systems
January 12, 2021

A YANG Data Model for In-Situ OAM
draft-ietf-ippm-ioam-yang-00

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in user packets while the packets traverse a path between two points in the network. This document defines a YANG module for the IOAM function.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 16, 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. Conventions used in this document	2
2.1. Tree Diagrams	3
3. Design of the IOAM YANG Data Model	3
3.1. Profiles	3
3.2. Preallocated Tracing Profile	5
3.3. Incremental Tracing Profile	5
3.4. Direct Export Profile	6
3.5. Proof of Transit Profile	6
3.6. Edge to Edge Profile	7
4. IOAM YANG Module	7
5. Security Considerations	21
6. IANA Considerations	22
7. Acknowledgements	23
8. References	23
8.1. Normative References	23
8.2. Informative References	24
Appendix A. Examples	24
Authors' Addresses	25

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) [I-D.ietf-ippm-ioam-data] records OAM information within user packets while the packets traverse a network. The data types and data formats for IOAM data records have been defined in [I-D.ietf-ippm-ioam-data]. The IOAM data can be embedded in many protocol encapsulations such as Network Services Header (NSH) and IPv6.

This document defines a data model for IOAM capabilities using the YANG data modeling language [RFC7950]. This YANG model supports all the five IOAM options, which are Incremental Tracing Option, Pre-allocated Tracing Option, Direct Export Option [I-D.ietf-ippm-ioam-direct-export], Proof of Transit (PoT) Option, and Edge-to-Edge Option.

2. Conventions used in this document

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

BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are used in this specification:

- o augment
- o data model
- o data node

The terminology for describing YANG data models is found in [RFC7950].

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Design of the IOAM YANG Data Model

3.1. Profiles

The IOAM model is organized as list of profiles as shown in the following figure. Each profile associates with one flow and the corresponding IOAM information.

The "ioam-info" is a container for all the read only assistant information such as units or timestamp format. So that monitoring systems can interpret the IOAM data.


```

module: ietf-ioam
  +--rw ioam
    +--ro ioam-info
    +--rw ioam-profiles
      +--rw admin-config
        | +--rw enabled? boolean
      +--rw ioam-profile* [profile-name]
        +--rw profile-name string
        +--rw filter
          | +--rw filter-type? ioam-filter-type
          | +--rw ace-name? -> /acl:acls/acl/aces/ace/name
        +--rw protocol-type? ioam-protocol-type
        +--rw incremental-tracing-profile {incremental-trace}?
          | ...
        +--rw preallocated-tracing-profile {preallocated-trace}?
          | ...
        +--rw direct-export-profile {direct-export}?
          | ...
        +--rw pot-profile {proof-of-transit}?
          | ...
        +--rw e2e-profile {edge-to-edge}?
          | ...
          ...

```

The "enabled" is an administrative configuration. When it is set to true, IOAM configuration is enabled for the system. Meanwhile, the IOAM data-plane functionality is enabled.

The "filter" is used to identify a flow, where the IOAM profile can apply. There may be multiple filter types. ACL [RFC8519] is a common way to specify a flow. Each IOAM profile can associate with an ACE (Access Control Entry). IOAM actions MUST be driven by the accepted packets, when the matched ACE "forwarding" action is "accept".

The IOAM data can be encapsulated into multiple protocols, e.g., IPv6 [I-D.ietf-ippm-ioam-ipv6-options] and NSH [I-D.ietf-sfc-ioam-nsh]. The "protocol-type" is used to indicate where the IOAM is applied. For example, if the "protocol-type" is IPv6, the IOAM ingress node will encapsulate the associated flow with the IPv6-IOAM [I-D.ietf-ippm-ioam-ipv6-options] format.

IOAM data includes five encapsulation types, i.e., incremental tracing data, preallocated tracing data, direct export data, prove of transit data and end to end data. In practice, multiple IOAM data types can be encapsulated into the same IOAM header. The "ioam-profile" contains a set of sub-profiles, each of which relates to one encapsulation type. The configured object may not support all the sub-profiles. The supported sub-profiles are indicated by 5 defined

features, i.e., "incremental-trace", "preallocated-trace", "direct export", "proof-of-transit", "edge-to-edge".

3.2. Preallocated Tracing Profile

The IOAM tracing data is expected to be collected at every node that a packet traverses to ensure visibility into the entire path a packet takes within an IOAM domain. The preallocated tracing option will create pre-allocated space for each node to populate its information. The "preallocated-tracing-profile" contains the detailed information for the preallocated tracing data. The information includes:

- o enabled: indicates whether the preallocated tracing profile is enabled.
- o node-action: indicates the operation (e.g., encapsulate IOAM header, transit the IOAM data, or decapsulate IOAM header) applied to the dedicated flow.
- o use-namespace: indicate the namespace used for the trace types.
- o trace-type: indicates the per-hop data to be captured by the IOAM enabled nodes and included in the node data list.
- o Loopback mode is used to send a copy of a packet back towards the source.
- o Active mode indicates that a packet is used for active measurement.

```
+--rw preallocated-tracing-profile {preallocated-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
    | +--rw use-namespace?      ioam-namespace
    | +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?   boolean
  +--rw enable-active-mode?     boolean
```

3.3. Incremental Tracing Profile

The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header. The "incremental-tracing-profile" contains the detailed information for the incremental tracing data. The detailed information is the same as the Preallocated Tracing

Profile, but with one more variable, "max-length", which restricts the length of the IOAM header.

```

+--rw incremental-tracing-profile {incremental-trace}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
  |   +--rw use-namespace?      ioam-namespace
  |   +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?   boolean
  +--rw enable-active-mode?     boolean
  +--rw max-length?             uint32

```

3.4. Direct Export Profile

The direct export option is used as a trigger for IOAM nodes to export IOAM data to a receiving entity (or entities). The "direct-export-profile" contains the detailed information for the direct export data. The detailed information is the same as the Preallocated Tracing Profile, but with one more optional variable, "flow-id", which is used to correlate the exported data of the same flow from multiple nodes and from multiple packets.

```

+--rw direct-export-profile {direct-export}?
  +--rw enabled?                boolean
  +--rw node-action?            ioam-node-action
  +--rw trace-types
  |   +--rw use-namespace?      ioam-namespace
  |   +--rw trace-type*         ioam-trace-type
  +--rw enable-loopback-mode?   boolean
  +--rw enable-active-mode?     boolean
  +--rw flow-id?                uint32

```

3.5. Proof of Transit Profile

The IOAM Proof of Transit data is to support the path or service function chain verification use cases. The "pot-profile" contains the detailed information for the prove of transit data. The detailed information are described in [I-D.ietf-sfc-proof-of-transit].

```

+--rw pot-profile {proof-of-transit}?
  +--rw enabled?                boolean
  +--rw active-profile-index?   pot:profile-index-range
  +--rw pot-profile-list* [pot-profile-index]
    +--rw pot-profile-index     profile-index-range
    +--rw prime-number          uint64
    +--rw secret-share          uint64
    +--rw public-polynomial     uint64
    +--rw lpc                   uint64
    +--rw validator?           boolean
    +--rw validator-key?       uint64
    +--rw bitmask?            uint64
      +--rw opot-masks
      +--rw downstream-mask*   uint64
      +--rw upstream-mask*    uint64

```

3.6. Edge to Edge Profile

The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node. The "e2e-profile" contains the detailed information for the edge to edge data. The detailed information includes:

- o enabled: indicates whether the edge to edge profile is enabled.
- o node-action is the same semantic as in Section 2.2.
- o use-namespace: indicate the namespace used for the edge to edge types.
- o e2e-type indicates data to be carried from the ingress IOAM node to the egress IOAM node.

```

+--rw e2e-profile {edge-to-edge}?
  +--rw enabled?                boolean
  +--rw node-action?           ioam-node-action
  +--rw e2e-types
    +--rw use-namespace?       ioam-namespace
    +--rw e2e-type*           ioam-e2e-type

```

4. IOAM YANG Module

```

<CODE BEGINS> file "ietf-ioam@2021-01-12.yang"
module ietf-ioam {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-ioam";
  prefix "ioam";

```

```
import ietf-pot-profile {
  prefix "pot";
  reference "draft-ietf-sfc-proof-of-transit";
}

import ietf-access-control-list {
  prefix "acl";
  reference
    "RFC 8519: YANG Data Model for Network Access Control
     Lists (ACLs)";
}

organization
  "IETF IPPM (IP Performance Metrics) Working Group";

contact
  "WG Web: <http://tools.ietf.org/wg/ippm>
  WG List: <ippm@ietf.org>
  Editor: zhoutianran@huawei.com
  Editor: james.n.guichard@futurewei.com
  Editor: fbrockne@cisco.com
  Editor: srihari@cisco.com";

description
  "This YANG module specifies a vendor-independent data
  model for the In Situ OAM (IOAM).

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

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see the
  RFC itself for full legal notices."

revision 2021-01-12 {
  description "Initial revision.";
  reference "draft-ietf-ippm-ioam-yang";
}

/*
 * FEATURES
 */
```

```
feature incremental-trace
{
  description
    "This feature indicated that the incremental tracing option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature preallocated-trace
{
  description
    "This feature indicated that the preallocated tracing option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature direct-export
{
  description
    "This feature indicated that the direct export option is
    supported";
  reference "ietf-ippm-ioam-direct-export";
}

feature proof-of-transit
{
  description
    "This feature indicated that the proof of transit option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

feature edge-to-edge
{
  description
    "This feature indicated that the edge to edge option is
    supported";
  reference "draft-ietf-ippm-ioam-data";
}

/*
 * IDENTITIES
 */
identity base-filter {
  description
    "Base identity to represent a filter. A filter is used to
    specify the flow to apply the IOAM profile. ";
}
```

```
identity acl-filter {
  base base-filter;
  description
    "Apply ACL rules to specify the flow.";
}

identity base-protocol {
  description
    "Base identity to represent the carrier protocol. It's used to
    indicate what layer and protocol the IOAM data is embedded.";
}

identity ipv6-protocol {
  base base-protocol;
  description
    "The described IOAM data is embedded in IPv6 protocol.";
  reference "ietf-ippm-ioam-ipv6-options";
}

identity nsh-protocol {
  base base-protocol;
  description
    "The described IOAM data is embedded in NSH.";
  reference "ietf-sfc-ioam-nsh";
}

identity base-node-action {
  description
    "Base identity to represent the node actions. It's used to
    indicate what action the node will take.";
}

identity action-encapsulate {
  base base-node-action;
  description
    "indicate the node is to encapsulate the IOAM packet";
}

identity action-decapsulate {
  base base-node-action;
  description
    "indicate the node is to decapsulate the IOAM packet";
}

identity base-trace-type {
  description
    "Base identity to represent trace types";
}
```

```
identity trace-hop-lim-node-id {
  base base-trace-type;
  description
    "indicates presence of Hop_Lim and node_id in the
    node data.";
}

identity trace-if-id {
  base base-trace-type;
  description
    "indicates presence of ingress_if_id and egress_if_id in the
    node data.";
}

identity trace-timestamp-seconds {
  base base-trace-type;
  description
    "indicates presence of time stamp seconds in the node data.";
}

identity trace-timestamp-nanoseconds {
  base base-trace-type;
  description
    "indicates presence of time stamp nanoseconds in the node data.";
}

identity trace-transit-delay {
  base base-trace-type;
  description
    "indicates presence of transit delay in the node data.";
}

identity trace-namespace-data {
  base base-trace-type;
  description
    "indicates presence of namespace specific data (short format)
    in the node data.";
}

identity trace-queue-depth {
  base base-trace-type;
  description
    "indicates presence of queue depth in the node data.";
}

identity trace-opaque-state-snapshot {
  base base-trace-type;
  description
```



```
        "indicates presence of variable length Opaque State Snapshot
        field.";
    }

    identity trace-hop-lim-node-id-wide {
        base base-trace-type;
        description
            "indicates presence of Hop_Lim and node_id wide in the
            node data.";
    }

    identity trace-if-id-wide {
        base base-trace-type;
        description
            "indicates presence of ingress_if_id and egress_if_id wide in
            the node data.";
    }

    identity trace-namespace-data-wide {
        base base-trace-type;
        description
            "indicates presence of namespace specific data in wide format
            in the node data.";
    }

    identity trace-buffer-occupancy {
        base base-trace-type;
        description
            "indicates presence of buffer occupancy in the node data.";
    }

    identity trace-checksum-complement {
        base base-trace-type;
        description
            "indicates presence of the Checksum Complement node data.";
    }

    identity base-pot-type {
        description
            "Base identity to represent Proof of Transit (PoT) types";
    }

    identity pot-bytes-16 {
        base base-pot-type;
        description
            "POT data is a 16 Octet field.";
    }
}
```

```
identity base-e2e-type {
  description
    "Base identity to represent e2e types";
}

identity e2e-seq-num-64 {
  base base-e2e-type;
  description
    "indicates presence of a 64-bit sequence number";
}

identity e2e-seq-num-32 {
  base base-e2e-type;
  description
    "indicates presence of a 32-bit sequence number";
}

identity e2e-timestamp-seconds {
  base base-e2e-type;
  description
    "indicates presence of timestamp seconds for the
    transmission of the frame";
}

identity e2e-timestamp-subseconds {
  base base-e2e-type;
  description
    "indicates presence of timestamp subseconds for the
    transmission of the frame";
}

identity base-namespace {
  description
    "Base identity to represent the namespace";
}

identity namespace-ietf {
  base base-namespace;
  description
    "namespace that specified in IETF.";
}

/*
 * TYPE DEFINITIONS
 */

typedef ioam-filter-type {
  type identityref {
```

```
    base base-filter;
  }
  description
    "Specifies a known type of filter.";
}

typedef ioam-protocol-type {
  type identityref {
    base base-protocol;
  }
  description
    "Specifies a known type of carrier protocol for the IOAM data.";
}

typedef ioam-node-action {
  type identityref {
    base base-node-action;
  }
  description
    "Specifies a known type of node action.";
}

typedef ioam-trace-type {
  type identityref {
    base base-trace-type;
  }
  description
    "Specifies a known trace type.";
}

typedef ioam-pot-type {
  type identityref {
    base base-pot-type;
  }
  description
    "Specifies a known pot type.";
}

typedef ioam-e2e-type {
  type identityref {
    base base-e2e-type;
  }
  description
    "Specifies a known e2e type.";
}

typedef ioam-namespace {
  type identityref {
```

```
    base base-namespace;
  }
  description
    "Specifies the supported namespace.";
}

/*
 * GROUP DEFINITIONS
 */

grouping ioam-filter {
  description "A grouping for IOAM filter definition";

  leaf filter-type {
    type ioam-filter-type;
    description "filter type";
  }

  leaf ace-name {
    when "../filter-type = 'ioam:acl-filter'";
    type leafref {
      path "/acl:acls/acl:acl/acl:aces/acl:ace/acl:name";
    }
    description "Access Control Entry name.";
  }
}

grouping encap-tracing {
  description
    "A grouping for the generic configuration for
    tracing profile.";

  container trace-types {
    description
      "the list of trace types for encapsulate";

    leaf use-namespace {
      type ioam-namespace;
      description
        "the namespace used for the encapsulation";
    }

    leaf-list trace-type {
      type ioam-trace-type;
      description
        "The trace type is only defined at the encapsulation node.";
    }
  }
}
```

```
leaf enable-loopback-mode {
  type boolean;
  default false;
  description
    "Loopback mode is used to send a copy of a packet back towards
    the source. The loopback mode is only defined at the
    encapsulation node.";
}

leaf enable-active-mode {
  type boolean;
  default false;
  description
    "Active mode indicates that a packet is used for active
    measurement. An IOAM decapsulating node that receives a
    packet with the Active flag set in one of its Trace options
    must terminate the packet.";
}
}

grouping ioam-incremental-tracing-profile {
  description
    "A grouping for incremental tracing profile.";

  leaf node-action {
    type ioam-node-action;
    description "node action";
  }

  uses encap-tracing {
    when "node-action = 'ioam:action-encapsulate'";
  }

  leaf max-length {
    when "../node-action = 'ioam:action-encapsulate'";
    type uint32;
    units bytes;
    description
      "This field specifies the maximum length of the node data list
      in octets. The max-length is only defined at the
      encapsulation node. And it's only used for the incremental
      tracing mode.";
  }
}

grouping ioam-preallocated-tracing-profile {
  description
    "A grouping for incremental tracing profile.";
```

```
leaf node-action {
  type ioam-node-action;
  description "node action";
}

uses encap-tracing {
  when "node-action = 'ioam:action-encapsulate'";
}
}

grouping ioam-direct-export-profile {
  description
    "A grouping for direct export profile.";

  leaf node-action {
    type ioam-node-action;
    description "node action";
  }

  uses encap-tracing {
    when "node-action = 'ioam:action-encapsulate'";
  }

  leaf flow-id {
    when "../node-action = 'ioam:action-encapsulate'";
    type uint32;
    description
      "A 32-bit flow identifier. 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.flow-id is used to
      correlate the exported data of the same flow from multiple
      nodes and from multiple packets.";
  }
}

grouping ioam-e2e-profile {
  description
    "A grouping for end to end profile.";

  leaf node-action {
    type ioam-node-action;
    description
      "indicate how the node act for this profile";
  }
}
```

```
    container e2e-types {
      when "../node-action = 'ioam:action-encapsulate'";
      description
        "the list of e2e types for encapsulate";

      leaf use-namespace {
        type ioam-namespace;
        description
          "the namespace used for the encapsulation";
      }

      leaf-list e2e-type {
        type ioam-e2e-type;
        description
          "The e2e type is only defined at the encapsulation node.";
      }
    }
  }

  grouping ioam-admin-config {
    description
      "IOAM top-level administrative configuration.";

    leaf enabled {
      type boolean;
      default false;
      description
        "When true, IOAM configuration is enabled for the system.
        Meanwhile, the IOAM data-plane functionality is enabled.";
    }
  }
}

/*
 * DATA NODES
 */

container ioam {
  description "IOAM top level container";

  container ioam-info {
    config false;
    description
      "Describes assistant information such as units or timestamp
      format. So that monitoring systems can interpret the IOAM
      data.";
  }
}
```

```
container ioam-profiles {
  description
    "Contains a list of IOAM profiles.";

  container admin-config {
    description
      "Contains all the administrative configurations related to
      the IOAM functionalities and all the IOAM profiles.";

    uses ioam-admin-config;
  }

  list ioam-profile {
    key "profile-name";
    ordered-by user;
    description
      "A list of IOAM profiles that configured on the node.";

    leaf profile-name {
      type string;
      mandatory true;
      description
        "Unique identifier for each IOAM profile";
    }

    container filter {
      uses ioam-filter;
      description
        "The filter which is used to indicate the flow to apply
        IOAM.";
    }

    leaf protocol-type {
      type ioam-protocol-type;
      description
        "This item is used to indicate the carrier protocol where
        the IOAM is applied.";
    }
  }

  container incremental-tracing-profile {
    if-feature incremental-trace;
    description
      "describe the profile for incremental tracing option";

    leaf enabled {
      type boolean;
      default false;
      description

```



```
        "When true, apply incremental tracing option to the
        specified flow identified by the filter.";
    }

    uses ioam-incremental-tracing-profile;
}

container preallocated-tracing-profile {
    if-feature preallocated-trace;
    description
        "describe the profile for preallocated tracing option";

    leaf enabled {
        type boolean;
        default false;
        description
            "When true, apply preallocated tracing option to the
            specified flow identified by the following filter.";
    }

    uses ioam-preallocated-tracing-profile;
}

container direct-export-profile {
    if-feature direct-export;
    description
        "describe the profile for direct-export option";

    leaf enabled {
        type boolean;
        default false;
        description
            "When true, apply direct-export option to the
            specified flow identified by the following filter.";
    }

    uses ioam-direct-export-profile;
}

container pot-profile {
    if-feature proof-of-transit;
    description
        "describe the profile for PoT option";

    leaf enabled {
        type boolean;
        default false;
        description

```


preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /ioam/ioam-profiles/admin-config

The items in the container above include the top level administrative configurations related to the IOAM functionalities and all the IOAM profiles. Unexpected changes to these items could lead to the IOAM function disruption and/ or misbehavior of all the IOAM profiles.

- o /ioam/ioam-profiles/ioam-profile

The entries in the list above include the whole IOAM profile configurations which indirectly create or modify the device configurations. Unexpected changes to these entries could lead to the mistake of the IOAM behavior for the corresponding flows.

6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

IANA is requested to assign a new URI from the IETF XML Registry [RFC3688]. The following URI is suggested:

```
URI: urn:ietf:params:xml:ns:yang:ietf-ioam
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document also requests a new YANG module name in the YANG Module Names registry [RFC7950] with the following suggestion:

```
name: ietf-ioam
namespace: urn:ietf:params:xml:ns:yang:ietf-ioam
prefix: ioam
reference: RFC XXXX
```

7. Acknowledgements

For their valuable comments, discussions, and feedback, we wish to acknowledge Greg Mirsky, Reshad Rahman and Tom Petch.

8. References

8.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.
- [I-D.ietf-sfc-proof-of-transit]
Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S. Youell, "Proof of Transit", draft-ietf-sfc-proof-of-transit-08 (work in progress), November 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

8.2. Informative References

- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-05 (work in progress), December 2020.

Appendix A. Examples

This appendix is non-normative.

tbd

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Jim Guichard
Futurewei
United States of America

Email: james.n.guichard@futurewei.com

Frank Brockners
Cisco Systems
Hansaallee 249, 3rd Floor
Duesseldorf, Nordrhein-Westfalen 40549
Germany

Email: fbrockne@cisco.com

Srihari Raghavan
Cisco Systems
Tril Infopark Sez, Ramanujan IT City
Neville Block, 2nd floor, Old Mahabalipuram Road
Chennai, Tamil Nadu 600113
India

Email: srihari@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

Y. Li
T. Sun
H. Yang
D. Chen
China Mobile
February 22, 2021

One-way Delay Measurement Based on Reference Delay
draft-li-ippm-ref-delay-measurement-00

Abstract

The end-to-end network one-way delay is an important performance indicator of the arising 5G network. One type of existing methods requires the end-to-end deployment of accurate clock synchronization mechanism, such as PTP or GPS, which results in relatively high deployment cost. Another type of existing methods uses round-trip delay measurements to estimate the one-way delay. However, since the delay of the downlink and uplink of the 5G network may be asymmetric, the accuracy is relatively low. This document introduces a method to accurately measure end-to-end network one-way delay using reference delay without deploying clock synchronization.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 26, 2021.

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 Used in This Document	4
2.1. Terminology	4
2.2. Requirements Language	4
3. The method of One-way Delay Measurement Based on Reference Delay	4
3.1. One-way Delay Measurement Method	5
3.2. Packet and Measurement Header Format	7
4. Acquisition of Reference Delay	8
5. Security Considerations	9
6. IANA Considerations	9
7. Normative References	9
Authors' Addresses	10

1. Introduction

With the gradual promotion of new-generation network technologies (such as 5G networks) and their application in various industries, SLA guarantees for network quality become more and more important. For example, different 5G services have different requirements for network performance indicators such as delay, jitter, packet loss, and bandwidth. Among them, the 5G network delay is defined as end-to-end one-way delay of the network. Real-time and accurate measurement of the end-to-end one-way delay is very important for the SLA guarantee of network services, and has become an urgent and important requirement.

A common scenario for end-to-end one-way delay measurement, such as a 5G network HD video surveillance service scenario, is shown in figure 1. One end of the network is a high-definition surveillance camera, as shown in the wireless access side on the left in figure 1, and the other end of the network is a video server. The end-to-end one-way delay measurement of the above-mentioned network is the one-way delay from the surveillance camera to the video server, including wireless access network, optical transmission network, 5G core network, and IP data network. The delay is the sum of T1, T2, T3 and T4 as shown in figure 1.

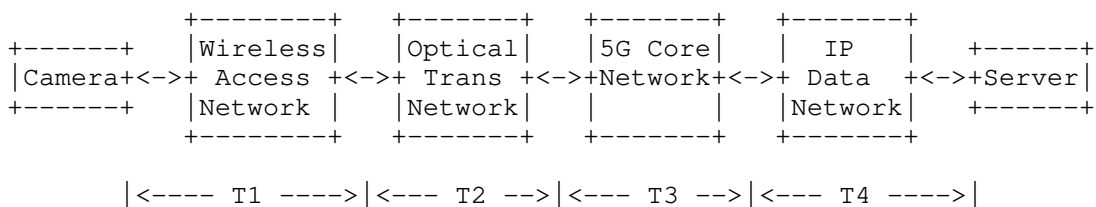


Figure 1:A Scenario for End-to-end One-way Delay

The existing one-way delay measurement solutions are divided into two categories. The first type is a one-way delay measurement solution based on round-trip communication delay, such as TWAMP[RFC8186]. The second type of one-way delay measurement solution is based on accurate network time synchronization mechanism, such as NTP[RFC5905] or PTP[IEEE.1588.2008].

The one-way delay measurement solution based on the round-trip delay requires the deployment of network measurement probes on the source end, namely the surveillance camera side, and the destination end, namely the server side. The source sends the measurement packet; the destination reflects the received measurement packet; after the source receives the measurement packet reflected by the destination, the round-trip delay can be obtained. By dividing the round-trip delay by two, the end-to-end one-way delay can be approximated. However, because the uplink delay and downlink delay of an end-to-end network (such as a 5G network) are not equal, the accuracy of the end-to-end one-way delay approximated by the round-trip communication delay is low. It cannot meet the requirements for accurate end-to-end one-way delay measurement.

The one-way delay measurement solution based on precise network time synchronization requires the deployment of an end-to-end time synchronization mechanism. The current time synchronization accuracy based on the NTP protocol can only reach millisecond level, which cannot fully meet the measurement accuracy requirements, while the time synchronization accuracy based on the GPS module or the PTP protocol can meet the requirements. On the basis of end-to-end time synchronization, the source sends a measurement packet with an accurate egress timestamp; when the destination receives the measurement packet, the accurate ingress timestamp is marked. Since the source and destination have already performed end-to-end time synchronization, the ingress timestamp is subtracted from the egress timestamp to obtain the end-to-end one-way delay. However, for GPS module-based time synchronization solutions, in actual deployment scenarios, many data centers are located underground or in rooms without GPS signals,so GPS clock information cannot be continuously obtained for time synchronization. For time synchronization

solutions based on the PTP protocol, end-to-end deployment is required, which is to say, each device in the wireless access network, optical transmission network, 5G core network and IP data network must support the PTP protocol, which is unrealistic at the moment. So the one-way delay measurement solution based on precise end-to-end time synchronization is expensive and difficult to be deployed.

This document introduces a one-way delay measurement solution based on reference delay. The end-to-end one-way delay of a reference packet with a stable delay is used as the reference delay, which is known in advance and has extremely low jitter. For example, next generation networks are gradually supporting deterministic network[RFC8655] and end-to-end network slicing technologies, which are characterized by bounded end-to-end delay and extremely low jitter of transmitted packets, which are ideal for reference packets. We can use the reference delay provided by the reference packet to indirectly measure the one-way delay of other packet streams under measurement.

2. Conventions Used in This Document

2.1. Terminology

NTP Network Time Protocol

PTP Precision Time Protocol

TWAMP Two-Way Active Measurement Protocol

SLA Service Level Agreement

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. The method of One-way Delay Measurement Based on Reference Delay

The end-to-end one-way delay of a reference packet with a stable delay in the network can be used as a reference delay, denoted as Dref, which is known in advance and has extremely low jitter. This section will describe in detail the end-to-end one-way delay measurement method based on reference delay of the reference packet. Assume that the end-to-end one-way delay from the sender to the

receiver is measured, as shown in figure 2. The intermediate network devices other than the sender and receiver are hidden in the figure.

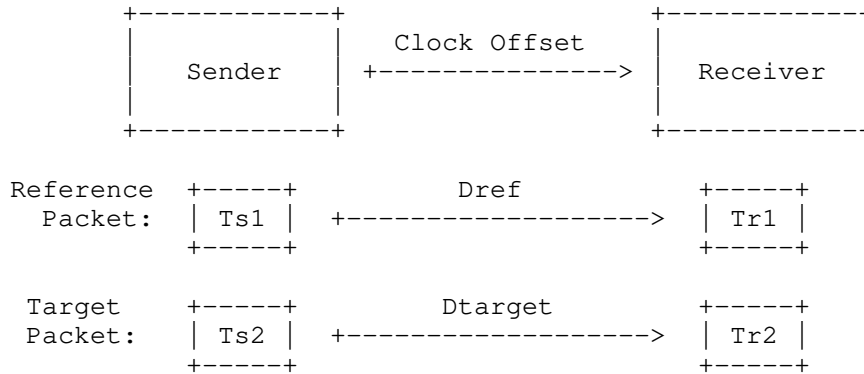


Figure 2:Topology of One-way Delay Measurement

3.1. One-way Delay Measurement Method

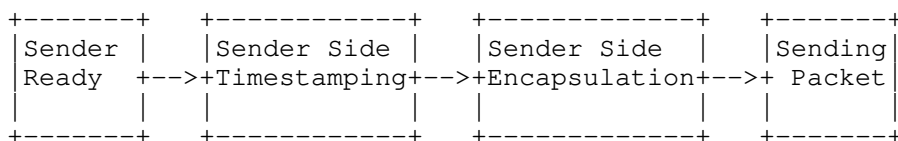
The measurement steps are shown in figure 3, which describe the measurement steps at the sender side and receiver side respectively. For the sender side, a reference packet is sent. In the first step, the sender gets ready to send a reference packet; in the second step, the sender marks an egress timestamp Ts1 for the reference packet; in the third step, the sender encapsulates the egress timestamp of the reference packet in the measurement header of the reference packet; in the fourth step, the sender sends the reference packet. For the target packet, the sender side procedures are the same, we omit it for simplicity. The sending time of the target packet is according to the traffic model of real applications. On the other hand, the sender can send the reference packet according to a fixed frequency or adjust the sending frequency according to the link usage rate, so that the target packet can always find a nearby reference packet to make sure that the sending time interval between the reference packet and the target packet is small.

For the reference packet, the processing steps at the receiver are shown in figure 3. In the first step, the reference packet arrives at the receiver, and the receiver receives the reference packet; in the second step, the receiver timestamps the reference packet at the entrance, which is denoted as Tr1; in the third step, the receiver decapsulates the measurement header of the reference packet to obtain the sender side timestamp Ts1; in the fourth step, the receiver records the timestamp information of Ts1 and Tr1; in the fifth step, the receiver uses the source/destination pair obtained by decapsulation in the third step as the search key, queries the

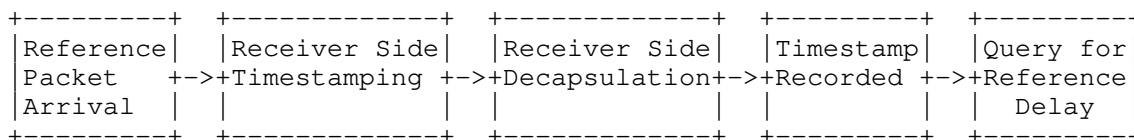
reference delay table and records the reference delay search result, denoted as Dref.

For the target packet, the processing steps at the receiver are also shown in figure 3. In the first step, the target packet arrives at the receiver, and the receiver receives the target packet; in the second step, the receiver timestamps the target packet at the entrance, which is denoted as Tr2; in the third step, the receiver decapsulates the measurement header of the target packet to obtain the sender side timestamp Ts2; in the fourth step, the receiver records the timestamp information of Ts2 and Tr2; in the fifth step, the receiver calculates the target one-way delay, which we want to measure, according to the recorded timestamp information Ts1, Ts2, Tr1, Tr2 and reference delay information Dref. The target one-way delay of the target packet is recorded as Dtarget.

Sender Side Procedures for both Reference and Target Packet:



Receiver Side Procedures for Reference Packet:



Receiver Side Procedures for Target Packet:

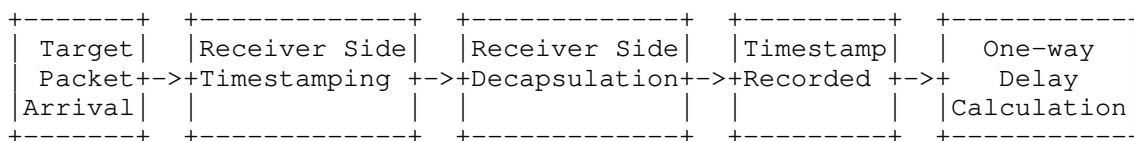


Figure 3: Measurement steps for Sender and Receiver Respectively

Now we describe the fifth step of the receiver procedures for the target packet in figure 3, that is, calculating the one-way delay Dtarget of the target packet based on the recorded timestamp information Ts1, Ts2, Tr1, Tr2 and the reference delay information Dref. The calculation method is the core of this solution. For the

reference packet, leveraging the receiver timestamp minus the sender timestamp, we can get Equation 1.

$$\text{Equation 1: } Tr1 - Ts1 = Dref + \text{Offset1}$$

where Offset1 is the time offset between the sender and the receiver when the reference packet transmission occurs. Similarly, for the target packet, we can get Equation 2 using the same method.

$$\text{Equation 2: } Tr2 - Ts2 = Dtarget + \text{Offset2}$$

where Offset2 is the time offset between the sender and the receiver when the target packet transmission occurs. Assuming that the sending time interval between the reference packet and the target packet is very small, we can get that $\text{Offset1} = \text{Offset2}$. By (Equation 2 - Equation 1), we can get Equation 3.

$$\text{Equation 3: } Dtarget = (Tr2 + Ts1) - (Tr1 + Ts2) + Dref$$

So the one-way delay of the target packet can be calculated by Equation 3.

3.2. Packet and Measurement Header Format

The sender encapsulates the timestamp information and sender-receiver pair information in the measurement header of the sent packet, as shown in figure 4. The position of measurement header is in the option field of the TCP protocol header. Detailed measurement header format is shown in figure 5. The Kind value can be 253 or 254, and the Length value is 8, which is in accordance with TCP option[RFC4727]. The sender ID is one octet, and the receiver ID is also one octet. The sender side timestamp is 4 octets, which can store accurate timestamp information.

specifications of a particular network device. Variable part of the delay is the internal queuing delay. The queuing delay of the device internal buffer is related to the queue depth, queue scheduling algorithm, message priority and message length. For each device along the end-to-end path, the queuing delay can reach microsecond or even millisecond level, depending on values of the above parameters and network congestion state.

With the continuous development of networking technologies and application requirements, a series of new network technologies have emerged which can guarantee bounded end-to-end delay and ultra small jitter. For example, deterministic network[RFC8655], by leveraging novel scheduling algorithms and packet priority settings, can stabilize queuing delay of network device on the end-to-end path. As a result, the end-to-end one-way delay is extremely low and bounded. So packets transmitted by a deterministic network with delay guarantee can be used as reference packets, and their end-to-end one-way delay can be used as reference delays. The acquisition method of reference delay is not limited to the above method based on deterministic network technology.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Normative References

[IEEE.1588.2008]

IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", July 2008.

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

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<https://www.rfc-editor.org/info/rfc4727>>.

- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <<https://www.rfc-editor.org/info/rfc8186>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.

Authors' Addresses

Yang Li
China Mobile
Beijing 100053
China

Email: liyangzn@chinamobile.com

Tao Sun
China Mobile
Beijing 100053
China

Email: suntao@chinamobile.com

Hongwei Yang
China Mobile
Beijing 100053
China

Email: yanghongwei@chinamobile.com

Danyang Chen
China Mobile
Beijing 100053
China

Email: chendanyang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 23, 2021

Z. Li
China Mobile
M. Chen
Huawei
G. Mirsky
ZTE Corp.
February 19, 2021

Simple Two-Way Active Measurement Protocol Extensions for Performance
Measurement on LAG
draft-li-ippm-stamp-on-lag-00

Abstract

This document defines extensions to Simple Two-Way Active Measurement Protocol (STAMP) to implement performance measurement on every member link of a Link Aggregation Group (LAG). Knowing the measured metrics of each member link of a LAG enables operators to enforce a performance metric-based traffic steering policy across the member links.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 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. Micro-Session on LAG	3
3. Micro-STAMP Session	4
3.1. Micro-STAMP-Test	4
3.1.1. LAG Member Link ID TLV	4
3.1.2. Micro-STAMP-Test Procedures	5
4. IANA Considerations	6
5. Security Considerations	6
6. Acknowledgements	6
7. References	6
7.1. Normative References	6
7.2. Informative References	7
Authors' Addresses	7

1. Introduction

Link Aggregation Group (LAG), as defined in [IEEE802.1AX], provides mechanisms to combine multiple physical links into a single logical link. This logical link offers higher bandwidth and better resiliency, because if one of the physical member links fails, the aggregate logical link can continue to forward traffic over the remaining operational physical member links.

Usually, when forwarding traffic over a LAG, a hash-based or similar mechanism is used to load-balance the traffic across the LAG member links. In some cases, the link delays of the member links are different because they are over different transport paths. To provide low delay service to time-sensitive traffic, we have to know the link delay of each member link of a LAG and then steer traffic accordingly. That requires a solution that could measure the performance metrics of each member link of a LAG.

However, when using Simple Two-Way Active Measurement Protocol (STAMP) [RFC8762] to measure a LAG's performance, the LAG is treated as a single logical link/path. The measured metrics reflect the performance of one member link or an average of some/all member links of the LAG.

In addition, for LAG, using passive or hybrid methods (like alternative marking[RFC8321] or iOAM [I-D.ietf-ippm-ioam-data]) can only monitor the link crossed by traffic. It means that the measured metrics reflect the performance of some member links or an average of some/all member links of the LAG. Therefore, in order to measure every link of a LAG, using active methods would be more appropriate.

This document defines extensions to STAMP [RFC8762] to implement performance measurement on every member link of a LAG.

2. Micro-Session on LAG

This document intends to address the scenario (e.g., Figure 1) where a LAG (e.g., the LAG includes three member links) directly connects two nodes (A and B). The goal is to measure the performance of each link of the LAG.

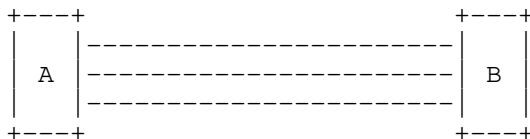


Figure 1: PM for LAG

To measure performance metrics of every member link of a LAG, multiple sessions (one session for each member link) need to be established between the two hosts that are connected by the LAG. These sessions are called micro-sessions for the remainder of this document.

All micro-sessions of a LAG share the same Sender Address, Receiver Address. As for the Sender Port and Receiver Port, the micro-sessions may share the same Sender Port and Receiver Port pair, or each micro session is configured with a different Sender Port and Receiver Port pair. But from simplifying operation point of view, the former is recommended.

In addition, with micro-sessions, there needs a way to correlate a session with a member link. For example, when the Reflector receives a Test packet, it needs to know from which member link the packet is

received, and correlate it with a micro-session. That is a new functionality for STAMP as defined in [RFC8762] and [RFC8972].

Upon receiving a Test packet for a micro-session, the receiver uses the receiving link's identifier to correlate the packet to a particular session. In addition, Test packets may need to carry the member link information for validation checking. For example, when a Session-Sender/Session-Reflector receives a Test packet, it may need to check whether the Test packet is from the expected member link.

3. Micro-STAMP Session

3.1. Micro-STAMP-Test

The micro-STAMP-Test protocol is based on the STAMP-Test protocol [RFC8762] and [RFC8972] with the following extensions.

3.1.1. LAG Member Link ID TLV

The LAG Member Link ID TLV is defined to carry the LAG member link identifiers associated with a micro-STAMP session. The member link identifiers are used for the Session-Sender and Session-Reflector to check whether a Test packet is received from the expected member link. The detailed procedures are defined in Section 3.1.2.

The format is as below:

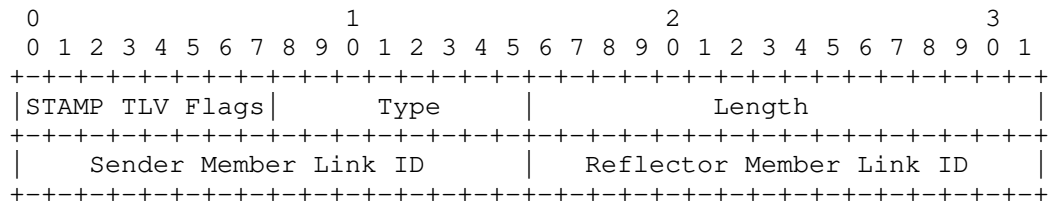


Figure 2: LAG Member Link ID TLV

Sender Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Sender side. The value of the Sender Member Link ID MUST be unique at the Session-Sender.

Reflector Member Link ID (2-octets in length): it is defined to carry the LAG member link identifier of the Reflector side. The value of the Reflector Member ID MUST be unique at the Session-Reflector.

3.1.2. Micro-STAMP-Test Procedures

The micro-STAMP-Test reuses the procedures as defined in Section 4 of STAMP [RFC8762] with the following additions:

The micro-STAMP Session-Sender MUST send the micro-STAMP-Test packets over the member link associated with the session. The micro STAMP Session-Reflector MUST send the reflected Test packets over the receiving member link.

The configuration and management of the association between a micro-STAMP session and the Sender/Reflector member link identifiers are outside the scope of this document.

When the Session-Sender sends a Test packet, the LAG Member Link ID TLV MUST be carried, and the Sender member link identifier associated with the micro-STAMP session MUST be put in the Sender Member Link ID field. If the Session-Sender knows the Reflector member link identifier, it MUST set it as the Reflector Member Link ID field's value. Otherwise, the Reflector Member Link ID field MUST be set to zero.

The Session-Sender uses the Sender Member Link ID field's value to check whether the reflected Test packet is received from the member link associated with the correct micro-STAMP session. Therefore, the Session-Reflector MUST copy the Sender Member Link ID value to the reflected Test packet.

The Session-Reflector uses the Reflector Member Link ID value to check whether a Test packet is received from the member link associated with the correct micro-STAMP session.

The Reflector member link identifier can be obtained from pre-configuration or learned through the data plane (e.g., learned from a reflected Test packet). How to obtain/learn the Reflector member link identifier is outside of this document's scope.

When the micro-STAMP Session-Reflector receives a Test packet, it MUST use the receiving member link to correlate the Test packet to a micro-STAMP session. If there is no such a micro-STAMP session, the Test packet MUST be discarded. Suppose the Reflector Member Link ID is not zero. In that case, the micro-STAMP Session-Reflector MUST use the Reflector member link identifier to check whether it is associated with the micro-STAMP session. If it is not, the Test packet MUST be discarded and no reflected Test packet will be sent back to the Session-Sender. If all validation passed, the Session-Reflector sends a reflected Test packet to the Session-Sender over the receiving member link. The micro-STAMP Session-Reflector MUST

put the Sender and Reflector member link identifiers associated with the micro-STAMP session in the Sender Member Link ID and Reflector Member Link ID fields, respectively. The Sender member link identifier is copied from the received Test packet.

When the micro-STAMP Session-Sender receives a reflected Test packet, it MUST use the receiving member link to correlate the reflected Test packet to a micro-STAMP session. If there is no such a session, the reflected Test packet MUST be discarded. If a matched micro-STAMP session exists, the Session-Sender MUST use the identifier carried in the Sender Member Link ID field to check whether it associates with the session. If the checking failed, the Test packet MUST be discarded.

4. IANA Considerations

This document requires the IANA to allocate the following the TLV type from the "STAMP TLV Types" sub-registry.

Value	Description	Reference
TBD1	LAG Member Link ID	This document

5. Security Considerations

This document does not introduce additional security requirements and mechanisms other than the ones described in [RFC8762] apply to this document.

6. Acknowledgements

The authors would like to thank Min Xiao, Fang Xin for the valuable comments to this work.

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>>.
- [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>>.
- [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

- [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.
- [IEEE802.1AX]
IEEE Std. 802.1AX, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", November 2008.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

Authors' Addresses

Zhenqiang Li
China Mobile

Email: li_zhenqiang@hotmail.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

IPPM
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

M. Cociglio
Telecom Italia
A. Ferrieux
Orange Labs
G. Fioccola
Huawei Technologies
I. Lubashev
Akamai Technologies
F. Bulgarella
Telecom Italia
I. Hamchaoui
Orange Labs
M. Nilo
Telecom Italia
R. Sisto
Politecnico di Torino
D. Tikhonov
LiteSpeed Technologies
February 22, 2021

Explicit Flow Measurements Techniques
draft-mdt-ippm-explicit-flow-measurements-01

Abstract

This document describes protocol independent methods called Explicit Flow Measurement Techniques that employ few marking bits, inside the header of each packet, for loss and delay measurement. The endpoints, marking the traffic, signal these metrics to intermediate observers allowing them to measure connection performance, and to locate the network segment where impairments happen. Different alternatives are considered within this document. These signaling methods apply to all protocols but they are especially valuable when applied to protocols that encrypt transport header and do not allow traditional methods for delay and loss detection.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 26, 2021.

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. Notational Conventions	4
3. Latency Bits	5
3.1. Spin Bit	5
3.2. Delay Bit	6
3.2.1. Generation Phase	8
3.2.2. Reflection Phase	8
3.2.3. T_Max Selection	9
3.2.4. Delay Measurement using Delay Bit	10
3.2.5. Observer's Algorithm	12
3.2.6. Two Bits Delay Measurement: Spin Bit + Delay Bit	13
4. Loss Bits	13
4.1. T Bit - Round Trip Loss Bit	14
4.1.1. Round Trip Packet Loss Measurement	15
4.1.2. Setting the Round Trip Loss Bit on Outgoing Packets	16
4.1.3. Observer's Logic for Round Trip Loss Signal	17
4.1.4. Loss Coverage and Signal Timing	18
4.2. Q Bit - Square Bit	18
4.2.1. Q Block Length Selection	18
4.2.2. Upstream Loss	19
4.2.3. Identifying Q Block Boundaries	20
4.3. L Bit - Loss Event Bit	20
4.3.1. End-To-End Loss	21
4.3.2. Loss Profile Characterization	21

4.4.	L+Q Bits - Upstream, Downstream, and End-to-End Loss Measurements	21
4.4.1.	Correlating End-to-End and Upstream Loss	22
4.5.	R Bit - Reflection Square Bit	23
4.5.1.	R+Q Bits - Using R and Q Bits for Passive Loss Measurement	24
4.5.2.	Enhancement of R Block Length Computation	28
4.5.3.	Improved Resilience to Packet Reordering	28
5.	Summary of Delay and Loss Marking Methods	28
6.	ECN-Echo Event Bit	30
6.1.	Setting the ECN-Echo Event Bit on Outgoing Packets	31
6.2.	Using E Bit for Passive ECN-Reported Congestion Measurement	31
7.	Protocol Ossification Considerations	31
8.	Examples of Application	32
8.1.	QUIC	32
8.2.	TCP	33
9.	Security Considerations	33
9.1.	Optimistic ACK Attack	34
10.	Privacy Considerations	34
11.	IANA Considerations	34
12.	Change Log	35
13.	Contributors	35
14.	Acknowledgements	35
15.	References	35
15.1.	Normative References	35
15.2.	Informative References	36
	Authors' Addresses	38

1. Introduction

Packet loss and delay are hard and pervasive problems of day-to-day network operation. Proactively detecting, measuring, and locating them is crucial to maintaining high QoS and timely resolution of crippling end-to-end throughput issues. To this effect, in a TCP-dominated world, network operators have been heavily relying on information present in the clear in TCP headers: sequence and acknowledgment numbers and SACKs when enabled (see [RFC8517]). These allow for quantitative estimation of packet loss and delay by passive on-path observation. Additionally, the problem can be quickly identified in the network path by moving the passive observer around.

With encrypted protocols, the equivalent transport headers are encrypted and passive packet loss and delay observations are not possible, as described in [TRANSPORT-ENCRYPT].

Measuring TCP loss and delay between similar endpoints cannot be relied upon to evaluate encrypted protocol loss and delay. Different

protocols could be routed by the network differently, and the fraction of Internet traffic delivered using protocols other than TCP is increasing every year. It is imperative to measure packet loss and delay experienced by encrypted protocol users directly.

This document defines Explicit Flow Measurement Techniques. These hybrid measurement path signals (see [IPM-Methods]) are to be embedded into a transport layer protocol and are explicitly intended for exposing RTT and loss rate information to on-path measurement devices. These measurement mechanisms are applicable to any transport-layer protocol, and, as an example, the document describes QUIC and TCP bindings.

The Explicit Flow Measurement Techniques described in this document can be used alone or in combination with other Explicit Flow Measurement Techniques. Each technique uses a small number of bits and exposes a specific measurement.

Following the recommendation in [RFC8558] of making path signals explicit, this document proposes adding a small number of dedicated measurement bits to the clear portion of the protocol headers. These bits can be added to an encrypted portion of a header belonging to any protocol layer, e.g. IP (see [IP]) and IPv6 (see [IPv6]) headers or extensions, such as [IPv6AltMark], UDP surplus space (see [UDP-OPTIONS] and [UDP-SURPLUS]), reserved bits in a QUIC v1 header (see [QUIC-TRANSPORT]).

The measurements are not designed for use in automated control of the network in environments where signal bits are set by untrusted hosts. Instead, the signal is to be used for troubleshooting individual flows as well as for monitoring the network by aggregating information from multiple flows and raising operator alarms if aggregate statistics indicate a potential problem.

The spin bit, delay bit and loss bits explained in this document are inspired by [AltMark], [SPIN-BIT], [I-D.trammell-tsvwg-spin] and [I-D.trammell-ippm-spin].

Additional details about the Performance Measurements for QUIC are described in the paper [ANRW19-PM-QUIC].

2. Notational Conventions

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].

3. Latency Bits

This section introduces bits that can be used for round trip latency measurements. Whenever this section of the specification refers to packets, it is referring only to packets with protocol headers that include the latency bits.

[QUIC-TRANSPORT] introduces an explicit per-flow transport-layer signal for hybrid measurement of RTT. This signal consists of a spin bit that toggles once per RTT. [SPIN-BIT] discusses an additional two-bit Valid Edge Counter (VEC) to compensate for loss and reordering of the spin bit and increase fidelity of the signal in less than ideal network conditions.

This document introduces a stand-alone single-bit delay signal that can be used by passive observers to measure the RTT of a network flow, avoiding the spin bit ambiguities that arise as soon as network conditions deteriorate.

3.1. Spin Bit

This section is a small recap of the spin bit working mechanism. For a comprehensive explanation of the algorithm, please see [SPIN-BIT].

The spin bit is an alternate marking [AltMark] generated signal, where the size of the alternation changes with the flight size each RTT.

The latency spin bit is a single bit signal that toggles once per RTT, enabling latency monitoring of a connection-oriented communication from intermediate observation points.

A "spin period" is a set of packets with the same spin bit value sent during one RTT time interval. A "spin period value" is the value of the spin bit shared by all packets in a spin period.

The client and server maintain an internal per-connection spin value (i.e. 0 or 1) used to set the spin bit on outgoing packets. Both endpoints initialize the spin value to 0 when a new connection starts. Then:

- when the client receives a packet with the packet number larger than any number seen so far, it sets the connection spin value to the opposite value contained in the received packet;
- when the server receives a packet with the packet number larger than any number seen so far, it sets the connection spin value to the same value contained in the received packet.

The computed spin value is used by the endpoints for setting the spin bit on outgoing packets. This mechanism allows the endpoints to generate a square wave such that, by measuring the distance in time between pairs of consecutive edges observed in the same direction, a passive on-path observer can compute the round trip delay of that network flow.

Spin bit enables round trip latency measurement by observing a single direction of the traffic flow.

Note that packet reordering can cause spurious edges that require heuristics to correct. The spin bit performance deteriorates as soon as network impairments arise as explained in Section 3.2.

3.2. Delay Bit

The delay bit has been designed to overcome accuracy limitations experienced by the spin bit under difficult network conditions:

- packet reordering leads to generation of spurious edges and errors in delay estimation;
- loss of edges causes wrong estimation of spin periods and therefore wrong RTT measurements;
- application-limited senders cause the spin bit to measure the application delays instead of network delays.

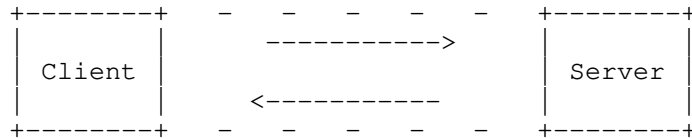
Unlike the spin bit, which is set in every packet transmitted on the network, the delay bit is set only once per round trip.

When the delay bit is used, a single packet with a marked bit (the delay bit) bounces between a client and a server during the entire connection lifetime. This single packet is called "delay sample".

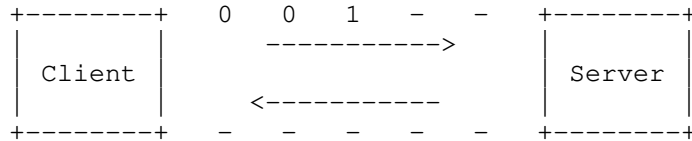
An observer placed at an intermediate point, observing a single direction of traffic, tracking the delay sample and the relative timestamp, can measure the round trip delay of the connection.

The delay sample lifetime is comprised of two phases: initialization and reflection. The initialization is the generation of the delay sample, while the reflection realizes the bounce behavior of this single packet between the two endpoints.

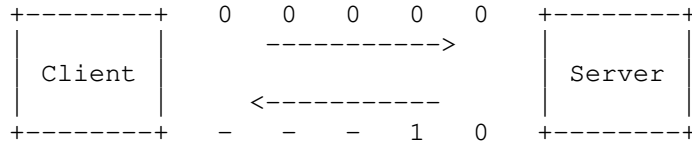
The next figure describes the elementary Delay bit mechanism.



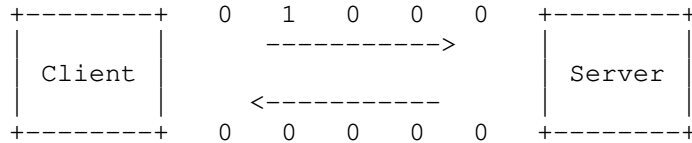
(a) No traffic at beginning.



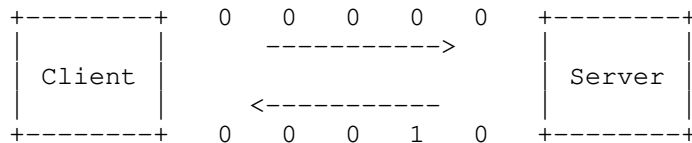
(b) The Client starts sending data and sets the first packet as Delay Sample.



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



(d) The Client reflects the Delay Sample.



(e) The Server reflects the Delay Sample and so on.

Delay bit mechanism

3.2.1. Generation Phase

Only client is actively involved in the generation phase. It maintains an internal per-flow timestamp variable ("ds_time") updated every time a delay sample is transmitted.

When connection starts, the client generates a new delay sample initializing the delay bit of the first outgoing packet to 1. Then it updates the "ds_time" variable with the timestamp of its transmission.

The server initializes the delay bit to 0 at the beginning of the connection, and its only task during the connection is described in Section 3.2.2.

In absence of network impairments, the delay sample should bounce between client and server continuously, for the entire duration of the connection. That is highly unlikely for two reasons:

1. the packet carrying the delay bit might be lost;
2. an endpoint could stop or delay sending packets because the application is limiting the amount of traffic transmitted;

To deal with these problems, the client generates a new delay sample if more than a predetermined time ("T_Max") has elapsed since the last delay sample transmission (including reflections). Note that "T_Max" should be greater than the max measurable RTT on the network. See Section 3.2.3 for details.

3.2.2. Reflection Phase

Reflection is the process that enables the bouncing of the delay sample between a client and a server. The behavior of the two endpoints is almost the same.

- Server side reflection: when a delay sample arrives, the server marks the first packet in the opposite direction as the delay sample.
- Client side reflection: when a delay sample arrives, the client marks the first packet in the opposite direction as the delay sample. It also updates the "ds_time" variable when the outgoing delay sample is actually forwarded.

In both cases, if the outgoing delay sample is being transmitted with a delay greater than a predetermined threshold after the reception of

the incoming delay sample (1ms by default), the delay sample is not reflected, and the outgoing delay bit is kept at 0.

By doing so, the algorithm can reject measurements that would overestimate the delay due to lack of traffic on the endpoints. Hence, the maximum estimation error would amount to twice the threshold (e.g. 2ms) per measurement.

3.2.3. T_Max Selection

The internal "ds_time" variable allows a client to identify delay sample losses. Considering that a lost delay sample is regenerated at the end of an explicit time ("T_Max") since the last generation, this same value can be used by an observer to reject a measure and start a new one.

In other words, if the difference in time between two delay samples is greater or equal than "T_Max", then these cannot be used to produce a delay measure. Therefore the value of "T_Max" must also be known to the on-path network probes.

There are two alternatives to select the "T_Max" value so that both client and observers know it. The first one requires that "T_Max" is known a priori ("T_Max_p") and therefore set within the protocol specifications that implements the marking mechanism (e.g. 1 second which usually is greater than the max expectable RTT). The second alternative requires a dynamic mechanism able to adapt the duration of the "T_Max" to the delay of the connection ("T_Max_c").

For instance, client and observers could use the connection RTT as a basis for calculating an effective "T_Max". They should use a predetermined initial value so that "T_Max = T_Max_p" (e.g. 1 second) and then, when a valid RTT is measured, change "T_Max" accordingly so that "T_Max = T_Max_c". In any case, the selected "T_Max" should be large enough to absorb any possible variations in the connection delay.

"T_Max_c" could be computed as two times the measured "RTT" plus a fixed amount of time ("100ms") to prevent low "T_Max" values in case of very small RTTs. The resulting formula is: "T_Max_c = 2RTT + 100ms". If "T_Max_c" is greater than "T_Max_p" then "T_Max_c" is forced to "T_Max_p" value.

Note that the observer's "T_Max" should always be less than or equal to the client's "T_Max" to avoid considering as a valid measurement what is actually the client's "T_Max". To obtain this result, the client waits for two consecutive incoming samples and computes the two related RTTs. Then it takes the largest of them as the basis of

the "T_Max_c" formula. At this point, observers have already measured a valid RTT and then computed their "T_Max_c".

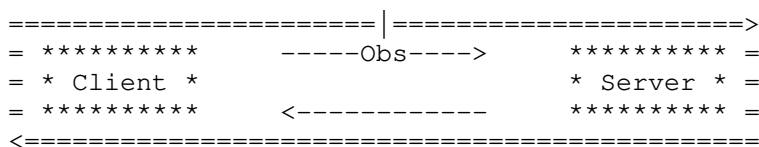
3.2.4. Delay Measurement using Delay Bit

When the Delay Bit is used, a passive observer can use delay samples directly and avoid inherent ambiguities in the calculation of the RTT as can be seen in spin bit analysis.

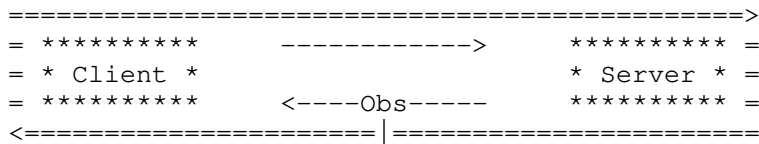
3.2.4.1. RTT Measurement

The delay sample generation process ensures that only one packet marked with the delay bit set to 1 runs back and forth between two endpoints per round trip time. To determine the RTT measurement of a flow, an on-path passive observer computes the time difference between two delay samples observed in a single direction.

To ensure a valid measurement, the observer must verify that the distance in time between the two samples taken into account is less than "T_Max".



(a) client-server RTT



(b) server-client RTT

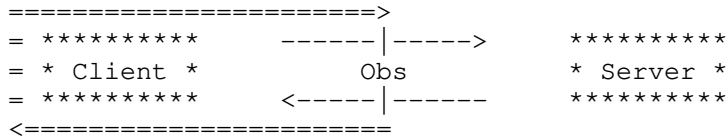
Round-trip time (both direction)

3.2.4.2. Half-RTT Measurement

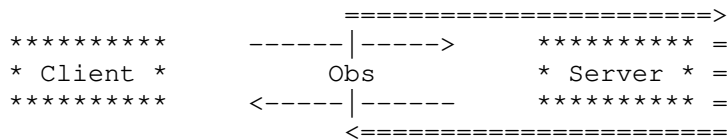
An observer that is able to observe both forward and return traffic directions can use the delay samples to measure "upstream" and "downstream" RTT components, also known as the half-RTT measurements. It does this by measuring the time between a delay sample observed in one direction and the delay sample previously observed in the opposite direction.

As with RTT measurement, the observer must verify that the distance in time between the two samples taken into account is less than "T_Max".

Note that upstream and downstream sections of paths between the endpoints and the observer, i.e. observer-to-client vs client-to-observer and observer-to-server vs server-to-observer, may have different delay characteristics due to the difference in network congestion and other factors.



(a) client-observer half-RTT

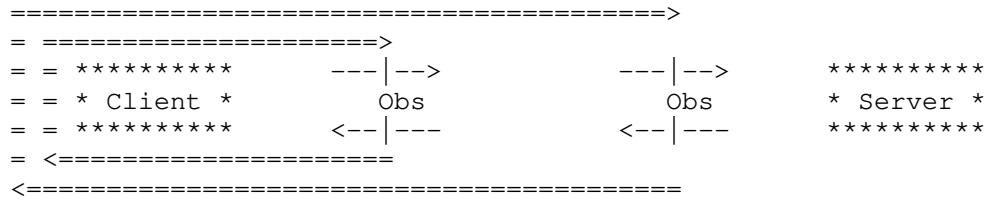


(b) observer-server half-RTT

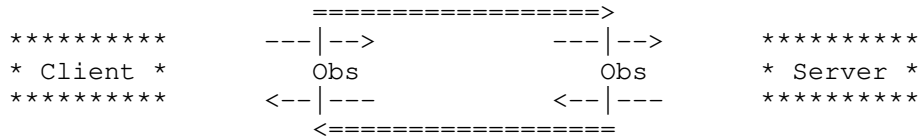
Half Round-trip time (both direction)

3.2.4.3. Intra-Domain RTT Measurement

Intra-domain RTT is the portion of the entire RTT used by a flow to traverse the network of a provider. To measure intra-domain RTT, two observers capable of observing traffic in both directions must be employed simultaneously at ingress and egress of the network to be measured. Intra-domain RTT is difference between the two computed upstream (or downstream) RTT components.



(a) client-observer RTT components (half-RTTs)



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

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

3.2.5. Observer's Algorithm

An on-path observer maintains an internal per-flow variable to keep track of time at which the last delay sample has been observed.

A unidirectional observer, upon detecting a delay sample:

- if a delay sample was also detected previously in the same direction and the distance in time between them is less than "T_Max - K", then the two delay samples can be used to calculate RTT measurement. "K" is a protection threshold to absorb differences in "T_Max" computation and delay variations between two consecutive delay samples (e.g. "K = 10% T_Max").

If the observer can observe both forward and return traffic flows, and it is able to determine which direction contains the client and the server (e.g. by observing the connection handshake), upon detecting a delay sample:

- if a delay sample was also detected in the opposite direction and the distance in time between them is less than "T_Max - K", then the two delay samples can be used to measure the observer-client half-RTT or the observer-server half-RTT, according to the direction of the last delay sample observed.

3.2.6. Two Bits Delay Measurement: Spin Bit + Delay Bit

Spin and Delay bit algorithms work independently. If both marking methods are used in the same connection, observers can choose the best measurement between the two available:

- when a precise measurement can be produced using the delay bit, observers choose it;
- when a delay bit measurement is not available, observers choose the approximate spin bit one.

4. Loss Bits

This section introduces bits that can be used for loss measurements. Whenever this section of the specification refers to packets, it is referring only to packets with protocol headers that include the loss bits - the only packets whose loss can be measured.

- T: the "round Trip loss" bit is used in combination with the Spin bit to measure round-trip loss. See Section 4.1.
- Q: the "square signal" bit is used to measure upstream loss. See Section 4.2.
- L: the "Loss event" bit is used to measure end-to-end loss. See Section 4.3.
- R: the "Reflection square signal" bit is used in combination with Q bit to measure end-to-end loss. See Section 4.1.

Loss measurements enabled by T, Q, and L bits can be implemented by those loss bits alone (T bit requires a working Spin Bit). Two-bit combinations Q+L and Q+R enable additional measurement opportunities discussed below.

Each endpoint maintains appropriate counters independently and separately for each separately identifiable flow (each sub-flow for multipath connections).

Since loss is reported independently for each flow, all bits (except for L bit) require a certain minimum number of packets to be exchanged per flow before any signal can be measured. Therefore, loss measurements work best for flows that transfer more than a minimal amount of data.

4.1. T Bit - Round Trip Loss Bit

The round Trip loss bit is used to mark a variable number of packets exchanged twice between the endpoints realizing a two round-trip reflection. A passive on-path observer, observing either direction, can count and compare the number of marked packets seen during the two reflections, estimating the loss rate experienced by the connection. The overall exchange comprises:

- The client selects, generates and consequently transmits a first train of packets, by setting the T bit to 1;
- The server, upon receiving each packet included in the first train, reflects to the client a respective second train of packets of the same size as the first train received, by setting the T bit to 1;
- The client, upon receiving each packet included in the second train, reflects to the server a respective third train of packets of the same size as the second train received, by setting the T bit to 1;
- The server, upon receiving each packet included in the third train, finally reflects to the client a respective fourth train of packets of the same size as the third train received, by setting the T bit to 1.

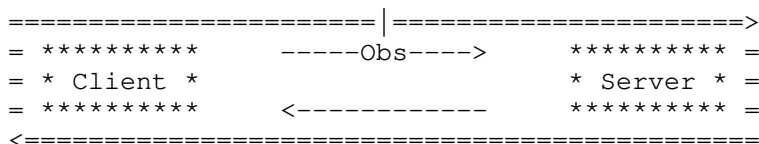
Packets belonging to the first round trip (first and second train) represent the Generation Phase, while those belonging to the second round trip (third and fourth train) represent the Reflection Phase.

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

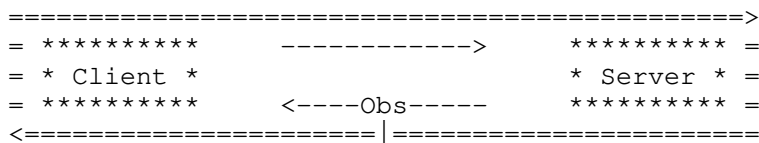
Since packet rates in two directions may be different, the number of marked packets in the train is determined by the direction with the lowest packet rate. See Section 4.1.2 for details on packet generation and for a mechanism to allow an observer to distinguish between trains belonging to different phases (Generation and Reflection).

4.1.1. Round Trip Packet Loss Measurement

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



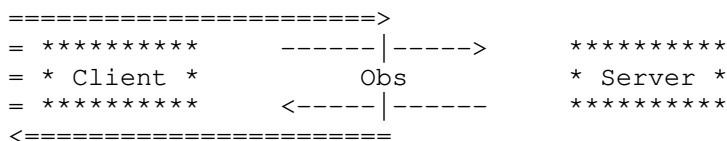
(a) client-server RTPL



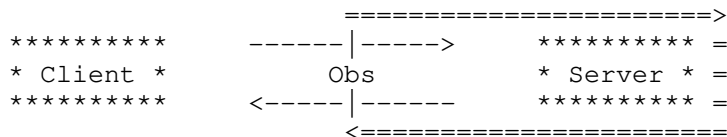
(b) server-client RTPL

Round-trip packet loss (both direction)

This methodology also allows the Half-RTPL measurement and the Intra-domain RTPL measurement in a way similar to RTT measurement.

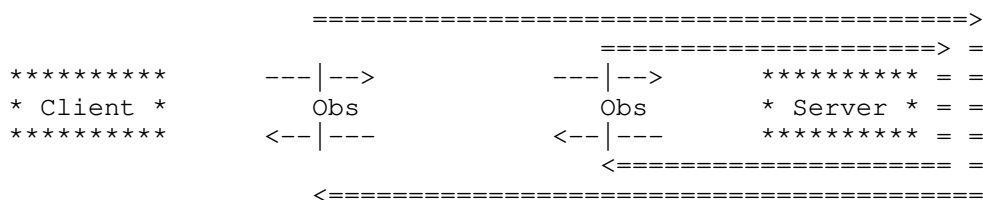


(a) client-observer half-RTPL

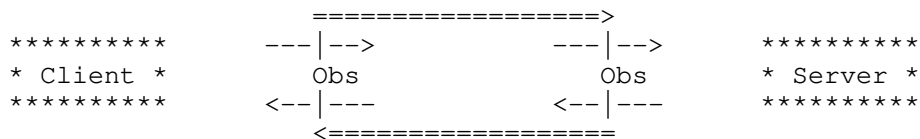


(b) observer-server half-RTPL

Half Round-trip packet loss (both direction)



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



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

Intra-domain Round-trip packet loss (observer-server)

4.1.2. Setting the Round Trip Loss Bit on Outgoing Packets

The round Trip loss signal requires a working Spin-bit signal to separate trains of marked packets (packets with T bit set to 1). A "pause" of at least one empty spin-bit period between each phase of the algorithm serves as such separator for the on-path observer.

The client is in charge of launching trains of marked packets and does so according to the algorithm:

1. Generation Phase. The client starts generating marked packets for two consecutive spin-bit periods; it maintains a "generation token" count that is reset to zero at the beginning of the algorithm phase and is incremented every time a packet arrives. When the client transmits a packet and a "generation token" is available, the client marks the packet and retires a "generation token". If no token is available, the outgoing packet is transmitted unmarked. At the end of the first spin-bit period spent in generation, the reflection counter is unlocked to start counting incoming marked packets that will be reflected later;
2. Pause Phase. When the generation is completed, the client pauses till it has observed one entire spin bit period with no marked packets. That spin bit period is used by the observer as a separator between generated and reflected packets. During this marking pause, all the outgoing packets are transmitted with T

bit set to 0. The reflection counter is still incremented every time a marked packet arrives;

3. Reflection Phase. The client starts transmitting marked packets, decrementing the reflection counter for each transmitted marked packet until the reflection counter reached zero. The "generation token" method from the generation phase is used during this phase as well. At the end of the first spin-period spent in reflection, the reflection counter is locked to avoid incoming reflected packets incrementing it;
4. Pause Phase 2. The pause phase is repeated after the reflection phase and serves as a separator between the reflected packet train and a new packet train.

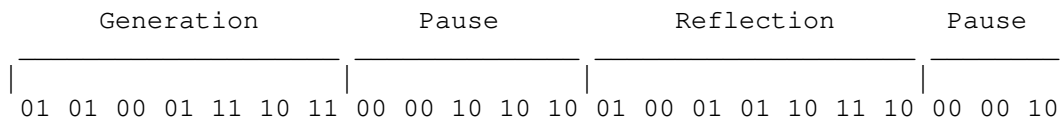
The generation token counter should be capped to limit the effects of a subsequent sudden reduction in the other endpoint's packet rate that could prevent that endpoint from reflecting collected packets. The most conservative cap value is "1".

A server maintains a "marking counter" that starts at zero and is incremented every time a marked packet arrives. When the server transmits a packet and the "marking counter" is positive, the server marks the packet and decrements the "marking counter". If the "marking counter" is zero, the outgoing packet is transmitted unmarked.

4.1.3. Observer's Logic for Round Trip Loss Signal

The on-path observer counts marked packets and separates different trains by detecting spin-bit periods (at least one) with no marked packets. The Round Trip Packet Loss (RTPL) is the difference between the size of the Generation train and the Reflection train.

In the following example, packets are represented by two bits (first one is the spin bit, second one is the loss bit):



Round Trip Loss signal example

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

4.1.4. Loss Coverage and Signal Timing

A cycle of the round Trip loss signaling algorithm contains 2 RTTs of Generation phase, 2 RTTs of Reflection phase, and two Pause phases at least 1 RTT in duration each. Hence, the loss signal is delayed by about 6 RTTs since the loss events.

The observer can only detect loss of marked packets that occurs after its initial observation of the Generation phase and before its subsequent observation of the Reflection phase. Hence, if the loss occurs on the path that sends packets at a lower rate (typically ACKs in such asymmetric scenarios), "2/6" ("1/3") of the packets will be sampled for loss detection.

If the loss occurs on the path that sends packets at a higher rate, " $\text{lowPacketRate}/(3*\text{highPacketRate})$ " of the packets will be sampled for loss detection. For protocols that use ACKs, the portion of packets sampled for loss in the higher rate direction during unidirectional data transfer is " $1/(3*\text{packetsPerAck})$ ", where the value of "packetsPerAck" can vary by protocol, by implementation, and by network conditions.

4.2. Q Bit - Square Bit

The sSquare bit (Q bit) takes its name from the square wave generated by its signal. Every outgoing packet contains the Q bit value, which is initialized to the 0 and inverted after sending N packets (a sSquare Block or simply Q Block). Hence, Q Period is $2*N$. The Q bit represents "packet color" as defined by [AltMark].

Observation points can estimate upstream losses by watching a single direction of the traffic flow and counting the number of packets in each observed Q Block, as described in Section 4.2.2.

4.2.1. Q Block Length Selection

The length of the block must be known to the on-path network probes. There are two alternatives to selecting the Q Block length. The first one requires that the length is known a priori and therefore set within the protocol specifications that implements the marking mechanism. The second requires the sender to select it.

In this latter scenario, the sender is expected to choose N (Q Block length) based on the expected amount of loss and reordering on the path. The choice of N strikes a compromise - the observation could become too unreliable in case of packet reordering and/or severe loss if N is too small, while short flows may not yield a useful upstream loss measurement if N is too large (see Section 4.2.2).

The value of N should be at least 64 and be a power of 2. This requirement allows an Observer to infer the Q Block length by observing one period of the square signal. It also allows the Observer to identify flows that set the loss bits to arbitrary values (see Section 7).

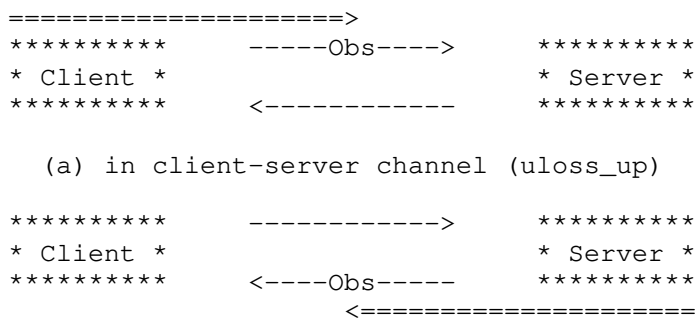
If the sender does not have sufficient information to make an informed decision about Q Block length, the sender should use N=64, since this value has been extensively tried in large-scale field tests and yielded good results. Alternatively, the sender may also choose a random power-of-2 N for each flow, increasing the chances of using a Q Block length that gives the best signal for some flows.

The sender must keep the value of N constant for a given flow.

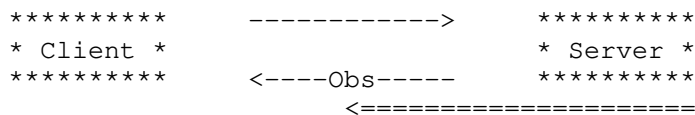
4.2.2. Upstream Loss

Blocks of N (Q Block length) consecutive packets are sent with the same value of the Q bit, followed by another block of N packets with an inverted value of the Q bit. Hence, knowing the value of N, an on-path observer can estimate the amount of upstream loss after observing at least N packets. The upstream loss rate ("uloss") is one minus the average number of packets in a block of packets with the same Q value ("p") divided by N ("uloss=1-avg(p)/N").

The observer needs to be able to tolerate packet reordering that can blur the edges of the square signal, as explained in Section 4.2.3.



(a) in client-server channel (uloss_up)



(b) in server-client channel (uloss_down)

Upstream loss

4.2.3. Identifying Q Block Boundaries

Packet reordering can produce spurious edges in the square signal. To address this, the observer should look for packets with the current Q bit value up to X packets past the first packet with a reverse Q bit value. The value of X, a "Marking Block Threshold", should be less than $N/2$.

The choice of X represents a trade-off between resiliency to reordering and resiliency to loss. A very large Marking Block Threshold will be able to reconstruct Q Blocks despite a significant amount of reordering, but it may erroneously coalesce packets from multiple Q Blocks into fewer Q Blocks, if loss exceeds 50% for some Q Blocks.

4.3. L Bit - Loss Event Bit

The Loss Event bit uses an Unreported Loss counter maintained by the protocol that implements the marking mechanism. To use the Loss Event bit, the protocol must allow the sender to identify lost packets. This is true of protocols such as QUIC, partially true for TCP and SCTP (losses of pure ACKs are not detected) and is not true of protocols such as UDP and IP/IPv6.

The Unreported Loss counter is initialized to 0, and L bit of every outgoing packet indicates whether the Unreported Loss counter is positive (L=1 if the counter is positive, and L=0 otherwise).

The value of the Unreported Loss counter is decremented every time a packet with L=1 is sent.

The value of the Unreported Loss counter is incremented for every packet that the protocol declares lost, using whatever loss detection machinery the protocol employs. If the protocol is able to rescind the loss determination later, a positive Unreported Loss counter may be decremented due to the rescission, but it should NOT become negative due to the rescission.

This loss signaling is similar to loss signaling in [ConEx], except the Loss Event bit is reporting the exact number of lost packets, whereas Echo Loss bit in [ConEx] is reporting an approximate number of lost bytes.

For protocols, such as TCP ([TCP]), that allow network devices to change data segmentation, it is possible that only a part of the packet is lost. In these cases, the sender must increment Unreported Loss counter by the fraction of the packet data lost (so Unreported

Loss counter may become negative when a packet with L=1 is sent after a partial packet has been lost).

Observation points can estimate the end-to-end loss, as determined by the upstream endpoint, by counting packets in this direction with the L bit equal to 1, as described in Section 4.3.1.

4.3.1. End-To-End Loss

The Loss Event bit allows an observer to estimate the end-to-end loss rate by counting packets with L bit value of 0 and 1 for a given flow. The end-to-end loss rate is the fraction of packets with L=1.

The assumption here is that upstream loss affects packets with L=0 and L=1 equally. If some loss is caused by tail-drop in a network device, this may be a simplification. If the sender's congestion controller reduces the packet send rate after loss, there may be a sufficient delay before sending packets with L=1 that they have a greater chance of arriving at the observer.

4.3.2. Loss Profile Characterization

In addition to measuring the end-to-end loss rate, the Loss Event bit allows an observer to characterize loss profile, since the distribution of observed packets with L bit set to 1 roughly corresponds to the distribution of packets lost between 1 RTT and 1 RTO before (see Section 4.4.1). Hence, observing random single instances of L bit set to 1 indicates random single packet loss, while observing blocks of packets with L bit set to 1 indicates loss affecting entire blocks of packets.

4.4. L+Q Bits - Upstream, Downstream, and End-to-End Loss Measurements

Combining L and Q bits allows a passive observer watching a single direction of traffic to accurately measure:

- upstream loss: sender-to-observer loss (see Section 4.2.2)
- downstream loss: observer-to-receiver loss (see Section 4.4.1.1)
- end-to-end loss: sender-to-receiver loss on the observed path (see Section 4.3.1) with loss profile characterization (see Section 4.3.2)

4.4.1. Correlating End-to-End and Upstream Loss

Upstream loss is calculated by observing packets that did not suffer the upstream loss (Section 4.2.2). End-to-end loss, however, is calculated by observing subsequent packets after the sender's protocol detected the loss. Hence, end-to-end loss is generally observed with a delay of between 1 RTT (loss declared due to multiple duplicate acknowledgments) and 1 RTO (loss declared due to a timeout) relative to the upstream loss.

The flow RTT can sometimes be estimated by timing protocol handshake messages. This RTT estimate can be greatly improved by observing a dedicated protocol mechanism for conveying RTT information, such as the Spin bit (see Section 3.1) or Delay bit (see Section 3.2).

Whenever the observer needs to perform a computation that uses both upstream and end-to-end loss rate measurements, it should use upstream loss rate leading the end-to-end loss rate by approximately 1 RTT. If the observer is unable to estimate RTT of the flow, it should accumulate loss measurements over time periods of at least 4 times the typical RTT for the observed flows.

If the calculated upstream loss rate exceeds the end-to-end loss rate calculated in Section 4.3.1, then either the Q Period is too short for the amount of packet reordering or there is observer loss, described in Section 4.4.1.2. If this happens, the observer should adjust the calculated upstream loss rate to match end-to-end loss rate, unless the following applies.

In case of a protocol like TCP and SCTP that does not track losses of pure ACK packets, observing a direction of traffic dominated by pure ACK packets could result in measured upstream loss that is higher than measured end-to-end loss, if said pure ACK packets are lost upstream. Hence, if the measurement is applied to such protocols, and the observer can confirm that pure ACK packets dominate the observed traffic direction, the observer should adjust the calculated end-to-end loss rate to match upstream loss rate.

4.4.1.1. Downstream Loss

Because downstream loss affects only those packets that did not suffer upstream loss, the end-to-end loss rate ("eloss") relates to the upstream loss rate ("uloss") and downstream loss rate ("dloss") as " $(1-uloss)(1-dloss)=1-eloss$ ". Hence, " $dloss=(eloss-uloss)/(1-uloss)$ ".

4.4.1.2. Observer Loss

A typical deployment of a passive observation system includes a network tap device that mirrors network packets of interest to a device that performs analysis and measurement on the mirrored packets. The observer loss is the loss that occurs on the mirror path.

Observer loss affects upstream loss rate measurement, since it causes the observer to account for fewer packets in a block of identical Q bit values (see Section 4.2.2). The end-to-end loss rate measurement, however, is unaffected by the observer loss, since it is a measurement of the fraction of packets with the L bit value of 1, and the observer loss would affect all packets equally (see Section 4.3.1).

The need to adjust the upstream loss rate down to match end-to-end loss rate as described in Section 4.4.1 is an indication of the observer loss, whose magnitude is between the amount of such adjustment and the entirety of the upstream loss measured in Section 4.2.2. Alternatively, a high apparent upstream loss rate could be an indication of significant packet reordering, possibly due to packets belonging to a single flow being multiplexed over several upstream paths with different latency characteristics.

4.5. R Bit - Reflection Square Bit

R bit requires a deployment alongside Q bit. Unlike the square signal for which packets are transmitted into blocks of fixed size, the Reflection square signal (being an alternate marking signal too) produces blocks of packets whose size varies according to these rules:

- when the transmission of a new block starts, its size is set equal to the size of the last Q Block whose reception has been completed;
- if, before transmission of the block is terminated, the reception of at least one further Q Block is completed, the size of the block is updated to the average size of the further received Q Blocks. Implementation details follow.

The Reflection square value is initialized to 0 and is applied to the R-bit of every outgoing packet. The Reflection square value is toggled for the first time when the completion of a Q Block is detected in the incoming square signal (produced by the opposite node using the Q-bit). When this happens, the number of packets ("p"), detected within this first Q Block, is used to generate a reflection

square signal which toggles every "M=p" packets (at first). This new signal produces blocks of M packets (marked using the R-bit) and each of them is called "Reflection Block" (R Block).

The M value is then updated every time a completed Q Block in the incoming square signal is received, following this formula:
"M=round(avg(p))".

The parameter "avg(p)" is the average number of packets in a marking period computed considering all the Q Blocks received since the beginning of the current R Block.

To ensure a proper computation of the M value, endpoints implementing the R bit must identify the boundaries of incoming Q Blocks. The same approach described in {#endmarkingblock} should be used.

Looking at the R-bit, unidirectional observation points have an indication of losses experienced by the entire unobserved channel plus those occurred in the path from the sender up to them.

Since the Q Block is sent in one direction, and the corresponding reflected R Block is sent in the opposite direction, the reflected R signal is transmitted with the packet rate of the slowest direction. Namely, if the observed direction is the slowest, there can be multiple Q Blocks transmitted in the unobserved direction before a complete R Block is transmitted in the observed direction. If the unobserved direction is the slowest, the observed direction can be sending R Blocks of the same size repeatedly before it can update the signal to account for a newly-completed Q Block.

4.5.1. R+Q Bits - Using R and Q Bits for Passive Loss Measurement

Since both square and Reflection square bits are toggled at most every N packets (except for the first transition of the R-bit as explained before), an on-path observer can count the number of packets of each marking block and, knowing the value of N, can estimate the amount of loss experienced by the connection. An observer can calculate different measurements depending on whether it is able to observe a single direction of the traffic or both directions.

Single directional observer:

- upstream loss in the observed direction: the loss between the sender and the observation point (see Section 4.2.2)

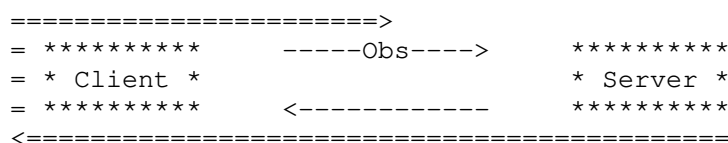
- "three-quarters" connection loss: the loss between the receiver and the sender in the unobserved direction plus the loss between the sender and the observation point in the observed direction
- end-to-end loss in the unobserved direction: the loss between the receiver and the sender in the opposite direction

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

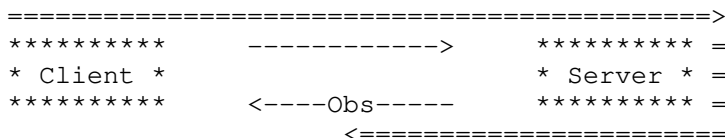
- client-observer half round-trip loss: the loss between the client and the observation point in both directions
- observer-server half round-trip loss: the loss between the observation point and the server in both directions
- downstream loss: the loss between the observation point and the receiver (applicable to both directions)

4.5.1.1. Three-Quarters Connection Loss

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



(a) in client-server channel (tqloss_up)



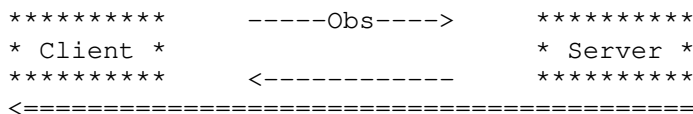
(b) in server-client channel (tqloss_down)

Three-quarters connection loss

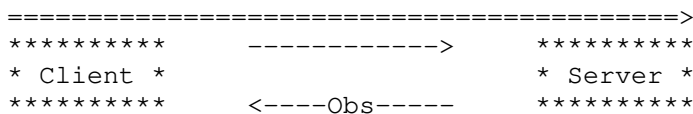
The following metrics derive from this last metric and the upstream loss produced by the Q Bit.

4.5.1.2. End-To-End Loss in the Opposite Direction

End-to-end loss in the unobserved direction ("eloss_unobserved") relates to the "three-quarters" connection loss ("tqloss") and upstream loss in the observed direction ("uloss") as $(1-eloss_unobserved)(1-uloss)=1-tqloss$. Hence, $eloss_unobserved=(tqloss-uloss)/(1-uloss)$.



(a) in client-server channel (eloss_down)

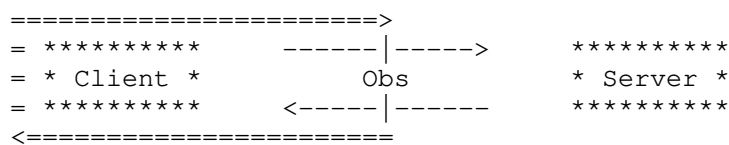


(b) in server-client channel (eloss_up)

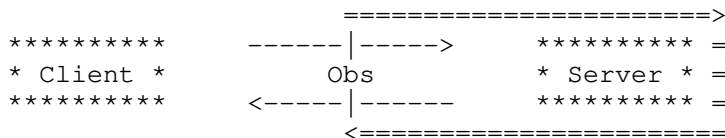
End-To-End loss in the opposite direction

4.5.1.3. Half Round-Trip Loss

If the observer is able to observe both directions of traffic, it is able to calculate two "half round-trip" loss measurements - loss from the observer to the receiver (in a given direction) and then back to the observer in the opposite direction. For both directions, "half round-trip" loss ("hrtloss") relates to "three-quarters" connection loss ("tqloss_opposite") measured in the opposite direction and the upstream loss ("uloss") measured in the given direction as " $(1-uloss)(1-hrtloss)=1-tqloss_opposite$ ". Hence, " $hrtloss=(tqloss_opposite-uloss)/(1-uloss)$ ".



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



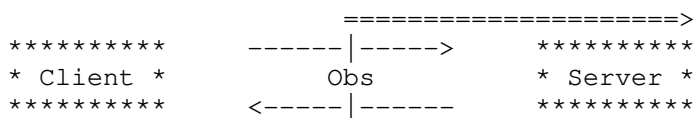
(b) observer-server half round-trip loss (hrtloss_os)

Half Round-trip loss (both direction)

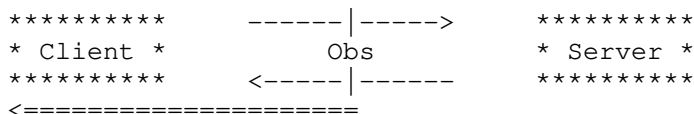
4.5.1.4. Downstream Loss

If the observer is able to observe both directions of traffic, it is able to calculate two downstream loss measurements using either end-to-end loss and upstream loss, similar to the calculation in Section 4.4.1.1 or using "half round-trip" loss and upstream loss in the opposite direction.

For the latter, " $dloss=(hrtloss-uloss_opposite)/(1-uloss_opposite)$ ".



(a) in client-server channel (dloss_up)



(b) in server-client channel (dloss_down)

Downstream loss

4.5.2. Enhancement of R Block Length Computation

The use of the rounding function used in the M computation introduces errors that can be minimized by storing the rounding applied each time M is computed, and using it during the computation of the M value in the following R Block.

This can be achieved introducing the new "r_avg" parameter in the computation of M. The new formula is "Mr=avg(p)+r_avg; M=round(Mr); r_avg=Mr-M" where the initial value of "r_avg" is equal to 0.

4.5.3. Improved Resilience to Packet Reordering

When a protocol implementing the marking mechanism is able to detect when packets are received out of order, it can improve resilience to packet reordering beyond what is possible using methods described in Section 4.2.3.

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

5. Summary of Delay and Loss Marking Methods

This section summarizes the marking methods described in this draft.

For the Delay measurement, it is possible to use the spin bit and/or the delay bit. A unidirectional or bidirectional observer can be used.

Method	# of bits	Available Delay Metrics		Impairments Resiliency
		UNIDIR Observer	BIDIR Observer	
S: Spin Bit	1	RTT	x2 Half RTT	low
D: Delay Bit	1	RTT	x2 Half RTT	high
SD: Spin Bit & Delay Bit *	2	RTT	x2 Half RTT	high

x2 Same metric for both directions

* Both algorithms work independtly; an observer could use approximate spin bit measures when delay bit ones aren't available

Figure 1: Delay Comparison

For the Loss measurement, each row in the table of Figure 2 represents a loss marking method. For each method the table specifies the number of bits required in the header, the available metrics using an unidirectional or bidirectional observer, applicable protocols, measurement fidelity and delay.

Method	Bits	Available Loss Metrics		Protocols	Measurement Aspects	
		UNIDIR Observer	BIDIR Observer		Fidelity	Delay
T: Round Trip Loss Bit	\$ 1	RT	x2 Half RT	*	Rate by sampling 1/3 to 1/(3*ppa) of pkts over 2 RTT	~6 RTT
Q: Square Bit	1	Upstream	x2	*	Rate over N pkts (e.g. 64)	N pkts (e.g. 64)
L: Loss Event Bit	1	E2E	x2	#	Loss shape (and rate)	Min: RTT Max: RTO
QL: Square + Loss Ev. Bits	2	Upstream Downstream E2E	x2 x2 x2	#	-> see Q -> see Q L -> see L	Up: see Q Others: see L
QR: Square + Ref. Sq. Bits	2	Upstream 3/4 RT !E2E	x2 x2 E2E Downstream Half RT	*	Rate over N*ppa pkts (see Q bit for N)	Up: see Q Others: N*ppa pk (see Q for N)

* All protocols
 # Protocols employing loss detection (w/ or w/o pure ACK loss detection)
 \$ Require a working spin bit
 ! Metric relative to the opposite channel
 x2 Same metric for both directions
 ppa Packets-Per-Ack
 Q|L See Q if Upstream loss is significant; L otherwise

Figure 2: Loss Comparison

6. ECN-Echo Event Bit

While the primary focus of the draft is on exposing packet loss and delay, modern networks can report congestion before they are forced to drop packets, as described in [ECN]. When transport protocols keep ECN-Echo feedback under encryption, this signal cannot be observed by the network operators. When tasked with diagnosing

network performance problems, knowledge of a congestion downstream of an observation point can be instrumental.

If downstream congestion information is desired, this information can be signaled with an additional bit.

- E: The "ECN-Echo Event" bit is set to 0 or 1 according to the Unreported ECN Echo counter, as explained below in Section 6.1.

6.1. Setting the ECN-Echo Event Bit on Outgoing Packets

The Unreported ECN-Echo counter operates identically to Unreported Loss counter (Section 4.3), except it counts packets delivered by the network with CE markings, according to the ECN-Echo feedback from the receiver.

This ECN-Echo signaling is similar to ECN signaling in [ConEx]. ECN-Echo mechanism in QUIC provides the number of packets received with CE marks. For protocols like TCP, the method described in [ConEx-TCP] can be employed. As stated in [ConEx-TCP], such feedback can be further improved using a method described in [ACCURATE].

6.2. Using E Bit for Passive ECN-Reported Congestion Measurement

A network observer can count packets with CE codepoint and determine the upstream CE-marking rate directly.

Observation points can also estimate ECN-reported end-to-end congestion by counting packets in this direction with a E bit equal to 1.

The upstream CE-marking rate and end-to-end ECN-reported congestion can provide information about downstream CE-marking rate. Presence of E bits along with L bits, however, can somewhat confound precise estimates of upstream and downstream CE-markings in case the flow contains packets that are not ECN-capable.

7. Protocol Ossification Considerations

Accurate loss and delay information is not critical to the operation of any protocol, though its presence for a sufficient number of flows is important for the operation of networks.

The delay and loss bits are amenable to "greasing" described in [RFC8701], if the protocol designers are not ready to dedicate (and ossify) bits used for loss reporting to this function. The greasing could be accomplished similarly to the Latency Spin bit greasing in [QUIC-TRANSPORT]. Namely, implementations could decide that a

fraction of flows should not encode loss and delay information and, instead, the bits would be set to arbitrary values. The observers would need to be ready to ignore flows with delay and loss information more resembling noise than the expected signal.

8. Examples of Application

8.1. QUIC

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

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

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

```

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|0|1|S|D|T|K|P|P|
+---+---+---+---+---+---+
    
```

Scheme 1

- alternatively, a two bits loss signal (QL or QR) can be placed in both reserved bits; the proposed schemes, in this case, are:

```

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|0|1|S|Q|L|K|P|P|
+---+---+---+---+---+---+
    
```

Scheme 2A

```

0 1 2 3 4 5 6 7
+---+---+---+---+---+---+
|0|1|S|Q|R|K|P|P|
+---+---+---+---+---+---+
    
```

Scheme 2B

A further option would be to substitute the spin bit with the delay bit leaving the two reserved bits for loss detection. The proposed schemes are:

```

  0 1 2 3 4 5 6 7
  +---+---+---+---+
  |0|1|D|Q|L|K|P|P|
  +---+---+---+---+

```

Scheme 3A

```

  0 1 2 3 4 5 6 7
  +---+---+---+---+
  |0|1|D|Q|R|K|P|P|
  +---+---+---+---+

```

Scheme 3B

8.2. TCP

The signals can be added to TCP by defining bit 4 of byte 13 of the TCP header to carry the spin bit or the delay bit, and possibly bits 5 and 6 to carry additional information, like the delay bit and the round-trip loss bit (DT), or a two bits loss signal (QL or QR).

9. Security Considerations

Passive loss and delay observations have been a part of the network operations for a long time, so exposing loss and delay information to the network does not add new security concerns for protocols that are currently observable.

In the absence of packet loss, Q and R bits signals do not provide any information that cannot be observed by simply counting packets transiting a network path. In the presence of packet loss, Q and R bits will disclose the loss, but this is information about the environment and not the endpoint state. The L bit signal discloses internal state of the protocol's loss detection machinery, but this state can often be gleaned by timing packets and observing congestion controller response.

Hence, loss bits do not provide a viable new mechanism to attack data integrity and secrecy.

9.1. Optimistic ACK Attack

A defense against an Optimistic ACK Attack, described in [QUIC-TRANSPORT], involves a sender randomly skipping packet numbers to detect a receiver acknowledging packet numbers that have never been received. The Q bit signal may inform the attacker which packet numbers were skipped on purpose and which had been actually lost (and are, therefore, safe for the attacker to acknowledge). To use the Q bit for this purpose, the attacker must first receive at least an entire Q Block of packets, which renders the attack ineffective against a delay-sensitive congestion controller.

A protocol that is more susceptible to an Optimistic ACK Attack with the loss signal provided by Q bit and uses a loss-based congestion controller, should shorten the current Q Block by the number of skipped packets numbers. For example, skipping a single packet number will invert the square signal one outgoing packet sooner.

Similar considerations apply to the R Bit, although a shortened R Block along with a matching skip in packet numbers does not necessarily imply a lost packet, since it could be due to a lost packet on the reverse path along with a deliberately skipped packet by the sender.

10. Privacy Considerations

To minimize unintentional exposure of information, loss bits provide an explicit loss signal - a preferred way to share information per [RFC8558].

New protocols commonly have specific privacy goals, and loss reporting must ensure that loss information does not compromise those privacy goals. For example, [QUIC-TRANSPORT] allows changing Connection IDs in the middle of a connection to reduce the likelihood of a passive observer linking old and new sub-flows to the same device. A QUIC implementation would need to reset all counters when it changes the destination (IP address or UDP port) or the Connection ID used for outgoing packets. It would also need to avoid incrementing Unreported Loss counter for loss of packets sent to a different destination or with a different Connection ID.

11. IANA Considerations

This document makes no request of IANA.

12. Change Log

TBD

13. Contributors

The following people provided valuable contributions to this document:

- Marcus Ihlar, Ericsson, marcus.ihlar@ericsson.com
- Jari Arkko, Ericsson, jari.arkko@ericsson.com
- Emile Stephan, Orange, emile.stephan@orange.com

14. Acknowledgements

TBD

15. References

15.1. Normative References

- [ConEx] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <<https://www.rfc-editor.org/info/rfc7713>>.
- [ConEx-TCP] Kuehlewind, M., Ed. and R. Scheffenegger, "TCP Modifications for Congestion Exposure (ConEx)", RFC 7786, DOI 10.17487/RFC7786, May 2016, <<https://www.rfc-editor.org/info/rfc7786>>.
- [ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [IP] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [IPM-Methods] 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>>.

- [IPv6] 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>>.
- [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>>.
- [RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/info/rfc8558>>.
- [TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

15.2. Informative References

- [ACCURATE] Briscoe, B., Kuehlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", draft-ietf-tcpm-accurate-ecn-13 (work in progress), November 2020.
- [AltMark] 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>>.
- [ANRW19-PM-QUIC] Bulgarella, F., Cociglio, M., Fioccola, G., Marchetto, G., and R. Sisto, "Performance measurements of QUIC communications", Proceedings of the Applied Networking Research Workshop, DOI 10.1145/3340301.3341127, July 2019.
- [I-D.trammell-ippm-spin] Trammell, B., "An Explicit Transport-Layer Signal for Hybrid RTT Measurement", draft-trammell-ippm-spin-00 (work in progress), January 2019.
- [I-D.trammell-tsvwg-spin] Trammell, B., "A Transport-Independent Explicit Signal for Hybrid RTT Measurement", draft-trammell-tsvwg-spin-00 (work in progress), July 2018.

[IPv6AltMark]

Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-02 (work in progress), October 2020.

[QUIC-TRANSPORT]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-34 (work in progress), January 2021.

[RFC8517] Dolson, D., Ed., Snellman, J., Boucadair, M., Ed., and C. Jacquenet, "An Inventory of Transport-Centric Functions Provided by Middleboxes: An Operator Perspective", RFC 8517, DOI 10.17487/RFC8517, February 2019, <<https://www.rfc-editor.org/info/rfc8517>>.

[RFC8701] Benjamin, D., "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility", RFC 8701, DOI 10.17487/RFC8701, January 2020, <<https://www.rfc-editor.org/info/rfc8701>>.

[SPIN-BIT]

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

[TRANSPORT-ENCRYPT]

Fairhurst, G. and C. Perkins, "Considerations around Transport Header Confidentiality, Network Operations, and the Evolution of Internet Transport Protocols", draft-ietf-tsvwg-transport-encrypt-18 (work in progress), November 2020.

[UDP-OPTIONS]

Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-09 (work in progress), November 2020.

[UDP-SURPLUS]

Herbert, T., "UDP Surplus Header", draft-herbert-udp-space-hdr-01 (work in progress), July 2019.

Authors' Addresses

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

E`Mail`: mauro.cociglio@telecomitalia.it

Alexandre Ferrieux
Orange Labs

E`Mail`: alexandre.ferrieux@orange.com

Giuseppe Fioccola
Huawei Technologies
Riesstrasse, 25
Munich 80992
Germany

E`Mail`: giuseppe.fioccola@huawei.com

Igor Lubashev
Akamai Technologies

E`Mail`: ilubashe@akamai.com

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

E`Mail`: fabio.bulgarella@guest.telecomitalia.it

Isabelle Hamchaoui
Orange Labs

E`Mail`: isabelle.hamchaoui@orange.com

Massimo Nilo
Telecom Italia

EMail: massimo.nilo@telecomitalia.it

Riccardo Sisto
Politecnico di Torino

EMail: riccardo.sisto@polito.it

Dmitri Tikhonov
LiteSpeed Technologies

EMail: dtikhonov@litespeedtech.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 27 September 2021

G. Mirsky
X. Min
ZTE Corp.
L. Han
China Mobile
26 March 2021

Error Performance Measurement in Packet-switched Networks
draft-mirsky-ippm-epm-03

Abstract

This document describes the use of the error performance metric to characterize a packet-switched network's conformance to the pre-defined set of performance objectives. In this document, metrics that characterize error performance in a packet-switched network (PSN) are defined, as well as methods to measure and calculate them. Also, the requirements for an active Operation, Administration, and Maintenance protocol to support the error performance measurement in PSN are discussed, and potential candidate protocols are analyzed. All metrics and measurement methods are equally applicable to underlay and overlay networks.

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 September 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. Conventions used in this document	3
2.1. Terminology and Acronyms	3
2.2. Requirements Language	4
3. Error Performance Metrics	4
3.1. Measure Error Performance Metrics	4
3.2. Calculate Error Performance Metrics	5
4. Requirements to EPM	5
5. Active OAM Protocol for EPM	6
6. IANA Considerations	6
7. Security Considerations	6
8. Acknowledgments	6
9. References	6
9.1. Normative References	6
9.2. Informative References	7
Authors' Addresses	7

1. Introduction

Operations, Administration, and Maintenance (OAM) is a collection of methods to detect, characterize, localize failures in a network, and monitor the network's performance using various measurement methods. Traditionally, the former set of OAM tools identified as Fault Management (FM) OAM. The latter - Performance Monitoring (PM) OAM. Some OAM protocols can be used for both groups of tasks, while some serve one particular group. But regardless of how many OAM protocols are in use, network operators and network users are faced with multiple metrics that characterize the network conditions. This document describes a new component of packet-switched network (PSN) OAM.

Error performance measurement (EPM) is a part of an OAM toolset that provides an operator with information related to network measurements for a uni-directional or a bidirectional connection between two systems. In current technology, EPM has been defined only for data communication methods that have a constant bit-rate transmission [ITU.G.826] and not for PSN, where transmissions are statistically

random. As a statistically multiplexed network in a PSN, a receiver node does not expect a packet to arrive from a sender node at a specific moment, less from a particular sender. That is what differentiates PSN from networks built on a constant bit-rate transmission, where a stream of bits between two nodes is always present, whether it represents data or not. That provides the receiver with a predictable number of measurements in a series of measurement intervals. In PSN, on-path OAM methods, i.e., measurement methods that use data flow, cannot provide such predictability and thus be used for EPM. In PSN, EPM needs to use active OAM methods, per definition in [RFC7799]. This document identifies metrics that characterize PSN error performance and methods to measure and calculate them. Also, the requirements for an active OAM protocol to support EPM in PSN are discussed, and potential candidate protocols are analyzed.

2. Conventions used in this document

2.1. Terminology and Acronyms

OAM Operations, Administration, and Maintenance

EP Error Performance

EPM Error Performance Measurement

ES Errored Second

ESR Errored Second Ratio

SES Severely Errored Second

SESR Severely Errored Second Ratio

EFS Error-Free Second

PSN Packet-switched Network

FM Fault Management

PM Performance Monitoring

2.2. Requirements Language

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

3. Error Performance Metrics

When analyzing the error performance of a path between two nodes, we need to select a time interval as the unit of EPM. In [ITU.G.826], a time interval of one second is used. It is reasonable to use the same time interval for EPM for PSNs. Further, for the purpose of EPM, each time interval, i.e., second, is classified either as Errored Second (ES), Severely Errored Second (SES), or Error-Free Second (EFS). These are defined as follows:

- * An ES is a time interval during which at least one of the performance parameters degraded below its optimal level threshold or a defect was detected.
- * An SES is a time interval during which at least one the performance parameters degraded below its critical threshold or a defect was detected.
- * Consequently, an EFS is a time interval during which all performance objectives are at or above their respective optimal levels, and no defect has been detected.

The definition of a state of a defect in the network is also necessary for understanding the EPM. In this document, the defect is interpreted as the state of inability to communicate between a particular set of nodes. It is important to note that it is being defined as a state, and thus, it has conditions that define entry into it and exit out of it. Also, the state of defect exists only in connection to the particular group of nodes in the network, not the network as a domain.

3.1. Measure Error Performance Metrics

The definitions of ES, SES, and EFS allow for characterization of the communication between two nodes relative to the level of required and acceptable performance and when performance degrades below the acceptable level. The former condition in this document referred to as network availability. The latter - network unavailability. Based on the definitions, SES is the one-second of network unavailability while ES and EFS present an interval of network availability. But

since the conditions of network are everchanging periods of network availability and unavailability need to be defined with duration larger than one-second interval to reduce the number of state changes while correctly reflecting the network condition. The method to determine the state of the network in terms of EPM OAM is described below:

- * If ten consecutive SES intervals been detected, then the EPM state of the network determined as unavailability and the beginning of that period of unavailability state is at the start of the first SES in the sequence of the consecutive SES intervals.
- * Similarly, ten consecutive non-SES intervals, i.e., either ES or EFS, indicate that the network is in the availability period, i.e., available. The start of that period is at the beginning of the first non-SES interval.
- * Resulting from these two definitions, a sequence of less than ten consecutive SES or non-SES intervals does not change the EPM state of the network. For example, if the EPM state is determined as unavailability, a sequence of seven EFS intervals is not viewed as an availability period.

3.2. Calculate Error Performance Metrics

Determining the period in which the path is currently EP-wise is helpful. But because switching between periods requires ten consecutive one-second intervals, conditions that last shorter intervals may not be adequately reflected. Two additional EP OAM metrics can be used, and they are defined as follows:

- * errored second ratio (ESR) is the ratio of ES to the total number of seconds in a time of the availability periods during a fixed measurement interval.
- * severely errored second ratio (SESR) - is the ratio of SES to the total number of seconds in a time of the availability periods during a fixed measurement interval.

4. Requirements to EPM

TBA

5. Active OAM Protocol for EPM

Digital communication methods characterized as the constant-bit rate digital paths and connections allow measurement of the error performance without using an active OAM. That is possible because a predictable flow of digital signals is expected at an egress system. That is not the case for packet-switched networks that are based on the principle of statistical multiplexing flows. The latter usually improves the utilization of the communication network's resources, but it also makes the flow unpredictable for the egress system. For that reason, an active OAM has to be used in measuring the error performance in a network. A combination of OAM protocols can provide the necessary for EPM functionality. For example, Bidirectional Forwarding Detection (BFD) [RFC5880] can be used to monitor the continuity of a path between the ingress and egress systems. And STAMP [RFC8762] can be used to measure and calculate performance metrics that are used as Service Level Objectives. But using two protocols and correlating the state of the network from them adds to the complexity in network operation.

The Integrated OAM, described in [I-D.mmm-rtgwg-integrated-oam], combines lightweight FM OAM with the comprehensive set of performance measurement methods. PM component of the Integrated OAM is based on [RFC6374] that supports, among other measurement methods, one-way and two-way packet loss and packet delay measurements.

6. IANA Considerations

TBA

7. Security Considerations

TBA

8. Acknowledgments

TBA

9. References

9.1. Normative References

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

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

9.2. Informative References

- [I-D.mmm-rtgwg-integrated-oam]
Mirsky, G., Min, X., and G. Mishra, "Integrated Operation, Administration, and Maintenance", Work in Progress, Internet-Draft, draft-mmm-rtgwg-integrated-oam-00, 18 February 2021, <<https://tools.ietf.org/html/draft-mmm-rtgwg-integrated-oam-00>>.
- [ITU.G.826]
ITU-T, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T G.826, December 2002.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <<https://www.rfc-editor.org/info/rfc8762>>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.com

Xiao Min
ZTE Corp.

Email: xiao.min2@zte.com.cn

Liuyan Han
China Mobile
32 XuanWuMenXi Street
Beijing
100053
China

Email: hanliuyan@chinamobile.com

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: 1 October 2021

G. Mirsky
ZTE Corp.
W. Lingqiang
G. Zhui
ZTE Corporation
H. Song
Futurewei Technologies
30 March 2021

Hybrid Two-Step Performance Measurement Method
draft-mirsky-ippm-hybrid-two-step-09

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 1 October 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. 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	6
4.2. Operation of the HTS Intermediate Node	8
4.3. Operation of the HTS Egress Node	9
4.4. Considerations for HTS Timers	10
4.5. Deploying HTS in a Multicast Network	10
5. Authentication in HTS	11
6. IANA Considerations	12
6.1. IOAM Option-Type for HTS	12
6.2. HTS TLV Registry	12
6.3. HTS Sub-TLV Type Sub-registry	13
6.4. HMAC Type Sub-registry	14
7. Security Considerations	15
8. Acknowledgments	15
9. References	15
9.1. Normative References	15
9.2. Informative References	16
Authors' Addresses	17

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, 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.

4. Theory of Operation

The HTS method consists of two phases:

- * performing a measurement or obtaining network state information, one or more than one type, on a node;
- * collecting and transporting the measurement.

HTS uses 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 [I-D.ietf-ippm-ioam-data]. A packet in the flow to which the Alternate-Marking method [RFC8321] 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.

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. The follow-up packet contains characteristic information, copied from the trigger packet, sufficient for participating HTS nodes to associate it with the original packet. The exact composition of the characteristic information is specific for each transport network, and its definition is outside the scope

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

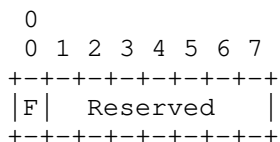


Figure 2: Flags Field Format

Flags is eight-bits long. The format of the Flags field displayed in Figure 2.

- 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 16 bits-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.

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.

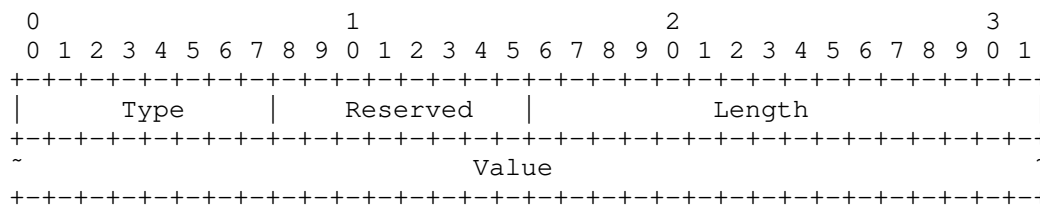


Figure 3: Telemetry Data TLV Format

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 3 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 MTU, 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 4.

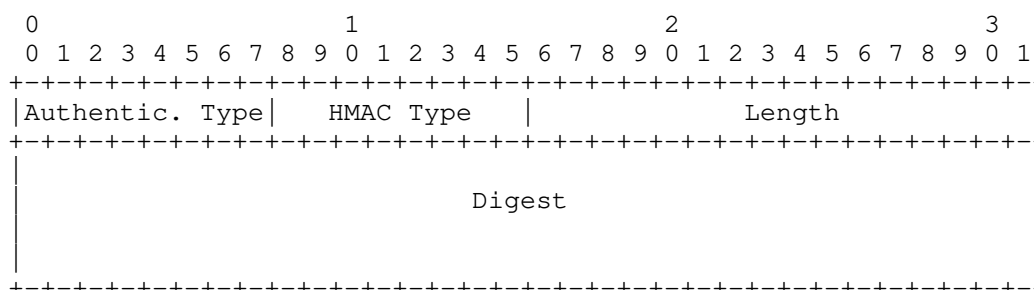


Figure 4: HMAC sub-TLV

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 1) 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-12, 21 February 2021, <<https://tools.ietf.org/html/draft-ietf-ippm-ioam-data-12>>.
- [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-03, 17 February 2021, <<https://tools.ietf.org/html/draft-ietf-ippm-ioam-direct-export-03>>.
- [I-D.song-ippm-postcard-based-telemetry] Song, H., Mirsky, G., Filsfils, C., Abdelsalam, A., Zhou, T., Li, Z., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", Work in Progress, Internet-Draft, draft-song-ippm-postcard-based-telemetry-09, 19 February 2021, <<https://tools.ietf.org/html/draft-song-ippm-postcard-based-telemetry-09>>.
- [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>>.
- [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>>.
- [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

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com, gregory.mirsky@ztetx.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

IPPM
Internet-Draft
Intended status: Standards Track
Expires: June 3, 2021

H. Song
D. Linda
Futurewei Technologies
November 30, 2020

In-band Edge-to-Edge Round Trip Time Measurement
draft-song-ippm-inband-e2e-rtt-measurement-00

Abstract

This draft describes a lightweight in-band edge-to-edge network round trip time measurement architecture and suggests implementations.

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 June 3, 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. In-band E2E RTT Measurement Architecture	3
3. Implementation Considerations	4
4. Security Considerations	5
5. IANA Considerations	6
6. Contributors	6
7. Acknowledgments	6
8. References	6
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

1. Introduction

In-network service-based traffic engineering or load balancing needs to monitor particular flows' edge-to-edge performance, such as round trip time (RTT), in the operator's network domain. The host-based ping using ICMPv6 [RFC4443] is of no use because it is usually beyond the access of network operators. The router-based ping, as an active measurement approach, cannot reflect the real performance of the specific flows under scrutiny. This is also true for the other active measurement approaches such as TWAMP [RFC5357].

In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] supports in-band flow-based performance measurement. However, on the one hand, the IOAM trace option is too heavy for the applications which do not care about the per-hop performance; on the other hand, the IOAM E2E option only supports the one-way measurement.

Alternate Marking(AM) [RFC8321], mainly designed for one-way measurement, can be used to measure the two-way edge-to-edge delay if both edges initiate a one-way measurement session. However, AM's measurement interval needs to be large enough to avoid the measurement ambiguity, and it requires both edges to conduct the measurements and export results to a controller.

We need a lightweight in-band flow RTT measurement method. "Lightweight" means the extra header overhead is low, and the extra network processing overhead is also low. A network operator should be able to pick a flow to monitor and get fine-grained per-packet RTT

measurement for edge to edge. Depending on the application scenario and the network domain scope, the edge can extend to the host, the network interface card (NIC), or the network switch or router. To this end, we propose an in-band edge-to-edge RTT measurement method and suggest the implementation approaches.

Such measurement only reflects the network delay for a flow but excludes the application layer delay incurred by server or client.

2. In-band E2E RTT Measurement Architecture

The measurement architecture is shown in Figure 1. The controller, either on a remote machine or on the edge node's control plane, configures the ingress edge node to measure some flow's RTT between the ingress edge and the egress edge. The ingress edge node uses ACL to filter the flow packets and, at given interval or probability, add the timestamp and the other metadata to the selected packets. The egress edge, after capturing the data, either piggyback the data on a reverse flow packet, or generate a feedback packet carrying the data back to the ingress edge node. Once the ingress edge node receives the feedback data, it sends the data along with the current timestamp to the controller. The controller can then calculate the flow RTT and react with followup actions.

The RTT calculation is done in the slow path, the metadata incurs only small and fix header overhead, and the nodes in the domain does not do any processing. All these make the measurement lightweight, accurate, and have little impact to the network forwarding performance.

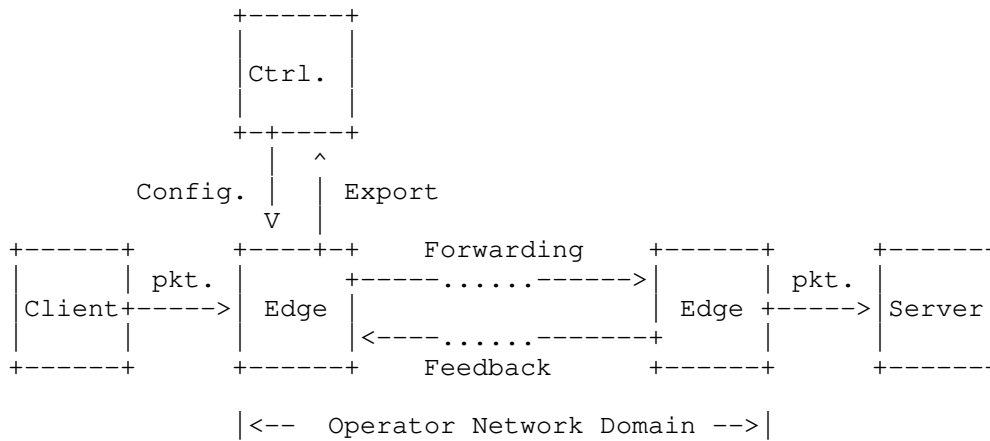


Figure 1: In-band E2E RTT Measurement

To differentiate a feedback packet from an original packet, a flag needs to be raised in the feedback. Optionally, to correlate a feedback with its original packet, the original packet can also include an identifier (e.g., a sequence number) which the feedback packet will carry back as well. The ingress edge node can use the reverse flow ID plus the identifier to pair an original packet with its feedback.

The feedback can also include some other local data at the egress edge (e.g., the egress edge node ID or the egress flow statistics) other than simply reflecting the original data back.

3. Implementation Considerations

One approach to implement the in-band E2E RTT measurement is to use the IOAM E2E option augmented with the feedback mechanism. Current IOAM E2E option only sends one-way data from one edge to the other edge. The data fields can include the ingress edge timestamp which is exactly what is needed. Moreover, the data fields can also include a packet sequence number used for correlating the feedback packet with the original packet. However, current IOAM E2E option lacks a feedback mechanism. It has no flag field reserved in its current option header specification, so it is not easy to indicate the feedback packets.

To enable the two-way measurement behavior, we need to add some indicator to the IOAM E2E option header to indicate the request for a feedback. We also need another indicator to tell if the current packet is a feedback.

To support this, we can either introduce another IOAM two-way E2E option while keeping the current IOAM E2E option unchanged, or simply modify the current IOAM E2E option header specification to extend its usage. The simplest modification is to reserve a few flag bits and among them, two bits are used for the two-way measurement. One possible layout is shown in Figure 2. Alternatively, the flags can take several bits from the Namespace-ID field.

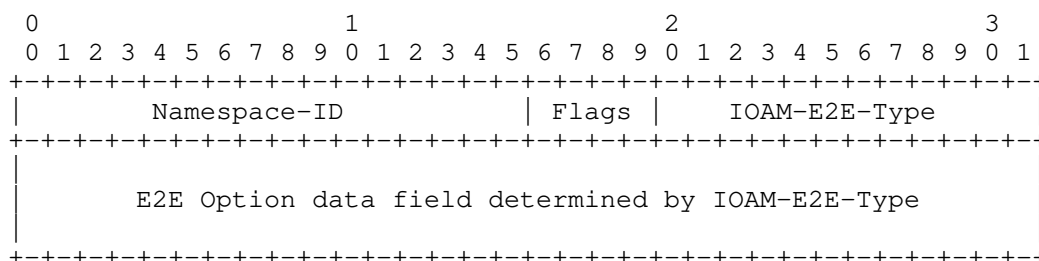


Figure 2: Modified IOAM E2E Option Header

The data field can carry the timestamp, the sequence number, of a unique packet identifier number. Other data types can also be carried to enrich the feedback information.

A packet can serve as both a forward packet and a feedback packet when both flags are set. In this case, there are two records for each data type in the data field. The forward packet's data are located in front of the feedback packet's data.

4. Security Considerations

To prevent the timestamp to be maliciously altered during the packet forwarding, the ingress edge can instead keep the timestamp locally and only send a packet identifier (e.g., a random data). When a reverse flow packet carrying the same identifier is received, the current timestamp along with the saved timestamp are forwarded to the controller.

The ingress edge node can limit the frequency of measurement to the flow packets. The egress edge node can also rate limit the feedback. So the potential DoS attack can be mitigated.

5. IANA Considerations

Depending on the discussion output, either a registry for a new IOAM option is required or a modification to the current IOAM E2E option specification is needed.

6. Contributors

TBD.

7. Acknowledgments

TBD.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

- [I-D.ietf-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.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [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>>.

[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
Santa Clara
USA

Email: haoyu.song@futurewei.com

Linda Dunbar
Futurewei Technologies
Plano
USA

Email: linda.dunbar@futurewei.com

IPPM
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2021

H. Song
Futurewei
Z. Li
S. Peng
Huawei Technologies
J. Guichard
Futurewei
January 4, 2021

Approaches on Supporting IOAM in IPv6
draft-song-ippm-ioam-ipv6-support-02

Abstract

It has been proposed to encapsulate IOAM tracing option data fields in IPv6 HbH options header. However, due to size of the trace data and the extension header location in the IPv6 packets, the proposal creates practical challenges for implementation, especially when other extension headers, such as a routing header, also exist and require in-network processing. We propose several alternative approaches to address this challenge, including separating the IOAM trace data to a different extension header, using the postcard-based telemetry (e.g., IOAM DEX) instead, and applying the segment IOAM trace export scheme, based on the network scenario and application requirements. We discuss the pros and cons of each approach and hope to foster standard convergence and industry adoption.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on July 8, 2021.

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. IOAM Data Separate and Postpose	4
2.1. IOAM Trace Data Encapsulation	5
3. Segment IOAM Data Export	5
3.1. Independent of SRv6	5
3.2. Export at SRv6 node	6
4. Direct Export Option	7
5. Comparison	7
6. Security Considerations	8
7. Normative References	8
Authors' Addresses	10

1. Introduction

In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] defines two tracing options, pre-allocated tracing option and incremental tracing option, which record hop-by-hop data along a packet's forwarding path. The draft [I-D.ietf-ippm-ioam-ipv6-options] proposes the method to encapsulate IOAM trace option data fields in IPv6. Because the tracing options requires per hop processing, such options can only be encapsulated in IPv6 Hop-by-Hop (HbH) options header. The draft [I-D.ioametal-ippm-6man-ioam-ipv6-deployment] further describes some deployment approaches.

[RFC8200] mandates that the HbH options header, if exists, must be the first extension header following the IPv6 header. However, the IOAM trace data can be large, which easily amount to tens to hundreds of bytes, making accessing other headers after it difficult or even impossible. There are practical limitations on how far the hardware

can reach into a packet in forwarding hardware. The IOAM tracing option cannot be applied if it makes other extension headers inaccessible. Even if the other headers can be reached, the deeper they are, the higher the cost to access and process them, and the lower the forwarding performance. A potentially more detrimental issue is that the incremental tracing option will expand the HbH header at each hop and push back all other headers, which keeps shifting the locations of the other extension headers, further complicating the hardware implementation and impeding the forwarding.

The issue becomes more severe when SRv6 and IOAM coexist. The Segment Routing Extension Header (SRH) [I-D.ietf-6man-segment-routing-header] is encapsulated in a routing header which is after the HbH options header. SRH itself can be large. It requires read and write operations at each SRv6 node. If it is deeply embedded in a packet and its location keeps shifting, either it is beyond the reach of hardware or the forwarding performance degrades.

We can avoid the problem by simply not using both at the same time, but apparently this is not ideal, because IOAM is an important OAM tool and it is even more wanted when SRv6 brings more operational complexity into IPv6 networks.

Our second recourse is to limit the IOAM to SRv6 nodes only. That is, consider SRv6 as an overlay tunnel over IPv6 and apply the IOAM pipe mode as discussed in [I-D.song-ippm-ioam-tunnel-mode], which only collects data at each SRv6 nodes. To realize this, [I-D.ali-spring-ioam-srv6] describes an approach that encapsulates the IOAM option data fields in an SRH TLV. [I-D.song-6man-srv6-pbt] and [I-D.ietf-6man-spring-srv6-oam] describe another approach to enable postcard-based telemetry for SRv6 without needing IOAM option encapsulation. In either case, the SRH is close to the packet front and its location is fixed. [I-D.song-spring-siam] proposes to support IOAM in the payload of the dedicated SRv6 probing packets only. While these approaches are useful for use cases that only need to monitor the segment end points, it fails to cover all the IPv6 nodes in a network.

So the proposition of this draft is, suppose we need to apply IOAM on all nodes in an SRv6 network, how we can amend the approach in [I-D.ietf-ippm-ioam-ipv6-options] or use alternative approaches to circumvent the aforementioned issues. In this draft, we propose three such approaches: (1) separating the IOAM trace data to a different extension header, (2) using the postcard-based telemetry (e.g., IOAM DEX) instead, and (3) applying the segment IOAM trace export scheme. We discuss the pros and cons of each approach and hope to foster standard convergence and industry adoption.

2. IOAM Data Separate and Postpose

An IOAM trace type data fields contain two parts: instruction and trace data. Although by convention the trace data part immediately follow the instruction part, there is not fundamental reason why these two parts must stick together. This observation provides us an optimization opportunity to amend the original proposal in [I-D.ietf-ippm-ioam-ipv6-options].

We separate the IOAM trace type data fields into the instruction part and the trace data part. We encapsulate only the instruction part in the HbH options header, and encapsulate the trace data part in another metadata extension header after all the IPv6 extension headers and before upper layer protocol headers. This arrangement allows high performance hardware implementation. When using the incremental data trace, even if the data trace size increases at each node, all IPv6 extension headers remain intact and new data is inserted at a fixed location.

Figure 1 shows the HbH option format for IOAM trace type instruction. The field specification is identical to that in [RFC8200] and [I-D.ietf-ippm-ioam-data].

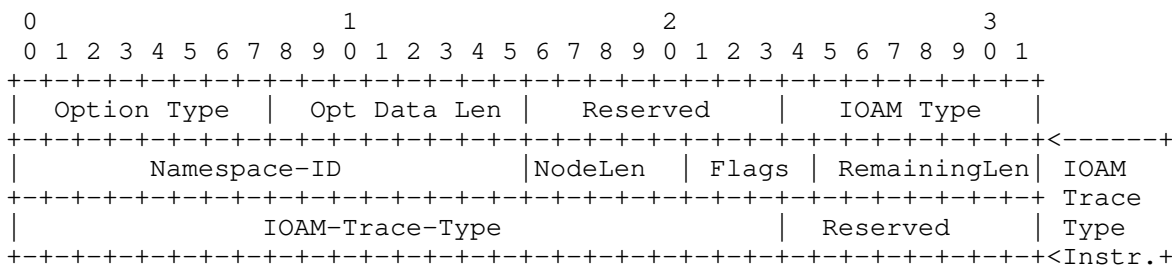


Figure 1: HbH Option Format for IOAM Trace Type Instruction

Figure 2 shows the TLV option format for IOAM trace type data. The IOAM trace type data format is compliant with [I-D.ietf-ippm-ioam-data].

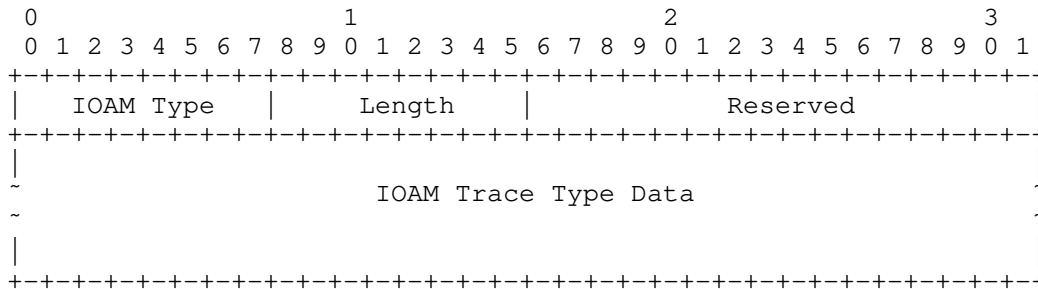


Figure 2: Option Format for IOAM Trace Type Data

2.1. IOAM Trace Data Encapsulation

We have basically two methods to encapsulate the IOAM trace data. First, we can define a new IPv6 extension header which is dedicated to metadata. Once standardized, this extension header can also be used to host potential metadata from other applications such as NSH for SFC [RFC8300]. Second, this option can be carried as a TLV option in another existing extension header such as the destination header. The only requirement is that this extension header should be the last one in the extension header chain. The first method is cleaner but it requires extra standard effort; the second method is simpler but it needs to overcome the access constraints exerted by [RFC8300].

3. Segment IOAM Data Export

If the overhead of the IOAM trace type data fields is under control, we may still manage to encapsulate both instruction and data in HbH options header as in [I-D.ietf-ippm-ioam-ipv6-options]. To this end, we introduce two sub-approaches.

3.1. Independent of SRv6

[I-D.song-ippm-segment-ioam] proposes an enhancement to IOAM trace type which can configure the allowable overhead of the IOAM trace type data fields. Once the trace data size is up to the limit at a network node (i.e., a segment or a fixed number of network nodes are traversed), the trace data will be stripped and exported so room is made to accommodate new trace data from nodes in the next segment of the forwarding path.

This approach requires some moderate updates to the IOAM trace type data fields, as described in [I-D.song-ippm-segment-ioam]. Figure 3 shows the format of the HbH Option Header containing Segment IOAM trace type data fields. A flag bit (#23) in the Flags field is used

to indicate the current header is a segment IOAM header. In this context, the last octet in the IOAM header is partitioned into two 4-bit nibbles. The first nibble (SSize) is used to save the segment size and the second nibble (RHop) is used to save the remaining hops. This limits the maximum segment size to 15.

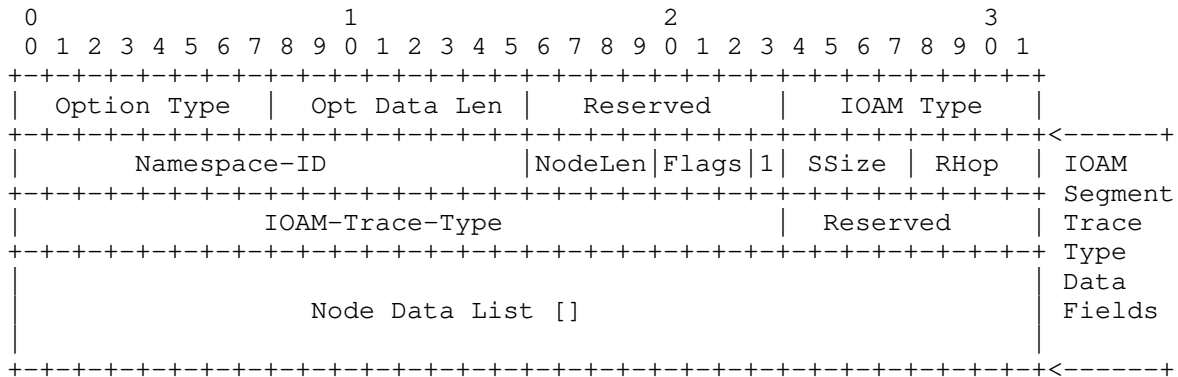


Figure 3: HbH Option Format for Segment IOAM Trace Type Data Fields

At the beginning of each segment, the segment size (SSize) and the remaining hops (RHop) are initialized: RHop is set to equal to SSize. At each hop, if RHop is not zero, the node data is added to the node data list and then RHop is decremented by 1. If RHop is equal to 0 when receiving the packet, the node needs to remove (in incremental trace option) or clear (in pre-allocated trace option) the IOAM node data list and reset RHop to SSize.

In this case, if we use the IOAM pre-allocated trace type, the size and location of each IPv6 extension header is fixed and predictable, and the hardware capability and performance can be guaranteed.

3.2. Export at SRv6 node

Whenever a packet with the IOAM option reaches a SRv6 node which needs to access the SRH, we can configure the node to export immediately the IOAM trace data accumulated so far. In this case, basically at each SRv6 node, the HbH header size is fixed and the header contains an IOAM option with only the instruction part. After the SRH processing, this node can add local IOAM trace data in the HbH option header before forwarding the packet.

The incremental trace type can be used in this approach. In an extreme case when every node is also an SRv6 node, this approach regresses to a per-hop postcard-based telemetry approach as described in [I-D.song-ippm-postcard-based-telemetry]. In this case, the HbH

option for IOAM can even be avoided altogether if we can find a way to simply mark the packet for postcard export.

4. Direct Export Option

As an embodiment of the PBT-I approach introduced in [I-D.song-ippm-postcard-based-telemetry], IOAM Direct Export (DEX) Option Type discussed in [I-D.ioamteam-ippm-ioam-direct-export] can be used to replace the IOAM trace type. IOAM DEX only needs to encapsulate a fix-size instruction header in the HbH option header.

Figure 4 shows the HbH option format for IOAM DEX type fields. The field specification is identical to that in [RFC8200] and [I-D.ioamteam-ippm-ioam-direct-export].

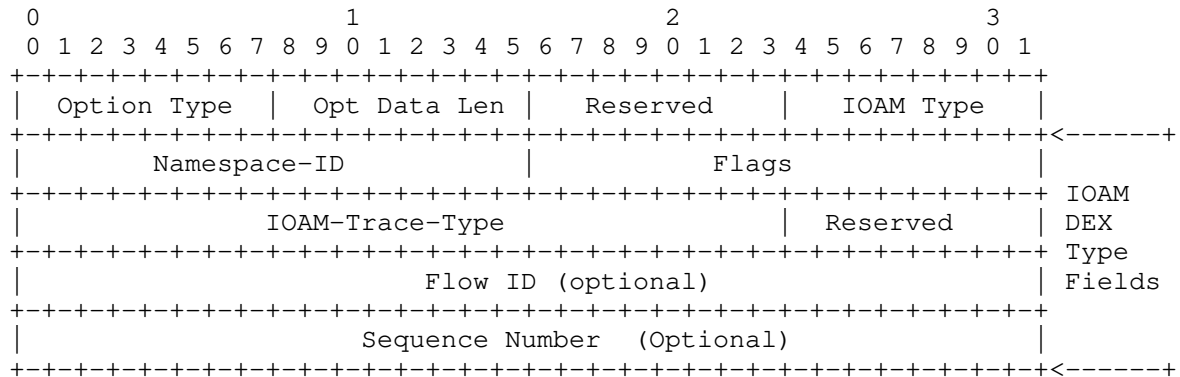


Figure 4: HbH Option Format for IOAM DEX Type Fields

5. Comparison

The following table compares the existing approach and the four other alternative approaches proposed in this draft.

Approach	Pros	Cons
IOAM Trace in HbH	Comply w/ IOAM Data Spec	Variable and long HbH header impeding access of later extension headers
IOAM Trace Data Separate and Postpose (Sec. 2)	Fix-size and short HbH header, good for later extension header access	Need extra extension header to hold trace data
Segment IOAM Data Export (Sec. 3.1)	Fix-size and controllable HbH header size	Need to enhance IOAM trace type data field spec.
Trace Export at SRv6 nodes (Sec. 3.2)	Can be done through configuration	Specific to SRv6; No better than PB & IOAM DEX in the worst case
IOAM Direct Export in HbH (Sec. 4)	Comply w/ IOAM DEX Spec; Fix-size and short HbH	Need export data correlation

Figure 5: Comparison of Different Approaches

6. Security Considerations

TBD.

7. Normative References

[I-D.ali-spring-ioam-srv6]

Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Nainar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-03 (work in progress), November 2020.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

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

Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

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

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

[I-D.ietf-ippm-ioam-ipv6-options]

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

[I-D.ioametal-ippm-6man-ioam-ipv6-deployment]

Bhandari, S., Brockners, F., Mizrahi, T., Kfir, A., Gafni, B., Spiegel, M., Krishnan, S., and M. Smith, "Deployment Considerations for In-situ OAM with IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-deployment-03 (work in progress), March 2020.

[I-D.ioamteam-ippm-ioam-direct-export]

Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.

[I-D.song-6man-srv6-pbt]

Song, H., "Support Postcard-Based Telemetry for SRv6 OAM", draft-song-6man-srv6-pbt-01 (work in progress), October 2019.

[I-D.song-ippm-ioam-tunnel-mode]

Song, H., Li, Z., Zhou, T., and Z. Wang, "In-situ OAM Processing in Tunnels", draft-song-ippm-ioam-tunnel-mode-00 (work in progress), June 2018.

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

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

- [I-D.song-ippm-segment-ioam]
Song, H. and T. Zhou, "Control In-situ OAM Overhead with Segment IOAM", draft-song-ippm-segment-ioam-01 (work in progress), April 2018.
- [I-D.song-spring-siam]
Song, H. and T. Pan, "SRv6 In-situ Active Measurement", draft-song-spring-siam-00 (work in progress), December 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.

Authors' Addresses

Haoyu Song
Futurewei
USA

Email: haoyu.song@futurewei.com

Zhenbin Li
Huawei Technologies
China

Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
China

Email: pengshuping@huawei.com

James Guichard
Futurewei
USA

Email: james.n.guichard@futurewei.com

IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 24, 2021

X. Min
G. Mirsky
ZTE Corp.
L. Bo
China Telecom
January 20, 2021

Echo Request/Reply for Enabled In-situ OAM Capabilities
draft-xiao-ippm-ioam-conf-state-08

Abstract

This document describes an extension to the echo request/reply mechanisms used in IPv6, MPLS and SFC environments, which can be used within an IOAM domain, allowing the IOAM encapsulating node to acquire the enabled IOAM capabilities of each IOAM transit node and/or IOAM decapsulating node.

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 24, 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. Conventions	4
2.1. Requirements Language	4
2.2. Abbreviations	4
3. IOAM Capabilities Formats	5
3.1. IOAM Capabilities Query TLV in the Echo Request	5
3.2. IOAM Capabilities Response TLV in the Echo Reply	6
3.2.1. IOAM Pre-allocated Tracing Capabilities sub-TLV	7
3.2.2. IOAM Incremental Tracing Capabilities sub-TLV	8
3.2.3. IOAM Proof of Transit Capabilities sub-TLV	9
3.2.4. IOAM Edge-to-Edge Capabilities sub-TLV	10
3.2.5. IOAM DEX Capabilities sub-TLV	11
3.2.6. IOAM End-of-Domain sub-TLV	12
4. Operational Guide	13
5. Security Considerations	13
6. IANA Considerations	14
6.1. IOAM SoR Capability Registry	14
6.2. IOAM TSF+TSL Capability Registry	15
7. Acknowledgements	15
8. References	16
8.1. Normative References	16
8.2. Informative References	16
Authors' Addresses	17

1. Introduction

The Data Fields for In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data] defines data fields that record OAM information within the packet while the packet traverses a particular network domain, which is called an IOAM domain. IOAM can be used to complement OAM mechanisms based on, e.g., ICMP or other types of probe packets, and IOAM mechanisms can be leveraged where mechanisms using, e.g., ICMP do not apply or do not offer the desired results.

As specified in [I-D.ietf-ippm-ioam-data], within the IOAM-domain, the IOAM data may be updated by network nodes that the packet traverses. The device which adds an IOAM data container to the packet to capture IOAM data is called the "IOAM encapsulating node". In contrast, the device which removes the IOAM data container is referred to as the "IOAM decapsulating node". Nodes within the domain that are aware of IOAM data and read and/or write or process the IOAM data are called "IOAM transit nodes". Both the IOAM

encapsulating node and the decapsulating node are referred to as domain edge devices, which can be hosts or network devices.

In order to add the correct IOAM data container to the packet, the IOAM encapsulating node needs to know the enabled IOAM capabilities at the IOAM transit nodes and/or the IOAM decapsulating node as a whole, e.g., how many IOAM transit nodes will add tracing data, and what kinds of data fields will be added. A centralized controller could be used in some IOAM deployments. The IOAM encapsulating node can acquire these IOAM capabilities info from the centralized controller, through, e.g., Netconf/Yang, PCEP, or BGP. In the IOAM deployment scenario where there is no centralized controller, Netconf/Yang or IGP may be used for the IOAM encapsulating node to acquire these IOAM capabilities info, however, whether Netconf/Yang or IGP has some limitations as follows.

- o When Netconf/Yang is used in this scenario, each IOAM encapsulating node (including the host when it takes the role of an IOAM encapsulating node) needs to implement a Netconf Client, each IOAM transit node and IOAM decapsulating node (including the host when it takes the role of an IOAM decapsulating node) needs to implement a Netconf Server, the complexity can be an issue. Furthermore, each IOAM encapsulating node needs to establish Netconf Connection with each IOAM transit node and IOAM decapsulating node, the scalability can be an issue.
- o When IGP is used in this scenario, the IGP domain and an IOAM domain don't always have the same coverage. For example, when the IOAM encapsulating node or the IOAM decapsulating node is a host, the availability can be an issue. Furthermore, it might be too challenging to reflect IOAM capabilities at the IOAM transit node and/or the IOAM decapsulating node if these are controlled by a local policy depending on the identity of the IOAM encapsulating node.

This document describes an extension to the echo request/reply mechanisms used in IPv6, MPLS and SFC environments, which can be used within an IOAM domain where no Centralized Controller exists, allowing the IOAM encapsulating node to acquire the enabled IOAM capabilities of each IOAM transit node and/or IOAM decapsulating node.

The following documents contain references to the echo request/reply mechanisms used in IPv6, MPLS and SFC environments:

- o [RFC4443] ("Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification"), [RFC4884]

- ("Extended ICMP to Support Multi-Part Messages") and [RFC8335] ("PROBE: A Utility for Probing Interfaces")
- o [RFC8029] ("Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures")
 - o [I-D.ietf-sfc-multi-layer-oam] ("Active OAM for Service Function Chains in Networks")

This feature described in this document is assumedly applied to explicit path (strict or loose), because the precondition for this feature to work is that the echo request reaches each IOAM transit node as live traffic traverses.

2. Conventions

2.1. Requirements Language

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

2.2. Abbreviations

BGP: Border Gateway Protocol

E2E: Edge to Edge

ICMP: Internet Control Message Protocol

IGP: Interior Gateway Protocol

IOAM: In-situ Operations, Administration, and Maintenance

LSP: Label Switched Path

MPLS: Multi-Protocol Label Switching

MBZ: Must Be Zero

MTU: Maximum Transmission Unit

NTP: Network Time Protocol

OAM: Operations, Administration, and Maintenance

PCEP: Path Computation Element (PCE) Communication Protocol

POSIX: Portable Operating System Interface

POT: Proof of Transit

PTP: Precision Time Protocol

SFC: Service Function Chain

TTL: Time to Live

3. IOAM Capabilities Formats

3.1. IOAM Capabilities Query TLV in the Echo Request

In echo request IOAM Capabilities Query uses TLV (Type-Length-Value tuple) which have the following format:

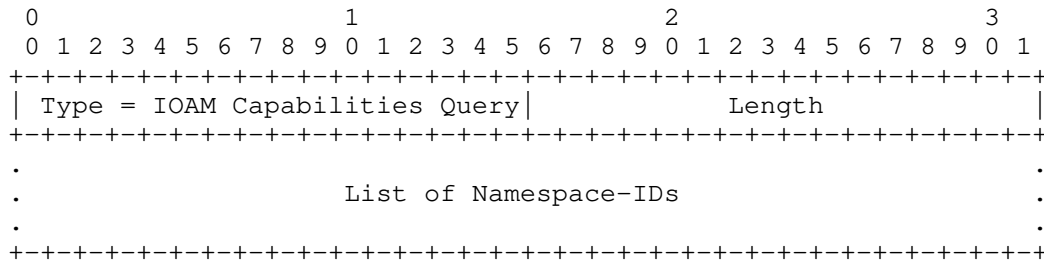


Figure 1: IOAM Capabilities Query TLV in the Echo Request

When this TLV is present in the echo request sent by an IOAM encapsulating node, it means that the IOAM encapsulating node requests the receiving node to reply with its enabled IOAM capabilities. If there is no IOAM capability to be reported by the receiving node, then this TLV SHOULD be ignored by the receiving node, which means the receiving node SHOULD send echo reply without IOAM capabilities or no echo reply, in the light of whether the echo request includes other TLV than IOAM Capabilities Query TLV. List of Namespace-IDs MAY be included in this TLV of the echo request. In that case, the IOAM encapsulating node requests only the IOAM capabilities that match one of the Namespace-IDs. The Namespace-ID has the same definition as what's specified in [I-D.ietf-ippm-ioam-data].

Type is set to the value that identifies it as an IOAM Capabilities Query TLV.

Length is the length of the TLV's Value field in octets, including a List of Namespace-IDs.

Value field of this TLV is zero-padded to align to a 4-octet boundary.

3.2. IOAM Capabilities Response TLV in the Echo Reply

In echo reply IOAM Capabilities Response uses TLV which have the following format:

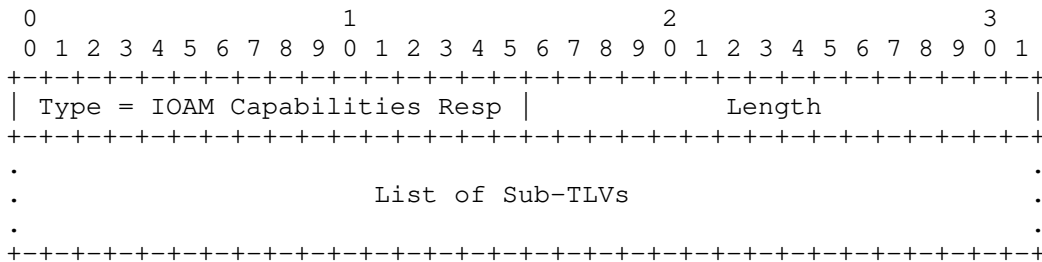


Figure 2: IOAM Capabilities Response TLV in the Echo Reply

When this TLV is present in the echo reply sent by an IOAM transit node and/or an IOAM decapsulating node, it means that the IOAM function is enabled at this node, and this TLV contains the enabled IOAM capabilities of the sender. A list of Sub-TLVs which contains the IOAM capabilities SHOULD be included in this TLV of the echo reply. Note that the IOAM encapsulating node or the IOAM decapsulating node can also be an IOAM transit node.

Type is set to the value that identifies it as an IOAM Capabilities Response TLV.

Length is the length of the TLV's Value field in octets, including a List of Sub-TLVs.

Value field of this TLV or any Sub-TLV is zero-padded to align to a 4-octet boundary. Based on the data fields for IOAM, specified in [I-D.ietf-ippm-ioam-data] and [I-D.ietf-ippm-ioam-direct-export], six kinds of Sub-TLVs are defined in this document. The same type of the sub-TLV MAY be in the IOAM Capabilities Response TLV more than once only if with a different Namespace-ID. Note that the IOAM encapsulating node may receive both IOAM Pre-allocated Tracing Capabilities sub-TLV and IOAM Incremental Tracing Capabilities sub-TLV in the process of traceroute, which means both pre-allocated tracing node and incremental tracing node are on the same path, or some node supports both pre-allocated tracing and incremental

tracing, the behavior of the IOAM encapsulating node in this scenario is outside the scope of this document.

3.2.1. IOAM Pre-allocated Tracing Capabilities sub-TLV

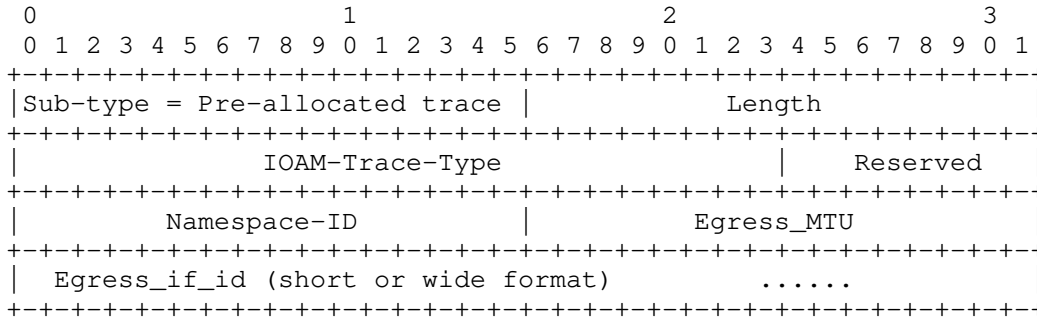


Figure 3: IOAM Pre-allocated Tracing Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM transit node and IOAM tracing function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM Pre-allocated Tracing Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets. If Egress_if_id is in the short format, which is 16 bits long, it MUST be set to 10. If Egress_if_id is in the wide format, which is 32 bits long, it MUST be set to 12.

IOAM-Trace-Type field has the same definition as what's specified in section 5.4 of [I-D.ietf-ippm-ioam-data].

Reserved field is reserved for future use and MUST be set to zero.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

Egress_MTU field has 16 bits and specifies the MTU of the egress direction out of which the sending node would forward the received echo request, it should be the MTU of the egress interface or the MTU between the sending node and the downstream IOAM transit node.

Egress_if_id field has 16 bits (in short format) or 32 bits (in wide format) and specifies the identifier of the egress interface out of which the sending node would forward the received echo request.

3.2.2. IOAM Incremental Tracing Capabilities sub-TLV

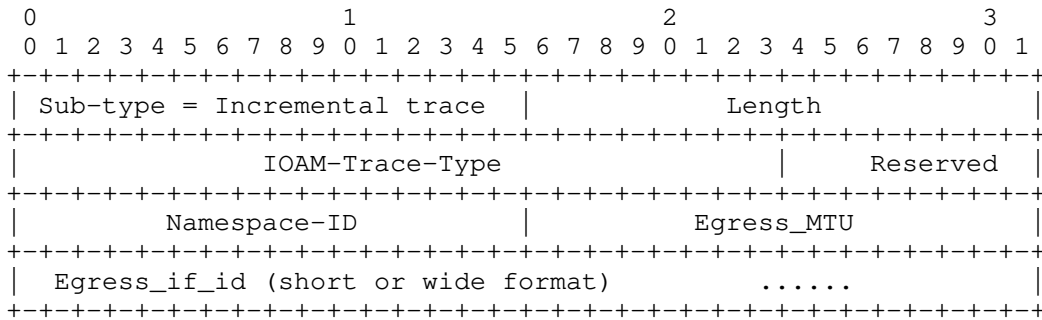


Figure 4: IOAM Incremental Tracing Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM transit node and IOAM tracing function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM Incremental Tracing Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets. If Egress_if_id is in the short format, which is 16 bits long, it MUST be set to 10. If Egress_if_id is in the wide format, which is 32 bits long, it MUST be set to 12.

IOAM-Trace-Type field has the same definition as what's specified in section 5.4 of [I-D.ietf-ippm-ioam-data].

Reserved field is reserved for future use and MUST be set to zero.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

Egress_MTU field has 16 bits and specifies the MTU of the egress direction out of which the sending node would forward the received echo request, it should be the MTU of the egress interface or the MTU between the sending node and the downstream IOAM transit node.

Egress_if_id field has 16 bits (in short format) or 32 bits (in wide format) and specifies the identifier of the egress interface out of which the sending node would forward the received echo request.

3.2.3. IOAM Proof of Transit Capabilities sub-TLV

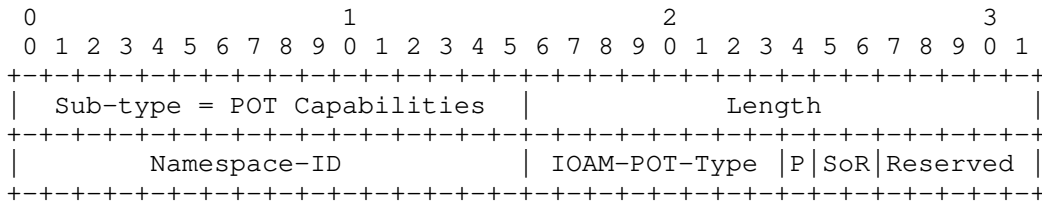


Figure 5: IOAM Proof of Transit Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM transit node and IOAM proof of transit function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM Proof of Transit Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST be set to 4.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

IOAM-POT-Type field and P bit have the same definition as what's specified in section 5.5 of [I-D.ietf-ippm-ioam-data]. If the IOAM encapsulating node receives IOAM-POT-Type and/or P bit values from an IOAM transit node that are different from its own, then the IOAM encapsulating node MAY choose to abandon the proof of transit function or to select one kind of IOAM-POT-Type and P bit, it's based on the policy applied to the IOAM encapsulating node.

SoR field has two bits, which means the size of "Random" and "Cumulative" data that are specified in section 5.5 of [I-D.ietf-ippm-ioam-data]. This document defines SoR as follow:

0b00 means 64-bit "Random" and 64-bit "Cumulative" data.

0b01~0b11: Reserved for future standardization

Reserved field is reserved for future use and MUST be set to zero.

3.2.4. IOAM Edge-to-Edge Capabilities sub-TLV

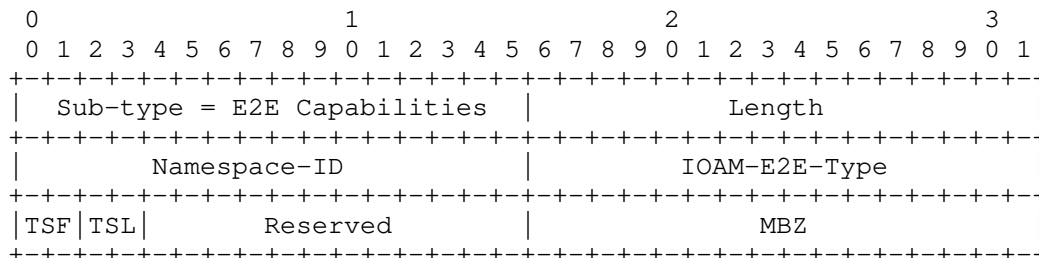


Figure 6: IOAM Edge-to-Edge Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM decapsulating node and IOAM edge-to-edge function is enabled at this IOAM decapsulating node. That is to say, if the IOAM encapsulating node receives this sub-TLV, the IOAM encapsulating node can determine that the node which sends this sub-TLV is an IOAM decapsulating node.

Sub-type is set to the value that identifies it as an IOAM Edge-to-Edge Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST be set to 8.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

IOAM-E2E-Type field has the same definition as what's specified in section 5.6 of [I-D.ietf-ippm-ioam-data].

TSF field specifies the timestamp format used by the sending node. This document defines TSF as follow:

- 0b00: PTP timestamp format
- 0b01: NTP timestamp format
- 0b10: POSIX timestamp format
- 0b11: Reserved for future standardization

TSL field specifies the timestamp length used by the sending node. This document defines TSL as follow.

When the TSF field is set to 0b00, which indicates the PTP timestamp format, the values of the TSL field are interpreted as follows:

0b00: 64-bit PTPv1 timestamp as defined in IEEE1588-2008 [IEEE1588v2]

0b01: 80-bit PTPv2 timestamp as defined in IEEE1588-2008 [IEEE1588v2]

0b10~0b11: Reserved for future standardization

When the TSF field is set to 0b01, which indicates the NTP timestamp format, the values of the TSL field are interpreted as follows:

0b00: 32-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b01: 64-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b10: 128-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b11: Reserved for future standardization

When the TSF field is set to 0b10 or 0b11, the TSL field would be ignored.

Reserved field is reserved for future use and MUST be set to zero.

3.2.5. IOAM DEX Capabilities sub-TLV

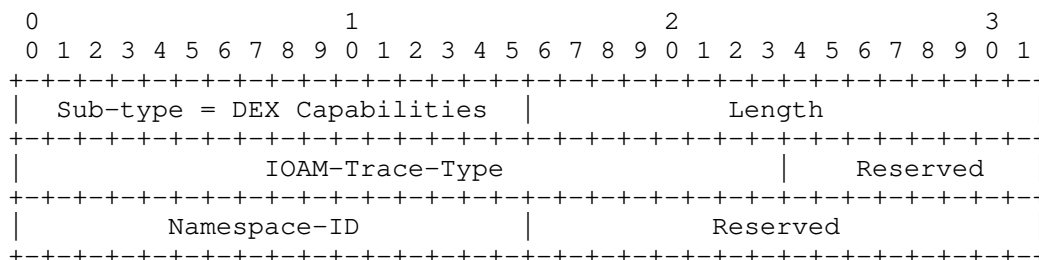


Figure 7: IOAM DEX Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM transit node and the IOAM DEX function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM DEX Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST be set to 8.

IOAM-Trace-Type field has the same definition as what's specified in section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Namespace-ID field has the same definition as what's specified in section 3.2 of [I-D.ietf-ippm-ioam-direct-export], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

Reserved field is reserved for future use and MUST be set to zero.

3.2.6. IOAM End-of-Domain sub-TLV

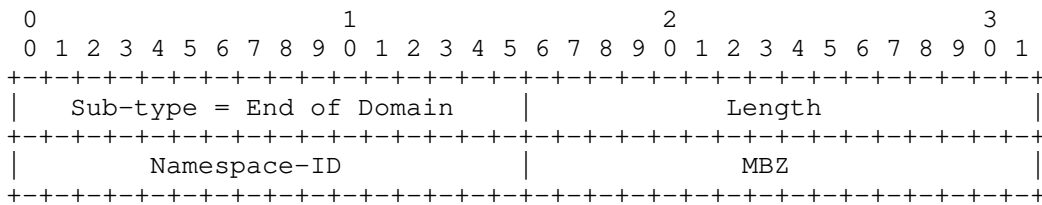


Figure 8: IOAM End of Domain Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM decapsulating node. That is to say, if the IOAM encapsulating node receives this sub-TLV, the IOAM encapsulating node can determine that the node which sends this sub-TLV is an IOAM decapsulating node. When the IOAM Edge-to-Edge Capabilities sub-TLV is present in the IOAM Capabilities Response TLV sent by the IOAM decapsulating node, the IOAM End-of-Domain sub-TLV doesn't need to be present in the same IOAM Capabilities Response TLV, otherwise the End-of-Domain sub-TLV MUST be present in the IOAM Capabilities Response TLV sent by the IOAM decapsulating node. Both the IOAM Edge-to-Edge Capabilities sub-TLV and the IOAM End-of-Domain sub-TLV can be used to indicate that the sending node is an IOAM decapsulating node. It's recommended to include only the IOAM Edge-to-Edge Capabilities sub-TLV if IOAM edge-to-edge function is enabled at this IOAM decapsulating node.

Sub-type is set to the value that identifies it as an IOAM End of Domain sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST be set to 4.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

4. Operational Guide

Once the IOAM encapsulating node is triggered to acquire the enabled IOAM capabilities of each IOAM transit node and/or IOAM decapsulating node, the IOAM encapsulating node will send echo requests that include the IOAM Capabilities Query TLV. First with TTL equal to 1 to reach the nearest node, which may be an IOAM transit node or not. Then with TTL equal to 2 to reach the second nearest node, which also may be an IOAM transit node or not. And further, increasing by 1 the TTL every time the IOAM encapsulating node sends a new echo request, until the IOAM encapsulating node receives an echo reply sent by the IOAM decapsulating node, which should contain the IOAM Capabilities Response TLV including the IOAM Edge-to-Edge Capabilities sub-TLV or the IOAM End-of-Domain sub-TLV. Alternatively, if the IOAM encapsulating node knows exactly all the IOAM transit nodes and/or IOAM decapsulating node beforehand, once the IOAM encapsulating node is triggered to acquire the enabled IOAM capabilities, it can send an echo request to each IOAM transit node and/or IOAM decapsulating node directly, without TTL expiration.

The IOAM encapsulating node may be triggered by the device administrator, the network management system, the network controller, or even the live user traffic. The specific triggering mechanisms are outside the scope of this document.

Each IOAM transit node and/or IOAM decapsulating node that receives an echo request containing the IOAM Capabilities Query TLV will send an echo reply to the IOAM encapsulating node, and within the echo reply, there should be an IOAM Capabilities Response TLV containing one or more sub-TLVs. The IOAM Capabilities Query TLV contained in the echo request would be ignored by the receiving node that is unaware of IOAM.

5. Security Considerations

Queries and responses about the state of an IOAM domain should be processed only from a trusted source. An unauthorized query MUST be discarded by an implementation that supports this specification. Similarly, unsolicited echo response with the IOAM Capabilities TLV MUST be discarded. Authentication of echo request/reply that

includes the IOAM Capabilities TLV is one of methods of the integrity protection. Implementations could also provide a means of filtering based on the source address of the received echo request/reply. The integrity protection for IOAM capabilities information collection can also be achieved using mechanisms in the underlay data plane. For example, if the underlay is an IPv6 network, IP Authentication Header [RFC4302] or IP Encapsulating Security Payload Header [RFC4303] can be used to provide integrity protection.

Information about the state of the IOAM domain collected in the IOAM Capabilities TLV is confidential. An implementation can use secure transport to provide privacy protection. For example, if the underlay is an IPv6 network, confidentiality can be achieved using the IP Encapsulating Security Payload Header [RFC4303].

6. IANA Considerations

This document requests the following IANA Actions.

IANA is requested to create a registry group named "In-Situ OAM (IOAM) Capabilities Parameters".

This group will include the following registries:

- o IOAM SoR Capability
- o IOAM TSF+TSL Capability

New registries in this group can be created via RFC Required process as per [RFC8126].

The subsequent sub-sections detail the registries herein contained.

Considering the TLVs/sub-TLVs defined in this document would be carried in different kinds of Echo Request/Reply message, such as ICMPv6 or LSP Ping, it is intended that the registries for Type and sub-Type would be requested in subsequent documents.

6.1. IOAM SoR Capability Registry

This registry defines 4 code points for the IOAM SoR Capability field for identifying the size of "Random" and "Cumulative" data as explained in section 5.5 of [I-D.ietf-ippm-ioam-data]. The following code points are defined in this draft:

SoR ----	Description -----
0b00	64-bit "Random" and 64-bit "Cumulative" data

0b01 - 0b11 are available for assignment via RFC Required process as per [RFC8126].

6.2. IOAM TSF+TSL Capability Registry

This registry defines 3 code points for the IOAM TSF Capability field for identifying the timestamp format as explained in section 6 of [I-D.ietf-ippm-ioam-data].

- o When the code point for the IOAM TSF Capability field equals 0b00 which means PTP timestamp format, this registry defines 2 code points for the IOAM TSL Capability field for identifying the timestamp length.
- o When the code point for the IOAM TSF Capability field equals 0b01 which means NTP timestamp format, this registry defines 3 code points for the IOAM TSL Capability field for identifying the timestamp length.

The following code points are defined in this draft:

TSF ----	TSL ----	Description -----
0b00		PTP Timestamp Format
	0b00	64-bit PTPv1 timestamp
	0b01	80-bit PTPv2 timestamp
0b01		NTP Timestamp Format
	0b00	32-bit NTP timestamp
	0b01	64-bit NTP timestamp
	0b10	128-bit NTP timestamp
0b10		POSIX Timestamp Format

Unassigned code points of TSF+TSL are available for assignment via RFC Required process as per [RFC8126].

7. Acknowledgements

The authors would like to acknowledge Tianran Zhou, Dhruv Dhody, Frank Brockners and Cheng Li for their careful review and helpful comments.

The authors appreciate the f2f discussion with Frank Brockners on this document.

The authors would like to acknowledge Tommy Pauly and Ian Swett for their good suggestion and guidance.

8. References

8.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-02 (work in progress), November 2020.
- [IEEE1588v2]
Institute of Electrical and Electronics Engineers, "IEEE Std 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, 2008, <<http://standards.ieee.org/findstds/standard/1588-2008.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>>.

8.2. Informative References

- [I-D.ietf-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active OAM for Service Function Chains in Networks", draft-ietf-sfc-multi-layer-oam-07 (work in progress), December 2020.

- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <<https://www.rfc-editor.org/info/rfc4884>>.
- [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>>.
- [RFC8335] Bonica, R., Thomas, R., Linkova, J., Lenart, C., and M. Boucadair, "PROBE: A Utility for Probing Interfaces", RFC 8335, DOI 10.17487/RFC8335, February 2018, <<https://www.rfc-editor.org/info/rfc8335>>.

Authors' Addresses

Xiao Min
ZTE Corp.
Nanjing
China

Phone: +86 25 88013062
Email: xiao.min2@zte.com.cn

Greg Mirsky
ZTE Corp.
USA

Email: gregimirsky@gmail.com

Lei Bo
China Telecom
Beijing
China

Phone: +86 10 50902903
Email: leibo@chinatelecom.cn

IPPM
Internet-Draft
Intended status: Standards Track
Expires: July 22, 2021

T. Zhou, Ed.
G. Fioccola
Huawei
S. Lee
LG U+
M. Cociglio
Telecom Italia
W. Li
Huawei
January 18, 2021

Enhanced Alternate Marking Method
draft-zhou-ippm-enhanced-alternate-marking-06

Abstract

This document extends the IPv6 alternate marking option to provide the enhanced capabilities.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on July 22, 2021.

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. Data Fields Format	2
3. Enhanced Alternate Marking capabilities	3
4. Security Considerations	4
5. IANA Considerations	4
6. References	4
6.1. Normative References	4
6.2. Informative References	4
Authors' Addresses	4

1. Introduction

The Alternate Marking [RFC8321] and Multipoint Alternate Marking [I-D.ietf-ippm-multipoint-alt-mark] define the Alternate Marking technique that is an hybrid performance measurement method, per [RFC7799] classification of measurement methods. This method is based on marking consecutive batches of packets and it can be used to measure packet loss, latency, and jitter on live traffic.

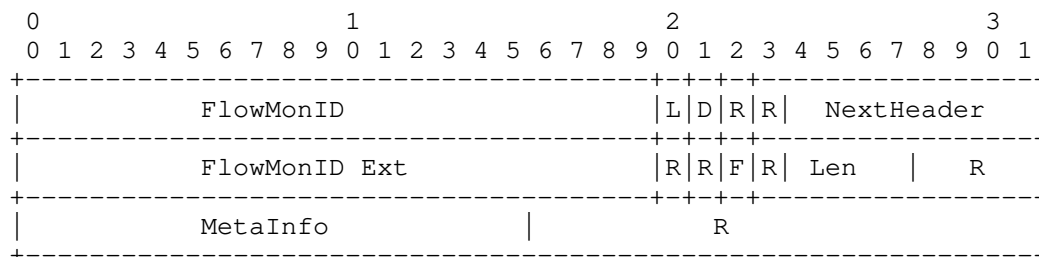
AltMark Option [I-D.ietf-6man-ipv6-alt-mark] applies the Alternate Marking Method for IPv6 protocol, and defines Extension Header Option to encode Alternate Marking Method for both Hop-by-Hop Options Header and Destination Options Header.

While the AltMark Option implement the basic alternate marking method, this document defines the extended data fields for the AltMark Option and provides the enhanced capabilities.

It is worth mentioning that the enhanced capabilities are intended for further use and are optional.

2. Data Fields Format

The following figure shows the data fields format for enhanced alternate marking. This data is expected to be encapsulated to specific transports.



where:

- o FlowMonID - Flow Monitoring Identification is the same as defined in AltMark Option [I-D.ietf-6man-ipv6-alt-mark].
- o L and D - Loss Flag and Delay Flag are the same as defined in AltMark Option [I-D.ietf-6man-ipv6-alt-mark].
- o NextHeader - Identify whether to carry the extended data fields.
- o FlowMonID Ext - 20 bits unsigned integer. This used to extend the FlowMonID to reduce the conflict when random allocation is applied
- o R - Reserved for further use. This bit MUST be set to zero.
- o F - Flow direction identification. F = 1, indicate the flow direction is forward.
- o Len - Length. It indicates the length of extension headers.
- o MetaInfo - A 16 bits Bitmap to indicate more meta data attached for the enhanced function.

3. Enhanced Alternate Marking capabilities

The extended data fields presented in the previous section can be used for several uses. Some possible applications can be:

1. shortest marking periods of single marking method for thicker packet loss measurements.
2. more dense delay measurements than double marking method (down to each packet).
3. increase the entropy of flow monitoring identifier by extending the size of FlowMonID.

4. and so on.

4. Security Considerations

TBD

5. IANA Considerations

This document has no request to IANA.

6. References

6.1. Normative References

- [I-D.ietf-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto,
"Multipoint Alternate Marking method for passive and
hybrid performance monitoring", draft-ietf-ippm-
multipoint-alt-mark-09 (work in progress), March 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with
Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799,
May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli,
L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi,
"Alternate-Marking Method for Passive and Hybrid
Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321,
January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

6.2. Informative References

- [I-D.ietf-6man-ipv6-alt-mark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R.
Pang, "IPv6 Application of the Alternate Marking Method",
draft-ietf-6man-ipv6-alt-mark-02 (work in progress),
October 2020.

Authors' Addresses

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Shinyoung Lee
LG U+
71, Magokjungang 8-ro, Gangseo-gu
Seoul
Republic of Korea

Email: leesy@lguplus.co.kr

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

Email: mauro.cociglio@telecomitalia.it

Weidong Li
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: poly.li@huawei.com