

MPLS Working Group
Internet-Draft
Updates: 3032, 7274 (if approved)
Intended status: Informational
Expires: August 30, 2019

L. Andersson
Bronze Dragon Consulting
K. Kompella
Juniper Networks
A. Farrel
Old Dog Consulting
February 26, 2019

Special Purpose Label terminology
draft-andersson-mpls-spl-terminology-01

Abstract

This document discusses and recommends a terminology that may be used when MPLS Special Purpose Labels (SPL) are specified and documented.

Note: The rest of the text in this section is not really part of the abstract even though the text is placed here. It is working notes.

Note: At least at the moment it is not the intention to take this document to an RFC, but it might be polled to become a wg document to see if the MPLS working group agree on the proposed terminology.

Note: The changes we propose are minor, but we might have to progress the document to RFC since there is a proposed change to the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry.

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 30, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Background	3
2.1. GMPLS Special Purpose Labels	3
3. Terminology and Abbreviations	3
4. Security Considerations	5
5. IANA Considerations	5
6. Acknowledgements	5
7. References	5
7.1. Normative References	5
7.2. Informative References	6
Authors' Addresses	6

1. Introduction

RFC 7274 [RFC7274] made some changes to the terminology used for MPLS Special Purpose Labels, but did not define consistent terminology.

One thing that RFC 7274 did was to deprecate use of the term "reserved labels" when describing a range of labels allocated from a registry maintained by IANA. The term "Reserved" in such a registry means "set aside, not to be used", but that range of labels was available for allocation according to the policies set out in the registry. The name "Special Purpose Labels" was introduced in RFC 7274 in place of the previous term, and the abbreviation SPL was recommended.

At the time of posting this Internet-Draft, the IETF is in the process of allocating the very first SPLs from the Extended SPL range [I-D.ietf-mpls-sfc]. This document discusses and recommends terminology and abbreviations to be used when talking about and documenting Special Purpose Labels.

2. Background

Two sets of SPLs are defined for use in MPLS:

The range of SPLs 0-15 is specified in RFC 3032 [RFC3032].

The range of SPLs 0-1048575 is specified in RFC 7274 [RFC7274].

- * the values 0-15 has been reserved never to be allocated
- * the values 15-239 are available for allocation
- * the values 240-255 are for experimental use
- * the values 256-1048575 are currently not available for allocation, and a standard tracks RFC will be needed to make the entire range or part of it available for allocation

2.1. GMPLS Special Purpose Labels

Note that IANA maintains a registry called "Special Purpose Generalized Label Values". Labels in that registry have special meaning when present in certain signalling objects, are 32 bits long, and are not to be confused with MPLS forwarding plane labels. This document does not make any changes to the registry or how labels from that registry are described.

3. Terminology and Abbreviations

IANA maintains a name space for 'Special-Purpose Multiprotocol Label Switching (MPLS) Label Values' code points [SPL-NAME-SPACE]. Within this name space there are two registries. One is called the 'Special-Purpose MPLS Label Values' registry [bSPL]. The other is called 'Extended Special-Purpose MPLS Label Values' registry [eSPL].

The difference in the name of the name space and the first registry is only that the MPLS abbreviation is expanded. This document changes the name of the first registry to 'Base Special-Purpose MPLS Label Values', but leaves the name of the latter registry unchanged as 'Extended Special-Purpose MPLS Label Values'.

The following conventions will be used in specifications and when talking about SPLs

- o Collectively, the two ranges are known as Special Purpose Labels (SPL).

- o The special purpose labels from the lower range will be called Base Special Purpose Labels (bSPL).
- o The special purpose labels from the higher range will be called Extended Special Purpose Labels (eSPL).
- o The combination of the Extension Label (XL) (value 15 which is an bSPL, but that is also called xSPL) and an eSPL is called a Composite Special Purpose Label (cSPL).

This results in a label stacks such as the illustrative examples shown in Figure 1 and Figure 2.

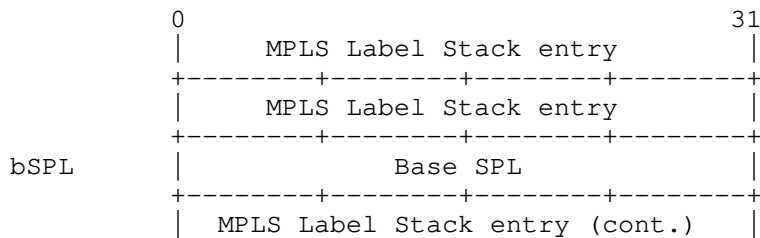


Figure 1: Example of Label Stack

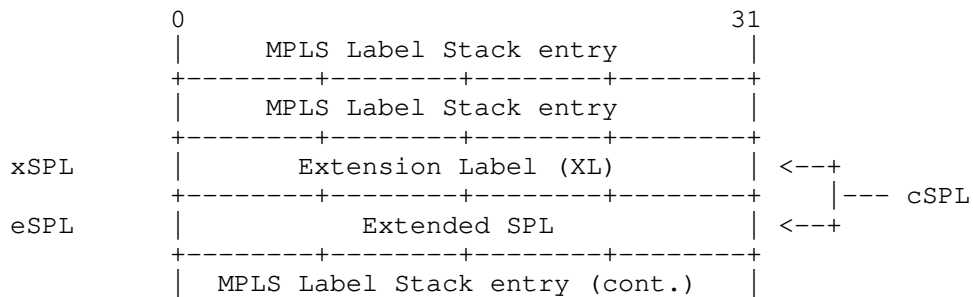


Figure 2: Example of Label Stack

4. Security Considerations

This document is entirely about terminology for SPLs and does not effect the forwarding in the MPLS data plane, nor does it have any effect on how LSPs are established by an MPLS control plane or by a centralized controller. The document describes a terminology to be used when describing and specifying the use of SPLs.

This document does not aim to describe existing implementations of SPLs or the potential vulnerabilities of SPLs.

5. IANA Considerations

We request that the name of the IANA registry that today is called "Special-Purpose MPLS Label Values" is changed to "Base Special-Purpose MPLS Label Values".

6. Acknowledgements

The authors of this document would like to thank Stewart Bryant for careful review and constructive suggestions.

-

7. References

7.1. Normative References

- [bSPL] "Special-Purpose MPLS Label Values",
<[https://www.iana.org/assignments/mpls-label-values/
mpls-label-values.xhtml#special-purpose/](https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xhtml#special-purpose/)>.
- [eSPL] "Extended Special-Purpose MPLS Label Values",
<[https://www.iana.org/assignments/mpls-label-values/
mpls-label-values.xhtml#extended/](https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xhtml#extended/)>.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y.,
Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack
Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001,
<<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating
and Retiring Special-Purpose MPLS Labels", RFC 7274,
DOI 10.17487/RFC7274, June 2014,
<<https://www.rfc-editor.org/info/rfc7274>>.

[SPL-NAME-SPACE]

"Special-Purpose Multiprotocol Label Switching (MPLS)
Label Values", <[https://www.iana.org/assignments/
mpls-label-values/mpls-label-values.xhtml](https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xhtml)>.

7.2. Informative References

[I-D.ietf-mpls-sfc]

Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based
Forwarding Plane for Service Function Chaining", draft-
ietf-mpls-sfc-05 (work in progress), February 2019.

Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu

Kireeti Kompella
Juniper Networks

Email: kireeti@juniper.net

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2019

Z. Chen
L. Qiang
Huawei
March 9, 2019

MPLS-LSP Data Plane for Cyclic Queuing and Forwarding
draft-chen-mpls-cqf-lsp-dp-00

Abstract

Large-scale Deterministic Network (LDN) [ldn] aims to achieve bounded latency forwarding on layer-3 networks that contain long-distance links, large number of nodes and flows. LDN requires a data plane mechanism to indicate different forwarding cycles in the upstream node. This document proposes to use multiple MPLS labels to indicate this kind of information, for MPLS-LSP data plane.

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 September 10, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction (LDN Background)	2
2. MPLS-LSP Data Plane for CQF	3
3. IANA Considerations	4
4. Security Considerations	4
5. Acknowledgements	4
6. Normative References	4
Authors' Addresses	5

1. Introduction (LDN Background)

Large-scale Deterministic Network (LDN) [ldn] aims to achieve bounded latency forwarding on layer-3 networks that contain long-distance links, large number of nodes and flows. Figure 1 illustrates the basic mechanism of LDN, where an upstream Node A and a downstream Node B are considered. Each interface of a LDN router has three cyclic scheduled queues, i.e., at any given time (or cycle), one of the queues is sending packets and the others are receiving.

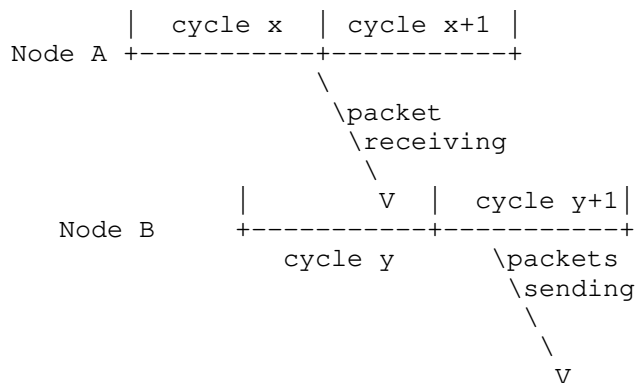


Figure 1

In order to achieve end-to-end bounded latency, LDN requires that all packets sent from the upstream router in a specific cycle MUST be sent by the downstream router within another (one) specific cycle.

For example, as shown in Figure 1, the packets sent by Node A within cycle x MUST be put into single receiving queue in Node B, and then be sent out within cycle $y+1$. The mapping relationship between x and $y+1$ could be configured by a centralized controller, or be self-learned by each peer of neighbors at the data plane.

Therefore, LDN requires a data plane mechanism to indicate which upstream node's cycle a packet belongs to, so that the downstream node could use this indication to put the packet into the right receiving queue. This document proposes to use multiple MPLS labels to indicate this kind of information, for MPLS-LSP data plane.

2. MPLS-LSP Data Plane for CQF

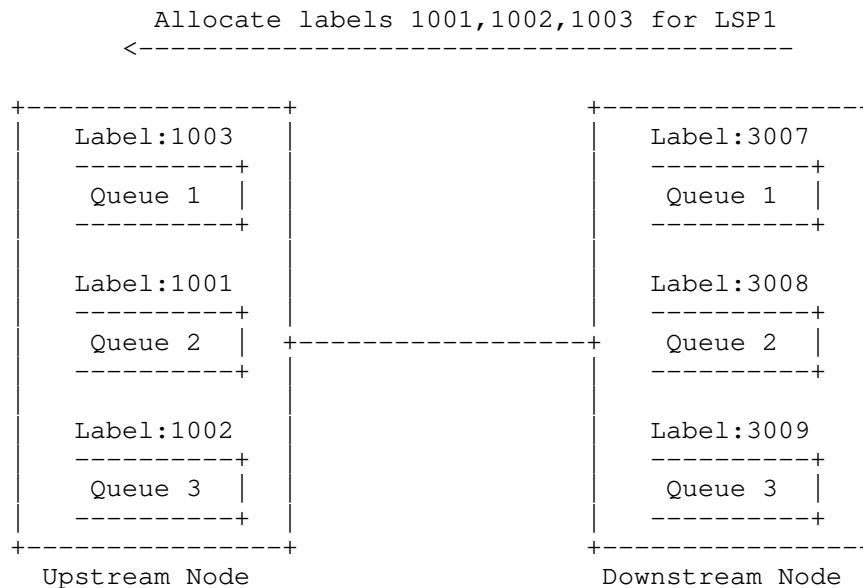


Figure 2

Figure 2 shows the overall mechanism of MPLS-LSP data plane for CQF, where the downstream node allocates three different MPLS labels (i.e., 1000, 1002, and 1003) for LSP1, and advertises this information to the upstream node by using signaling protocols such as RSVP-TE. Each of these labels is associated with a specific queue in the upstream node.

Assume that packets sent from the upstream node's queue 1, queue 2, and queue 3 SHOULD be put into the downstream node's queue 3, queue 1, and queue 2, respectively. Note that how to establish such mapping

relationships is out of the scope of this document. Based on these mapping relationships, the downstream node SHOULD install its FIB like the one shown in Figure 3.

Downstream Node's FIB			
In-label	OutIF	OutQ	Out-label
1003	3	3	3009
1001	3	1	3007
1002	3	2	3008

Figure 3

Therefore, the packets sent from the upstream node's queue 1 will be put into the downstream node's queue 3, the packets sent from the upstream node's queue 2 will be put into the downstream node's queue 1, and the packets sent from the upstream node's queue 3 will be put into the downstream node's queue 2. In this way, end-to-end latency could be bounded, as per [ldn].

3. IANA Considerations

TBD.

4. Security Considerations

TBD.

5. Acknowledgements

TBD.

6. Normative References

- [ldn] Qiang, L., Liu, B., Eckert, T., and L. Geng, "Large-Scale Deterministic Network", March 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

Authors' Addresses

Zhe Chen
Huawei

Email: chenzhe17@huawei.com

Li Qiang
Huawei

Email: qiangli3@huawei.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 12, 2021

W. Cheng
China Mobile
X. Min
ZTE
T. Zhou
Huawei
X. Dong
FiberHome
Y. Peleg
Broadcom
September 8, 2020

Encapsulation For MPLS Performance Measurement with Alternate Marking
Method
draft-cheng-mpls-inband-pm-encapsulation-04

Abstract

This document defines the encapsulation for MPLS performance measurement with alternate marking method, which performs flow-based packet loss, delay, and jitter measurements on live traffic.

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 12, 2021.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
1.1. Conventions Used in This Document	3
1.1.1. Abbreviations	3
1.1.2. Requirements Language	4
2. Flow-based PM Encapsulation in MPLS	4
2.1. Examples for Applying Flow-ID Label in a label stack . .	5
3. Procedures of Encapsulation, Look-up and Decapsulation . . .	8
4. Procedures of Flow-ID allocation	9
5. FLC and FRLD Considerations	10
6. Security Considerations	10
7. IANA Considerations	11
8. Acknowledgements	11
9. References	11
9.1. Normative References	11
9.2. Informative References	12
Authors' Addresses	13

1. Introduction

[RFC8321] describes a passive performance measurement method, which can be used to measure packet loss, delay, and jitter on live traffic. Since this method is based on marking consecutive batches of packets, the method is often referred to as Alternate Marking Method.

[RFC8372] discusses the desired capabilities for MPLS flow identification, in order to perform a better in-band performance monitoring of user data packets. Synonymous Flow Label (SFL), which is introduced in [I-D.ietf-mpls-sfl-framework], is identified as a method of accomplishing MPLS flow identification. This document employs a method, other than SFL, to accomplish MPLS flow identification. The method described in this document is simple and flexible, furthermore, it complies with the current MPLS forwarding paradigm.

On one hand, the method described in this document is complementary to the SFL method [I-D.ietf-mpls-sfl-framework] [I-D.bryant-mpls-sfl-control], the former targets at hop-by-hop performance measurement, and the latter targets at end-to-end

performance measurement, furthermore, the former supports the application scenario where Flow-ID is applied to MPLS LSP and MPLS VPN synchronously, and the latter doesn't support this kind of application scenario. On the other hand, the method described in this document is complementary to the In-situ OAM method [I-D.ietf-ippm-ioam-data] [I-D.ietf-ippm-ioam-direct-export], the former doesn't introduce any new header but the latter introduces a new In-situ OAM header, furthermore, the former allows the network nodes to report the refined data (e.g. calculated performance metrics) associated with a specified flow, nevertheless the latter requests the network nodes to report the data (e.g. ingress interface and egress interface) associated with a specified packet.

This document defines the encapsulation for MPLS performance measurement with alternate marking method, which performs flow-based packet loss, delay, and jitter measurements on live traffic.

1.1. Conventions Used in This Document

1.1.1. Abbreviations

ELC: Entropy Label Capability

ERLD: Entropy Readable Label Depth

FLC: Flow-ID Label Capability

FRLD: Flow-ID Readable Label Depth

LSP: Label Switched Path

MPLS: Multi-Protocol Label Switching

NMS: Network Management System

PM: Performance Measurement

PW: PseudoWire

SFL: Synonymous Flow Label

SID: Segment ID

SR: Segment Routing

TC: Traffic Class

TTL: Time to Live

VC: Virtual Channel

VPN: Virtual Private Network

1.1.2. Requirements Language

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

2. Flow-based PM Encapsulation in MPLS

Flow-based MPLS performance measurement encapsulation with alternate marking method has the following format:

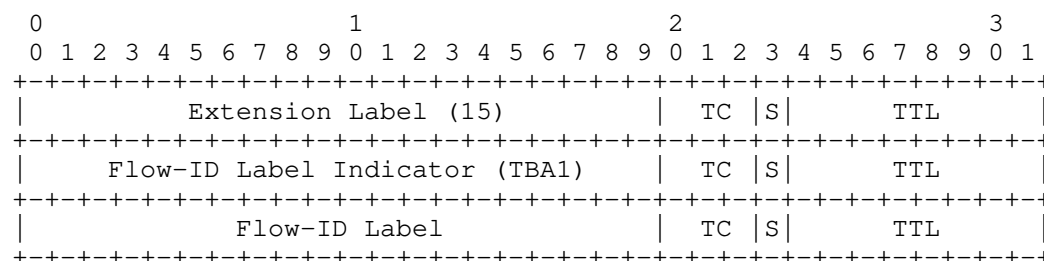


Figure 1: Flow-based PM Encapsulation in MPLS

Flow-ID Label Indicator is an Extended Special Purpose Label (eSPL), which is combined with the Extension Label (XL, value 15) to form a Composite Special Purpose Label (cSPL), as defined in [I-D.ietf-mpls-spl-terminology]. Flow-ID Label Indicator is defined in this document as value TBA1.

Analogous to Entropy Label Indicator [RFC6790], the TC and TTL for the Extension Label and the Flow-ID Label Indicator SHOULD follow the same field values of that label immediately preceding the Extension Label, otherwise, the TC and TTL for the Extension Label and the Flow-ID Label Indicator MAY be different values if it is known that the Extension Label will not be exposed as the top label at any point along the LSP. The S bit for the Extension Label and the Flow-ID Label Indicator MUST be zero.

Flow-ID Label is used as MPLS flow identification [RFC8372], its value should be unique within the administrative domain. Flow-ID

values can be allocated by an external NMS or a controller, based on measurement object instance such as LSP or PW. There is a one-to-one mapping between Flow-ID and flow. The specific method on how to allocate the Flow-ID values is described in Section 4.

Analogous to Entropy Label [RFC6790], the Flow-ID Label can be placed at either the bottom or the middle of the MPLS label stack, and the Flow-ID Label MAY appear multiple times in a label stack. Section 2.1 of this document provides several examples to illustrate how to apply Flow-ID Label in a label stack. Again analogous to Entropy Label, the TTL for the Flow-ID Label MUST be zero to ensure that it is not used inadvertently for forwarding, the TC for the Flow-ID Label may be any value, the S bit for the Flow-ID Label depends on whether or not there are more labels in the label stack.

Besides flow identification, a color-marking field is also necessary for alternate marking method. To achieve the purpose of coloring the MPLS traffic, the current practice when writing this document is to reuse the Flow-ID Label's TC, i.e., using TC's highest order two bits (called double-marking methodology [RFC8321]) as color-marking bits. Alternatively, allocating multiple Flow-ID Labels to the same flow may be used for the purpose of alternate marking.

2.1. Examples for Applying Flow-ID Label in a label stack

Three examples on different layout of Flow-ID Label (4 octets) are illustrated as follows:

- (1) Layout of Flow-ID Label when applied to MPLS LSP.

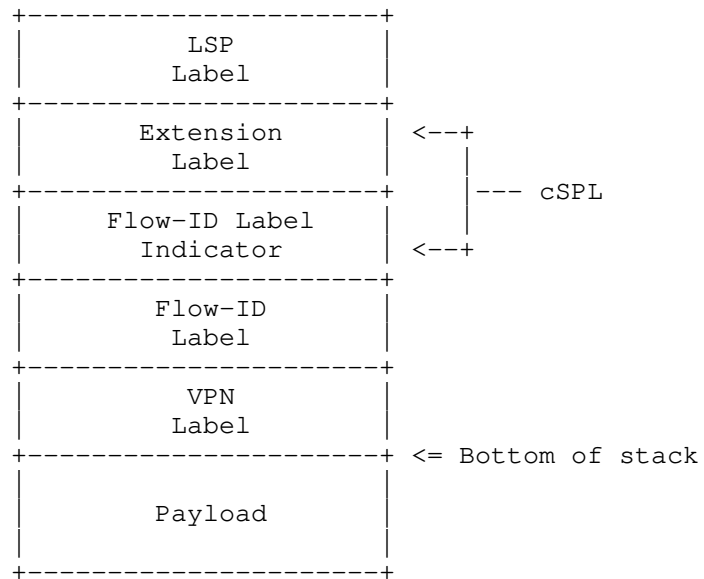


Figure 2: Applying Flow-ID to MPLS LSP

(2) Layout of Flow-ID Label when applied to MPLS VPN traffic.

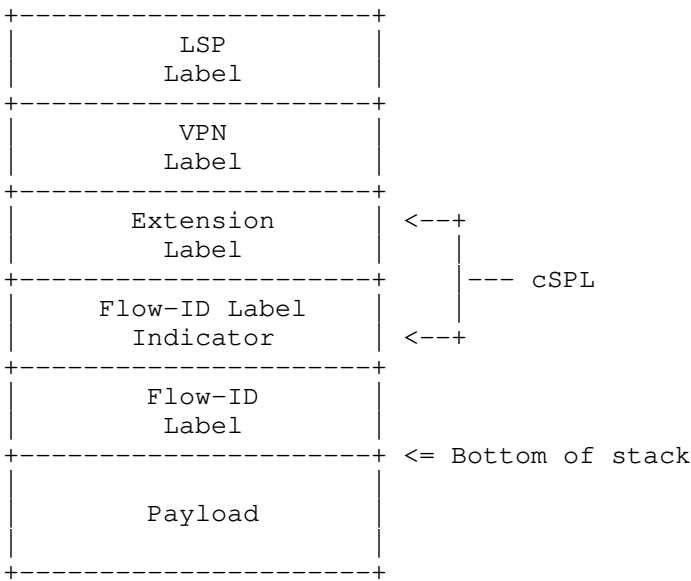


Figure 3: Applying Flow-ID to MPLS VPN

(3) Layout of Flow-ID Label when applied to both MPLS LSP and MPLS VPN traffic.

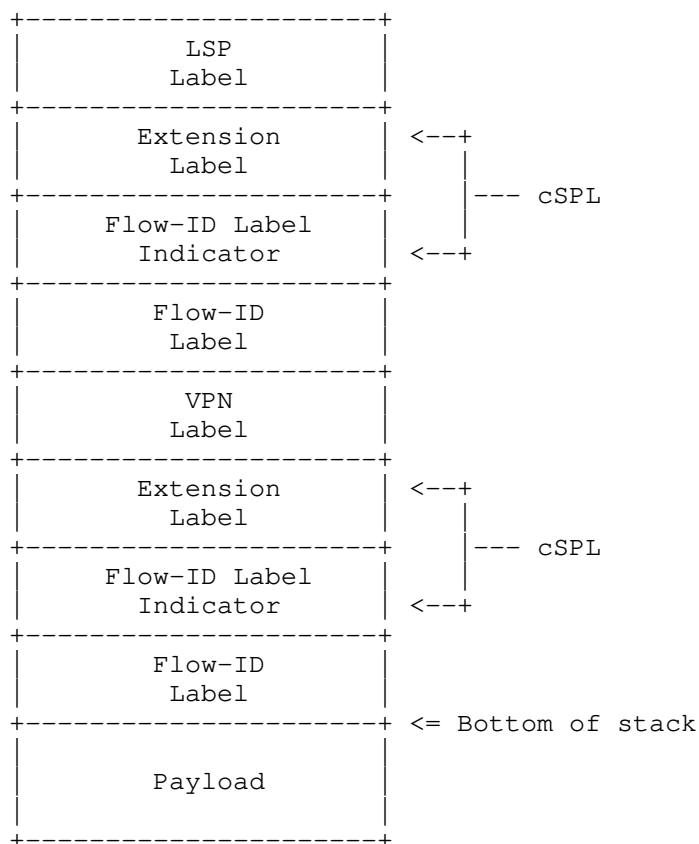


Figure 4: Applying Flow-ID to both MPLS LSP and MPLS VPN

Note that here VPN label can be MPLS PW label, MPLS Ethernet VPN label or MPLS IP VPN label, and it's also called VC label as defined in [RFC4026].

Also note that for this example the two Flow-ID values appearing in a label stack MUST be different, that is to say, Flow-ID Label applied to MPLS LSP and Flow-ID Label applied to MPLS VPN share the same value space.

3. Procedures of Encapsulation, Look-up and Decapsulation

The procedures for Flow-ID label encapsulation, look-up and decapsulation are summarized as follows:

- o The ingress node inserts the Extension Label, the Flow-ID Label Indicator, alongside with the Flow-ID label, into the MPLS label stack. At the same time, the ingress node sets the color-marking field, as needed by alternate-marking technique, and sets the Flow-ID value, as defined in this document.
- o The transit nodes look up the Flow-ID label with the help of the Extension Label and the Flow-ID Label Indicator, and transmit the collected information to an external NMS or a controller, which includes the values of the block counters and the timestamps of the marked packets, along with the value of the Flow-ID, referring to the procedures of alternate marking method.
- o The egress node pops the Extension Label and the Flow-ID Label Indicator, alongside with the Flow-ID label, from the MPLS label stack. This document doesn't introduce any new procedure regarding to the process of the decapsulated packet.

4. Procedures of Flow-ID allocation

There are two ways of allocating Flow-ID, one way is to allocate Flow-ID by manual trigger from the network operator, and the other way is to allocate Flow-ID by automatic trigger from the ingress node, details are as follows:

- o In the case of manual trigger, the network operator would manually input the characteristics (e.g. IP five tuples and IP DSCP) of the measured IP traffic flow, then the NMS or the controller would generate one or two Flow-IDs based on the input from the network operator, and provision the ingress node with the characteristics of the measured IP traffic flow and the corresponding allocated Flow-ID(s).
- o In the case of automatic trigger, the ingress node would identify the IP traffic flow entering the measured path, export the characteristics of the identified IP traffic flow to the NMS or the controller by IPFIX [RFC7011], then the NMS or the controller would generate one or two Flow-IDs based on the export from the ingress node, and provision the ingress node with the characteristics of the identified IP traffic flow and the corresponding allocated Flow-ID(s).

The policy pre-configured at the NMS or the controller decides whether one Flow-ID or two Flow-IDs would be generated. If the performance measurement on VPN traffic is enabled, then one Flow-ID applied to MPLS VPN would be generated; if the performance measurement on LSP tunnel is enabled, then one Flow-ID applied to MPLS LSP would be generated; if both of them are enabled, then two

Flow-IDs respectively applied to MPLS VPN and MPLS LSP would be generated.

Whether using manual trigger or using automatic trigger, the NMS or the controller MUST guarantee every generated Flow-ID is unique within the administrative domain.

5. FLC and FRLD Considerations

Analogous to the Entropy Label Capability (ELC) defined in Section 5 of [RFC6790], and the Entropy Readable Label Depth (ERLD) defined in Section 4 of [RFC8662], the Flow-ID Label Capability (FLC) and the Flow-ID Readable Label Depth (FRLD) are defined in this document. Both FLC and FRLD have the similar semantics with ELC and ERLD to a router, except that the Flow-ID is used in its flow identification function while the Entropy is used in its load-balancing function.

The ingress node MUST insert each Flow-ID Label at an appropriate depth, which ensures the node that needs to process the Flow-ID Label has the FLC. How the ingress node knows the Flow-ID Label processing node has the FLC is outside the scope of this document.

The ingress node SHOULD insert each Flow-ID Label within an appropriate FRLD, which is the minimum FRLD of all on-path nodes that needs to read and use the Flow-ID Label in question. How the ingress node knows the appropriate FRLD for each Flow-ID Label is outside the scope of this document.

When SR paths are used as transport, the label stack grows as the number of on-path segments increases, if the number of on-path segments is high, that may become a challenge for the Flow-ID Label to be placed within an appropriate FRLD. In order to overcome this potential challenge, an implementation MAY provide flexibility to the ingress node to place Flow-ID Label between SID labels, i.e., multiple identical Flow-ID Labels at different depths MAY be interleaved with SID labels, when that happens a sophisticated network planning may be needed and it's beyond the scope of this document.

6. Security Considerations

This document introduces the performance measurement domain that is the scope of a Flow-ID Label. The Flow-ID Label Indicator and Flow-ID Label MUST NOT be signaled and distributed outside one performance measurement domain. Improper configuration so that the Flow-ID Label being passed from one domain to another would likely result in potential Flow-ID conflicts.

To prevent packets carrying Flow-ID Label from leaking from one domain to another, the domain boundary nodes SHOULD deploy some policies (e.g., ACL) to filter out the packets. Specifically, in the sending end, the domain boundary node SHOULD filter out the packets that carry the Flow-ID Label Indicator and are sent to other domain; in the receiving end, the domain boundary node SHOULD drop the packets that carry the Flow-ID Label Indicator and are from other domains.

7. IANA Considerations

In the Special-Purpose MPLS Label Values registry defined in [SP-MPLS-Label], a new Extended Special-Purpose MPLS Label Value for Flow-ID Label Indicator is requested from IANA as follows:

Extended Special-Purpose MPLS Label Value	Description	Semantics Definition	Reference
TBA1	Flow-ID Label Indicator	Section 2	This Document

Table 1: New Extended Special-Purpose MPLS Label Value for Flow-ID Label Indicator

8. Acknowledgements

The authors would like to acknowledge Loa Andersson, Tarek Saad, Stewart Bryant, Rakesh Gandhi, Greg Mirsky, Aihua Liu, Shuangping Zhan and Ming Ke for their careful review and very helpful comments.

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>>.

[SP-MPLS-Label]

"Special-Purpose MPLS Label Values", 2019,
<<https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>>.

9.2. Informative References

[I-D.bryant-mpls-sfl-control]

Bryant, S., Swallow, G., and S. Sivabalan, "A Simple Control Protocol for MPLS SFLs", draft-bryant-mpls-sfl-control-08 (work in progress), June 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-10 (work in progress), July 2020.

[I-D.ietf-ippm-ioam-direct-export]

Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-01 (work in progress), August 2020.

[I-D.ietf-mpls-sfl-framework]

Bryant, S., Chen, M., Swallow, G., Sivabalan, S., and G. Mirsky, "Synonymous Flow Label Framework", draft-ietf-mpls-sfl-framework-10 (work in progress), August 2020.

[I-D.ietf-mpls-spl-terminology]

Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-03 (work in progress), August 2020.

[RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <<https://www.rfc-editor.org/info/rfc4026>>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

- [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>>.
- [RFC8372] Bryant, S., Pignataro, C., Chen, M., Li, Z., and G. Mirsky, "MPLS Flow Identification Considerations", RFC 8372, DOI 10.17487/RFC8372, May 2018, <<https://www.rfc-editor.org/info/rfc8372>>.
- [RFC8662] Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy Label for Source Packet Routing in Networking (SPRING) Tunnels", RFC 8662, DOI 10.17487/RFC8662, December 2019, <<https://www.rfc-editor.org/info/rfc8662>>.

Authors' Addresses

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Xiao Min
ZTE
Nanjing
China

Email: xiao.min2@zte.com.cn

Tianran Zhou
Huawei
Beijing
China

Email: zhoutianran@huawei.com

Ximing Dong
FiberHome
Wuhan
China

Email: dxm@fiberhome.com

Yoav Peleg
Broadcom
USA

Email: yoav.peleg@broadcom.com

MPLS Workgroup
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2019

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Weiqiang Cheng
China Mobile
March 10, 2019

Inter-domain Use Cases of Segment Routing with MPLS Data Plane for
Transport Network
draft-hu-mpls-sr-inter-domain-use-cases-01

Abstract

This document discusses the inter-domain scenarios for Transport Profile of SR-MPLS (SR-MPLS-TP), including SR-MPLS-TP inter-working with MPLS-TP network.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 11, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Transport Profile in SR-MPLS	3
4. SR-MPLS-TP Inter-domain	4
4.1. SR-MPLS-TP Stitching Inter-domain	4
4.1.1. Inter-domain Path Segment	4
4.1.2. Border Node Inter-domain Scenario	5
4.1.3. Border Link Inter-domain Scenario	5
4.2. SR-MPLS-TP Nesting Inter-domain	7
5. SR-MPLS-TP Inter-working with MPLS-TP	8
6. Security Considerations	10
7. Acknowledgements	10
8. IANA Considerations	10
9. Normative References	10
Authors' Addresses	11

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an SR Policy instantiated as an ordered list of instructions called "segments". A segment can represent any instruction, topological or service based. A segment can have a semantic local to an SR node or global within an SR domain. SR supports per-flow explicit routing while maintaining per-flow state only at the ingress nodes of the SR domain. Segment Routing can be instantiated on MPLS data plane or IPv6 data plane. The former is referred to as SR-MPLS [I-D.ietf-spring-segment-routing-mpls], the latter is SRv6 [I-D.ietf-6man-segment-routing-header]. SR-MPLS leverages the MPLS label stack to construct the SR path, and SRv6 uses the Segment Routing Header to construct the SR path.

[I-D.cheng-spring-mpls-path-segment] defines a Path Segment identifier to support bidirectional path correlation for transport network. This document defines an inter-domain path segment and discusses the inter-domain use cases in the context of the Transport Profile of SR-MPLS, referred to in this document as SR-MPLS-TP, and describes the use case related to SR-MPLS-TP inter-working with the MPLS-TP network.

2. Conventions used in this document

2.1. Terminology

A->B SID list: The SID List from SR node A to SR node B.

B-SID: Binding SID.

e-Path: End-to-end Path segment.

MPLS-TP: MPLS Transport Profile.

s-Path: Sub-path Path Segment.

i-Path/i-PSID: Inter-domain Path Segment.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SR-MPLS-TP: Transport Profile of SR-MPLS.

Border node inter-domain: Two domains interconnects with an edge node which belongs to both domains.

Border link inter-domain: Two domains interconnects with an inter-link which connects the edge node in each domain.

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. Transport Profile in SR-MPLS

In the SR-MPLS network, an SR path is a unidirectional path. [I-D.cheng-spring-mpls-path-segment] defines a Path Segment identifier to support SR bidirectional path correlation for transport network. In the context of the Transport Profile of SR-MPLS, referred to in this document as SR-MPLS-TP, a Path Segment uniquely identifies an SR path in a specific context. For example, the Path Segment is used to indicate the intra-domain path in a single domain and correlate the two unidirectional SR paths at both ends of the paths.

In multi-domain scenario, the SR bidirectional end-to-end path MUST to be established in transport network. The SR-MPLS-TP inter-domain models include the stitching inter-domain model and the nesting inter-domain model. Path Segment MAY also be used to indicate the inter-domain path or the end-to-end path and correlate the inter-domain paths or end-to-end unidirectional paths.

4. SR-MPLS-TP Inter-domain

Two SR-MPLS-TP inter-domain models are discussed in this document including the stitching inter-domain model and the nesting inter-domain model. Two use cases of stitching SR-MPLS-TP domains, using a border node inter-domain and a border link inter-domain, are described in Section 4.1.1 and Section 4.1.2 respectively.

4.1. SR-MPLS-TP Stitching Inter-domain

4.1.1. Inter-domain Path Segment

In the stitching inter-domain model, the end-to-end SR path being split into multiple segments. And each segment can be identified by an inter-domain path segment (i-Path or i-PSID). The inter-domain path segment is valid in the corresponding domain and the border nodes maintain the forwarding entries of that i-Path segment mapping to the next i-Path. In the headend node, the i-Path can be mapped to the inter-domain path of reverse direction and correlates the two unidirectional paths. The border nodes should install the following MPLS data entries for Path segments:

incoming label: i-Path
outgoing label: the SID list of the next domain or link + next i-Path

Taking Figure 1 as an example, the border node X installs the MPLS data entries:

incoming label: i-Path(A->X)
outgoing label: X->Y SID list + i-Path(X->Y)

The i-Path can be a locally unique label and assigned from the Segment Routing Local Block (SRLB). It is required that the controller(e.g., PCE) assigns the label to ensure the ingress and the egress node can recognize it and it also can be assigned from egress node of each domain. PCEP based i-Path allocation and procedure is defined in [I-D.xiong-pce-stateful-pce-sr-inter-domain].

4.1.2. Border Node Inter-domain Scenario

The Figure 1 displays the border node inter-domain scenario. SR node X and SR node Y are the border nodes of two different domains. The i-Paths from A->X, X->Y, and Y->Z are used for the inter-domain path segment. The ingress SR node A encapsulates the data packet with i-Path (A->X) and A->X SID list. The data packet is forwarded to SR node X according to the A->X SID list. Node X pushes the i-Path (X->Y) and X->Y SID list based on the above mentioned forwarding entry. The data packet is forwarded to node Y and then to the SR node Z based on the same forwarding procedure. In node Z, the i-Path (Y->Z) can be mapped to the path from Z to Y of reverse direction and correlates the two unidirectional paths. The packet transmission of the reverse direction is the same with the forwarding direction with different i-Paths.

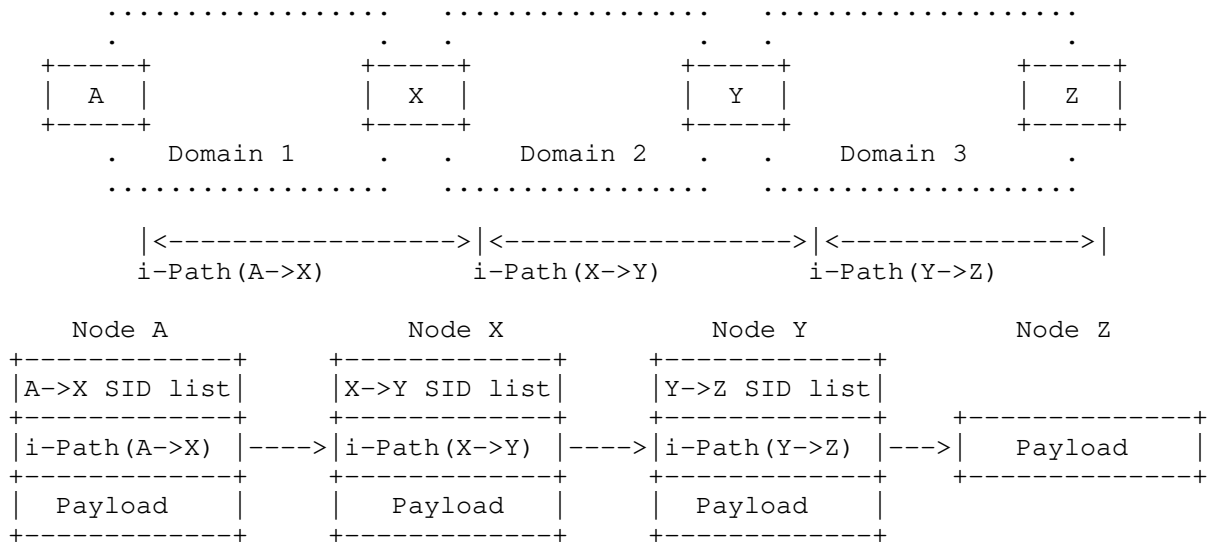


Figure 1: Stitching Border Node Inter-Domain Scenario

4.1.3. Border Link Inter-domain Scenario

Figure 2 illustrates the border link inter-domain scenario. All the SR nodes belong to a single domain. Neighboring border nodes of different domains are interconnected by direct physical or logical links. Ingress SR node A encapsulates the data packet with i-Path (A->B) and A->B SID list. The data packet is forwarded to SR node B

according to the A->B SID list. Node B pushes i-Path (B->C) and the inter-domain link label(B->C SID) based on the forwarding entry, and forwards the packet to node C. SR node C forwards the packet to node X, then node X forwards the packets to node Y. The data packet reaches the destination SR node Z according to the same forwarding procedure. In node Z, the i-Path (Y->Z) can be mapped to the path from Z to Y of reverse direction and correlates the two unidirectional paths. The packet transmission of the reverse direction is the same with the forwarding direction with different i-Paths.

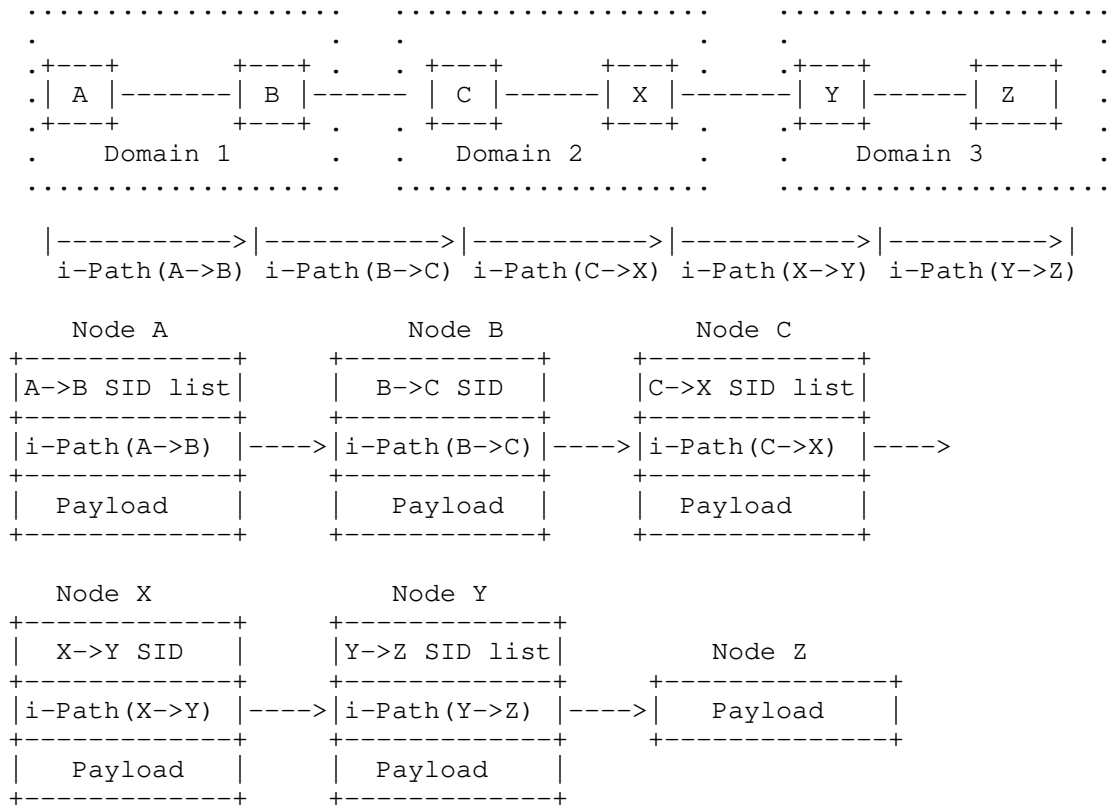


Figure 2: Stitching Border Link Inter-Domain Scenario

4.2. SR-MPLS-TP Nesting Inter-domain

The nesting inter-domain model is described in [I-D.cheng-spring-mpls-path-segment], an end-to-end path segment, also referred to as e-Path, is used to indicate the end-to-end path, and an s-Path is used to indicate the intra-domain path. The e-Path is encapsulated at the ingress nodes and decapsulated at the egress nodes. The transit nodes, even the border nodes of domains, are not aware of the e-Path segment. Only the s-Path is pushed and popped at the border nodes of the corresponding domain.

Figure 3 shows the SR-MPLS-TP nesting inter-domain scenario. The e-Path(A->Z) is used to indicate the end-to-end path. The s-Path is used to identify the domain's sub-path. The e-Path, s-Path and SR list are pushed by the ingress node. To reduce the size of the label stacks, the use of the binding SID [RFC8402] is recommended to replace the SR list of each domain. As shown in Figure 3, the B-SID(X->Y) is used to replace the X->Y SID list. Ingress node A pushes e-Path(A->Z), B-SID(Y->Z), B-SID(X->Y), s-Path(A->X) and A->X SID list in turn. When the packet is received at node X, the s-Path(A->X) and X->Y SID list are popped, and the new s-Path(X->Y) is pushed. Also, X->Y SID list replaces B-SID(X->Y) to indicate that packet to be forwarded from node X to node Y. The data packet reaches the SR node Z according to the same forwarding procedure. In SR node Z, the e-Path (A->Z) is used to correlate the two unidirectional end-to-end paths.

The e-Path can be a globally unique or local label. If the e-Path is globally unique, it MUST be assigned from the SRGB block of each domain. If the e-Path is a local label, it is required that the controller(e.g., PCE) or a super controller (e.g., hierarchical PCE) assigns the label to ensure the ingress(A) and the egress node(Z) can recognize it and there is no SID collision in the ingress and egress domains.

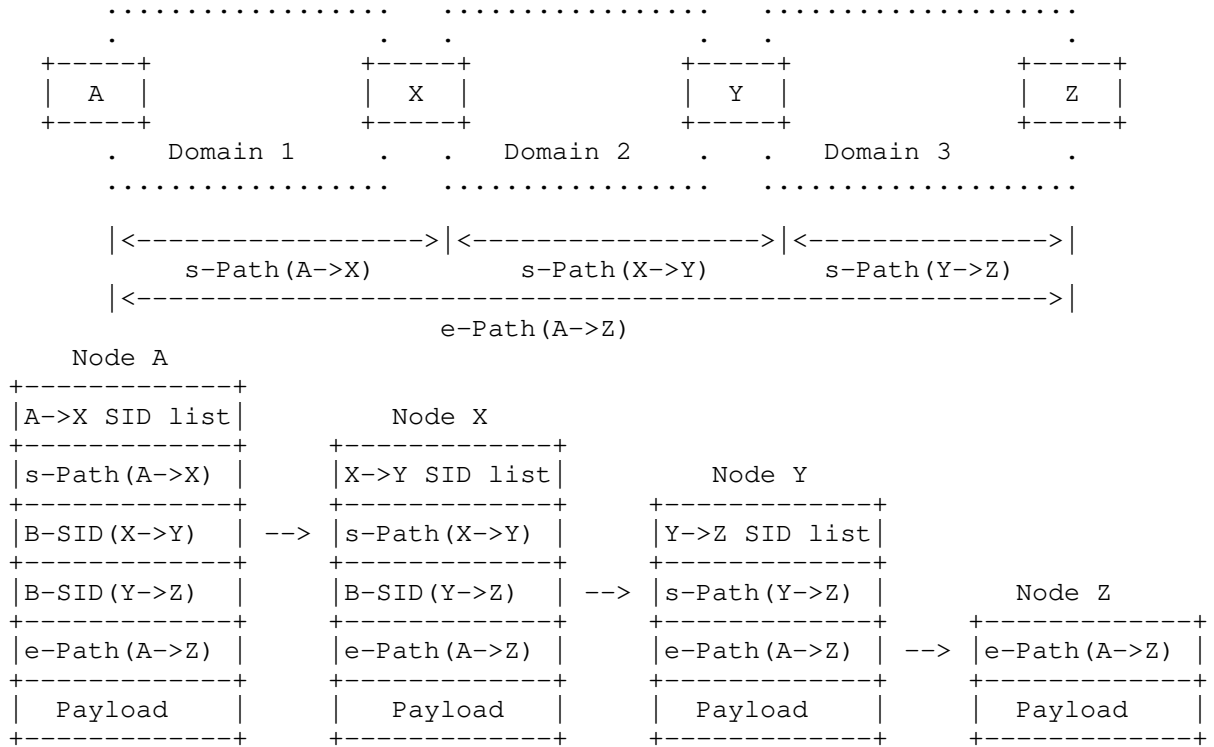


Figure 3: Nesting Inter-Domain Scenario

5. SR-MPLS-TP Inter-working with MPLS-TP

It is a common requirement that SR-MPLS-TP needs to inter-work with MPLS-TP when SR is incrementally being deployed in the MPLS-TP domain.

Figure 4 shows the stitching model of SR-MPLS-TP inter-working with MPLS-TP. The left is the SR-MPLS-TP network, and the right is the MPLS-TP network. The path segment which is i-Path is used for the bidirectional tunnel correlation in SR-MPLS-TP network. The edge nodes of the SR-MPLS-TP network should map the path segment to the corresponding MPLS-TP label for bidirectional tunnel indication in the MPLS-TP network.

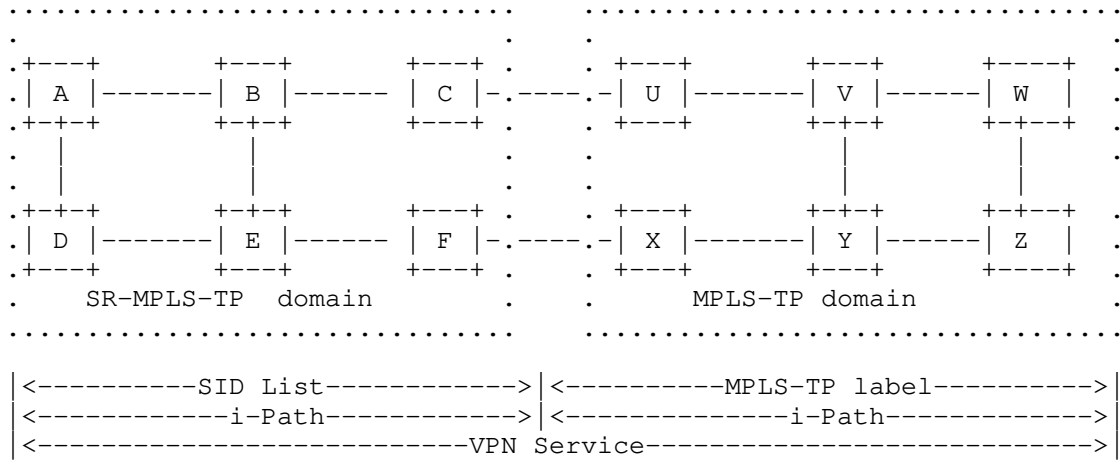


Figure 4: Stitching SR-MPLS-TP inter-working with MPLS-TP

Figure 5 displays the nesting model of SR-MPLS-TP and MPLS-TP inter-working. Compared with stitching SR-MPLS-TP inter-working with MPLS-TP, the path segment is e-Path and presents end-to-end encapsulation in the packet from SR-MPLS-TP domain to MPLS-TP domain.

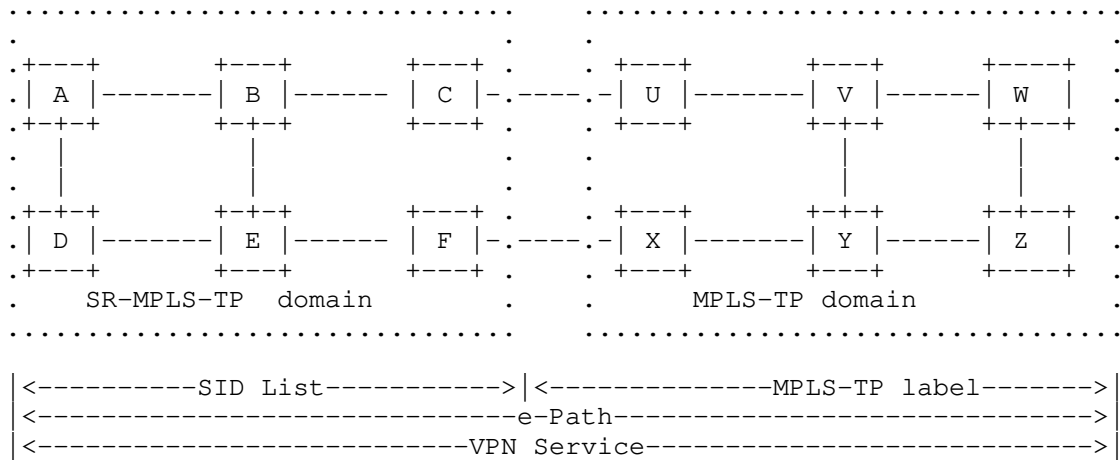


Figure 5: Nesting SR-MPLS-TP inter-working with MPLS-TP

The requirements for the SR-MPLS-TP inter-working with MPLS-TP are as follows:

- o It is required to establish the end-to-end VPN service through the SR-MPLS-TP domain and the MPLS-TP domain;
- o The path segment MUST be carried through SR-MPLS-TP and MPLS-TP domains in the nesting SR-MPLS-TP inter-working with MPLS-TP model.
- o The edge nodes of the MPLS-TP network SHOULD process the path segment from the SR-MPLS-TP network.
- o The edge nodes in the MPLS-TP SHOULD process MPLS SID sent from the MPLS-SR-TP domain
- o The edge nodes in the SR-MPLS-TP network SHOULD process the MPLS-TP labels sent from the MPLS-TP domain;

6. Security Considerations

TBA

7. Acknowledgements

TBA

8. IANA Considerations

TBA

9. Normative References

[I-D.cheng-spring-mpls-path-segment]

Cheng, W., Wang, L., Li, H., Chen, M., Gandhi, R., Zigler, R., and S. Zhan, "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment-03 (work in progress), October 2018.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.

[I-D.ietf-pce-association-group]

Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-07 (work in progress), December 2018.

- [I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-18 (work in progress), December 2018.
- [I-D.xiong-pce-stateful-pce-sr-inter-domain]
Xiong, Q., hu, f., Mirsky, G., and W. Cheng, "Stateful PCE for SR-MPLS-TP Inter-domain", draft-xiong-pce-stateful-pce-sr-inter-domain-00 (work in progress), December 2018.
- [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>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [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>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [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>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual
China

Email: hufwei@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

INTERNET-DRAFT
Intended Status: Standards Track

J. Kumar
J. Lemon
Y. Peleg
Broadcom Inc.
K. Kompella
Juniper Networks
March 11, 2019

Expires: September 12, 2019

MPLS Inband Network Telemetry
draft-kumar-mpls-mint-00

Abstract

MPLS Inband Network Telemetry (MINT) records flow specific information from end stations and/or switches across a network. This document describes the method to collect data on a per hop basis across a network and perform localized or end to end analytics operations on the data. This document also describes the use of the Extension Label, value 15, for specifying extended special purpose labels for specifying presence of MINT data.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
1.1 Terminology	4
1.2 Scope	5
1.3 Applicability	5
1.4 Motivation	6
2. Requirements	6
2.1 Encapsulation Requirements	6
2.2 Operational Requirements	6
3. MINT Domains	6
3.1 MINT Domain	8
3.2 MINT Function Nodes	8
3.2.1 Initiating Function Node	8
3.2.2 Transit Function Node	9
3.2.3 Terminating Function Node	9
3.2.4 Metadata Fragmentation Function	9
3.3 MINT Cloning, Truncation, and Drop	10
3.4 MINT Label Stack	10
3.4.1 MINT Metadata Header	12
3.4.2 MINT Checksum Header	13
3.4.3 MINT Metadata Fragmentation (MF) Header	14
3.5 MINT Metadata	15
3.5.1 Global Name Space (GNS) Identifier	15
3.5.2 Local Name Space (LNS) Identifier	16
3.5.3 Device ID	16
3.6 MINT Network Overhead	16
3.7 MINT Analytics	16
3.8 MINT Packet Format	17
3.8.1 MINT Packet Format with TS Flag Set	17
3.9 MINT Load Balancing	18
4. IANA Considerations	19
4.1. Extended Special-Purpose MPLS Label Values Registry	19

5. Security Considerations	19
6. References	19
6.1 Normative References	19
6.2 Informative References	19
Authors' Addresses	20

1. Introduction

This document describes MPLS Inband Network Telemetry (MINT), which is a mechanism to enable the collection of metadata for an analyzed MPLS flow.

The sequence of per hop metadata in the packet preceded by a MINT header is referred to as a MINT stack. The presence of a MINT stack is indicated by an extended special purpose label. The "Extension Label", value 15, is used to indicate a following label, which will be the MINT label in this case.

The MINT header and metadata definition are the same as defined in [I-D.draft-kumar-ippm-ifa-01].

1.1 Terminology

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

LSP: Label Switched Path

MPLS: Multi Path Label Switching

MINT: MPLS Inband Network Telemetry

MTU: Maximum Transmit Unit

MINT Domain: A set of MPLS nodes within a single administration with all nodes participating in MINT.

MINT Node: A single node in a MINT domain.

MINT Metadata: Information inserted by a MINT node.

MINT Stack: Ordered collection of per hop metadata.

MINT Label: IANA allocated extended special purpose label. Indicates the presence of a MINT stack.

MINT Initiator: A MINT node at the beginning edge of a MINT domain that is responsible for inserting an Extension label, a MINT label, and the initial MINT metadata.

MINT Transit: A MINT node within the MINT domain, responsible for

adding MINT metadata for the current node.

MINT Terminator: A MINT node at the ending edge of a MINT domain that is responsible for removing the Extension label, the MINT label, and the MINT Stack and exporting the MINT Stack to a collector.

1.2 Scope

This document describes MINT deployment, header definitions, packet format, and data path functions.

MINT deployment involves defining a MINT Domain and understanding the requirements in terms of traffic overhead and points of data collection. Given that MINT inserts two extra labels in the label stack, consideration **MUST** be given to network devices' capability to parse the depth of the label stack. The scope of MINT is from an end station and/or ToR, through any/all nodes in the MPLS path, and terminating in a network switch and/or an end station. In case of a service provider network, the scope of MINT **MAY** encompass one carrier to another carrier. Special care **MUST** be taken for leaking the MINT stack across a MINT domain for security reasons.

MINT can create a synthetic stream of MPLS traffic and use it to collect metadata along the path. This sampled stream is later discarded. MINT can also insert metadata on a per packet basis in live traffic.

This draft defines an identification mechanism using an extended special purpose label for MINT packets.

1.3 Applicability

MINT is capable of providing traffic analysis for a given LSP. The LSP may be tunneled in another LSP in part of the MINT domain. In such cases more than one MINT stack **MAY** be present in the MINT packet.

A very important requirement of MINT traffic is preserving the same path for the flow. Since inserting a MINT label will change the label stack and possibly keys to load balancing functions, a MINT domain **SHOULD** use an entropy label as defined in [RFC 6790] to preserve the flow path.

Inserting additional an label and metadata will increase the packet size and may impact path MTU. MINT provides a metadata fragmentation function to keep the packet size below path MTU.

1.4 Motivation

The main motivation for MINT is to collect analyzed metadata from packets within a flow for a given application.

Each hop in the flow path MAY insert metadata in the packet or MAY export metadata to a collector.

2. Requirements

MINT requirements are the same as IFA requirements and are defined with operational efficiency, performance of the network, and cost of hardware in mind.

2.1 Encapsulation Requirements

MINT packets MUST be clearly marked and identifiable so that a networking element in the flow path can insert metadata or perform other MINT operations.

MINT packets need to be easily identified for performance reasons. A new extended special purpose label is requested for MINT.

MINT encapsulation MAY support the ability to fragment metadata.

MINT encapsulation MAY support direct export of metadata instead of inserting it into the packet.

2.2 Operational Requirements

MINT MUST preserve the flow path across the network. This can be done either by using specific key fields for load balancing functions or by using an entropy label.

MINT MUST support cloning, live traffic, checksum, fragmentation, global name spaces, and local name spaces using the MINT header definitions.

3. MINT Domains

MINT performs flow analysis, and possible actions on the flow data, in-band. Once a flow is enabled for analysis, a MINT node with the role of "Initiator" makes a copy of the flow or samples the live traffic flow, or tags a live traffic flow for analysis and data

collection. Copying of a flow is done by sampling or cloning the flow. These new packets are representative packets of the original flow and possess the exact same characteristics as the original flow.

This means that MINT packets traverse the same path in the network and same queues in the networking element as the original packet would. Figure 1 shows the MINT based Telemetry Framework. The terminating node is responsible for terminating the MINT flow by summarizing the metadata of the entire path and sending it to a collector.

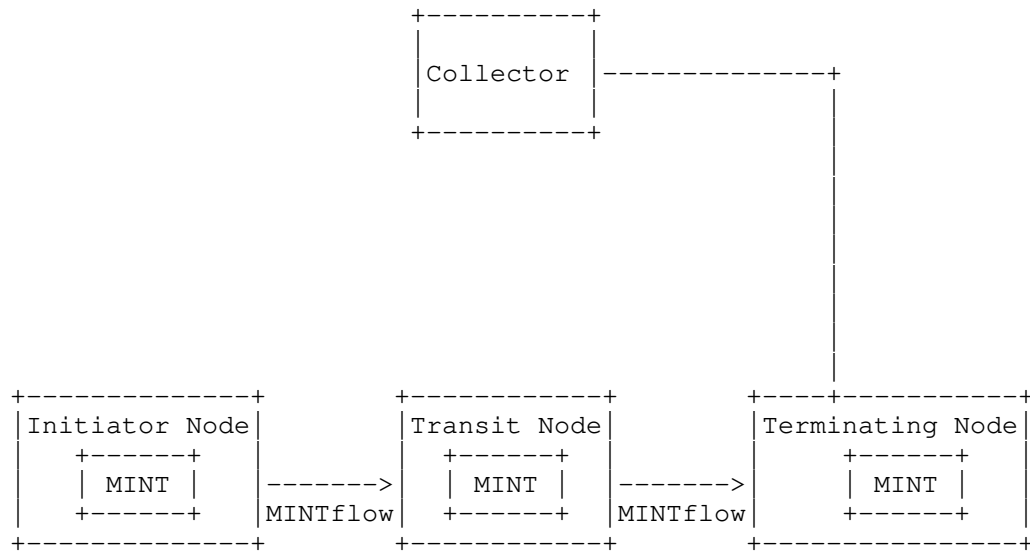


Figure 1 MINT Domain Framework without fragmentation

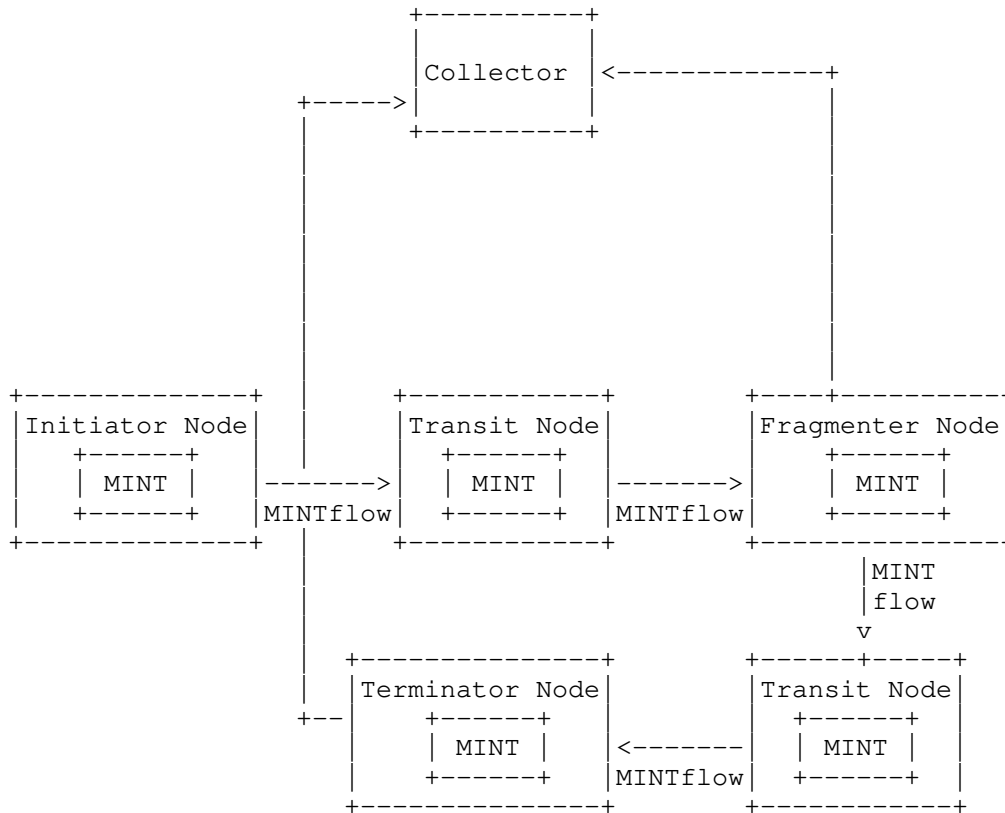


Figure 2 MINT Domain Framework with metadata fragmentation

3.1 MINT Domain

A MINT Domain is the domain of interest where MINT monitoring is enabled. A MINT Domain MUST have designated MINT function nodes. Initiating and Terminating function MINT nodes are always at the edge of the MINT domain. Internal nodes in the MINT Domain are always Transit function nodes.

3.2 MINT Function Nodes

There are three types of MINT functional nodes with respect to a specific set of flows. Each node MAY perform metadata fragmentation function as well.

3.2.1 Initiating Function Node

An end station, a switch, or any other middleware can perform the MINT initiating function. It is advantageous to keep this role closest to the application to maximize flow visibility. A MINT

initiating function node performs the following functions for a flow:

- Samples the flow traffic of interest based on a configuration.
- Converts the traffic into a MINT flow by adding a MINT header to each sample.
- Updates the packet with initiating function node metadata.
- MAY mandate a specific template ID metadata by all networking elements.
- MAY mandate tail stamping of metadata by all networking elements.

3.2.2 Transit Function Node

A MINT transit node is responsible for inserting transit node metadata in the MINT packets in the specified flow.

3.2.3 Terminating Function Node

A MINT terminating node is responsible for the following for a flow:

- Inserts terminating node metadata in a MINT packet.
- Performs a local analytics function on one or more segments of metadata, e.g., threshold breach for residence time, congestion notifications, and so on.
- Filters a MINT flow in case of cloned traffic.
- Sends a copy or report of the packet to a collector.
- Removes the MINT headers and forwards the packet in case of live traffic.

3.2.4 Metadata Fragmentation Function

There are cases where the size of metadata may grow too big for link MTU or path MTU, or where it imposes excessive overhead for the terminating function node to remove it. This is especially true in networks with a large number of hops between the initiator function node and the terminating function node. This is also true where the size of per hop metadata itself is large. For such cases, MINT defines a metadata fragmentation function. The metadata fragmentation function allows removal of metadata from the packet and the sending of a copy/report of the packet to the collector. Correlation of metadata fragments and recreation of metadata stack for the entire flow path is done by the collector.

There is no dedicated node performing the metadata fragmentation function. As a MINT packet traverses the hops in a MINT Domain, any node MAY detect the need to fragment the packet's metadata stack and perform metadata fragmentation.

Metadata fragmentation is done if the MINT header in the packet has "MDF" bit set and the current length of the metadata would exceed the maximum length after the addition of metadata by the current node. A

node MAY create a copy of the packet or create a MINT report, remove the existing metadata stack from the packet, insert its own metadata, and finally forward the packet. A node MAY also update the MINT MDF (Meta Data Fragment) header, fragment identifier, and current length.

The maximum length in a MINT header, if set to 0, MAY trigger the metadata fragmentation special function. This mechanism can be used to generate MINT reports at each hop and never insert metadata in the packet. If the maximum length is set to 0, a node MAY NOT insert MINT metadata, SHOULD create a MINT report or copy of the packet including its own metadata, and MUST increment the MINT MDF header fragment identifier field.

3.3 MINT Cloning, Truncation, and Drop

MINT allows cloning of live traffic. It is expected that cloned traffic will have the same network path characterization as the original traffic, i.e., follow the same network path, use the same queues, etc.

Cloned traffic can be truncated to accommodate the MTU of the MINT Domain.

Cloned traffic MUST be dropped by the terminating function node of the MINT Domain.

3.4 MINT Label Stack

The MINT label stack is described below. An extended special-purpose label (ESPL) is used to identify a MINT packet in the MPLS label stack at the level to be measured.

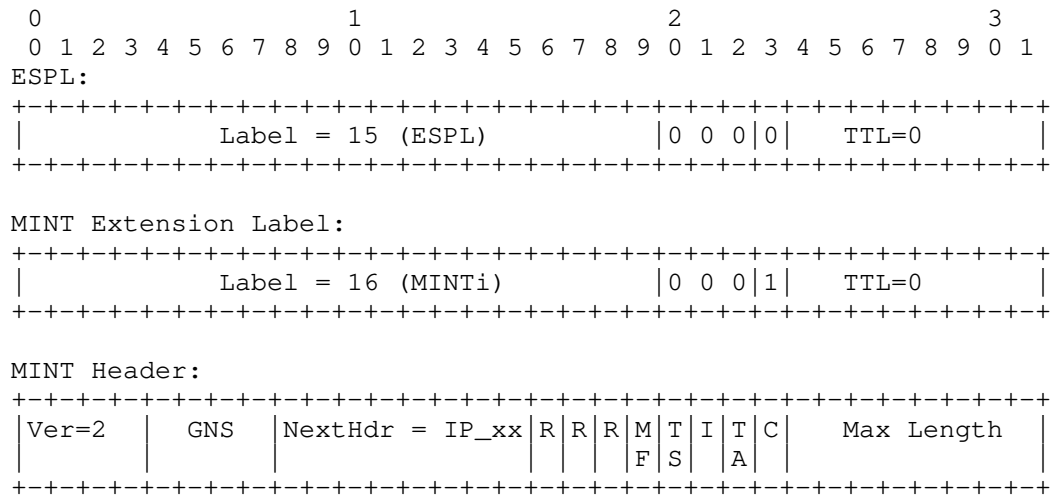


Figure 3 MINT Label Headers Format

ESPL:

A label value of 15 identifies an extended special-purpose label (ESPL), which indicates that the following label is an extension label. The EXP bits SHOULD be set to 0. The S bit MUST be set to 0 to indicate that more labels follow. The TTL for the ESPL MUST be zero to ensure that it is not used inadvertently for forwarding.

MINT Extension Label:

A label value of 16 identifies an MINT extension label, which indicates that header following this label is a MINT header. The EXP bits SHOULD be set to 0. The S bit MUST be set to 1 to indicate that no more labels follow. The TTL for the MINT extension label MUST be zero to ensure that it is not used inadvertently for forwarding.

MINT Header:

(1) Version (4 bits). Specifies the version of MINT header. The version MUST be set to 2 for this.

(2) GNS (4 bits) - Global Name Space. Specifies the MINT Domain scoped name space for the MINT metadata.

(3) Protocol Type (8 bits) - IP Header protocol type. This indicates what protocol follows MINT.

(4) Flags (8 bits)

0: R - Reserved. MUST be initialized to 0 on transmission and

ignored on receipt.

1: R - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

2: R - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

3: MF - Metadata Fragment. Indicates the presence of the optional metadata fragment header. This header is inserted and initialized by the initiator node. If the MF bit is set, nodes in the path MAY perform fragmentation of metadata stack if the current length exceeds the maximum length.

4: TS - Tail Stamp. Indicates the MINT Domain is requiring tail stamping of metadata.

5: I - Inband. Indicates this is live traffic. Strip and forward MUST be performed by the terminator node if this bit is set.

6: TA - Turn Around. Indicates that the MINT packet needs to be turned around at the terminating node of the MINT Domain and sent back to the source if a return path is known. This bit MAY be used for probe packets where probes are collection bidirectional information in the network. This is analogous to an echo request and echo reply. A packet with the TA bit set collects metadata in one direction, and after it is turned around by the terminating function node, collects metadata in the reverse direction.

7: C - Checksum. Indicates the presence of the optional checksum header. The checksum MUST be computed and updated for the MINT header and metadata at each node that modifies the header and/or metadata. A node MAY perform checksum validation before updating the checksum.

(5) Max Length (8 bits). Specifies the maximum allowed length of the metadata stack in multiples of 4 octets. This field is initialized by the initiator node. Each node in the path MUST compare the current length with the max length, and if the current length equals or exceeds the max length, the transit nodes MUST stop inserting metadata.

3.4.1 MINT Metadata Header

The MINT metadata header is always present.

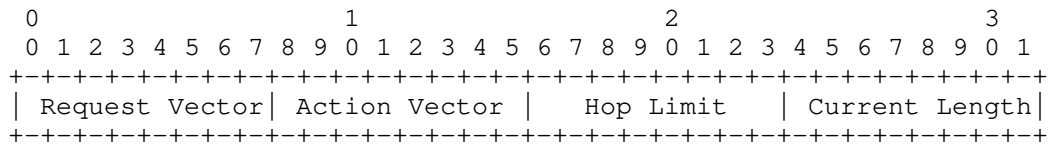


Figure 4 MINT Metadata Header Format

Request Vector (8 bits) - This vector specifies the presence of fields as specified by GNS. Fields are always 4-octet aligned. This field can be made extensible by defining a new GNS for a MINT Domain.

Action Vector (8 bits) - This vector specifies node-local or end-to-end action on the MINT packets.

Hop Limit (8 bits) - Specifies the maximum allowed hops in a MINT Domain. This field is initialized by the initiator node. The hop limit MUST be decremented at each hop. If the incoming hop limit is 0, current nodes MUST NOT insert metadata. A value of 0xFF means that the Hop limit check MUST be ignored.

Current Length (8 bits) - Specifies the current length of the metadata in multiples of 4 octets.

3.4.2 MINT Checksum Header

The MINT checksum header is optional. Presence of the checksum header is indicated by the C bit in the flags field of the MINT header.

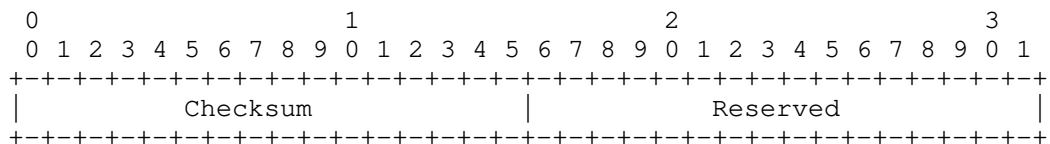


Figure 5 MINT Checksum Header Format

Checksum (16 bits) - The checksum covers the MINT header and metadata stack. An initiator function node MAY compute the full checksum including MINT header and metadata. Other nodes MAY compute delta checksum for the inserted/deleted metadata.

Reserved (16 bits) - Reserved. MUST be initialized to 0 on transmission and ignored on receipt.

3.4.3 MINT Metadata Fragmentation (MF) Header

The MINT metadata fragmentation (MF) header is optional. Presence of the fragmentation header is indicated by the MF bit in the flags field of the MINT header.

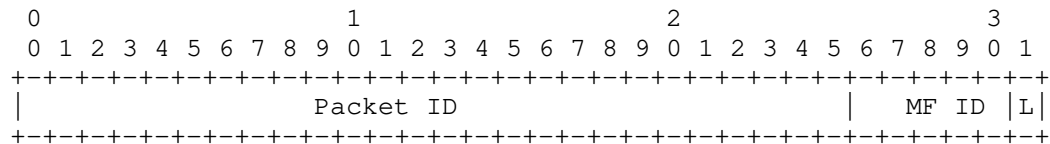


Figure 6 MINT MF Header Format

Packet ID (26 bits) - Packet identification value generated by the initiator node. This value is node scoped.

Metadata Fragment ID (5 bits) - The initiator **MUST** initialize this value to 0. A node performing metadata fragmentation function **MUST** increment the value by 1.

L (1 bit) - This bit is set by the node creating the last metadata fragment. This will **ALWAYS** be the terminating function node. If incoming hop limit is 0, the terminating function node will still generate a copy/report of the packet and **MUST** set the L bit. A collector **MUST** implement mechanism to recover from lost packets/reports with L bit set.

The MF header is a fixed overhead of 4 octets per packet. A network operator **MUST** identify the need for using MINT metadata fragmentation. The following network conditions can be considered:

- If a MINT packet may exceed the link or path MTU of the flow path
- If there are large number of hops in a flow path that could trigger link or path MTU breach
- If the length of metadata creates excessive overhead for terminating function node to delete the metadata
- If each hop needs to generate its own MINT report (postcard mechanism)

With 26 bits of packet id, a maximum datagram lifetime (MDL) of 3 seconds, and an average Internet mix (IMIX) packet size of 512 bytes,

we get 183.25 Gbps of MINT traffic bit rate per node before the packet identifier wraps around. The collector can use [device id, packet id, MF id, L] to rebuild the fragmented packet.

5 bits of MF id will support 32 metadata fragments.

3.5 MINT Metadata

The MINT metadata is the information inserted by each hop after the MINT header. The MINT metadata can be inserted at the following offsets:

- Payload Stamping: Immediately after the layer 4 header. This is the default setting.
- Tail Stamping: After the end of the packet, but before the FCS. This is controlled by the TS bit in the flags field of the MINT header.

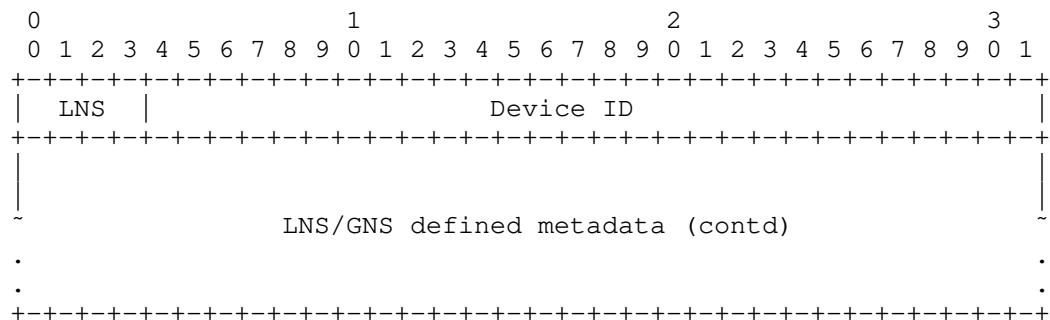


Figure 7 MINT Metadata Format

The MINT metadata header contains a set fields as defined by the name space identifier. Two types of name space identifiers are proposed.

3.5.1 Global Name Space (GNS) Identifier

A Global Name Space (GNS) is specified in the MINT header by the initiator node. The scope of a GNS is a MINT Domain. All networking elements in a MINT Domain MUST insert metadata as per the GNS ID specified in the MINT header. This is defined as the "Uniform Mode" of deployment.

A GNS value of 0xF indicates that metadata in a MINT Domain is defined by the LNS of each hop.

The advantage of using the uniform mode is having a simple and uniform metadata stack. This means less load on a collector for

parsing.

The disadvantage is that metadata fields are supported based on the least capable networking element in the MINT Domain.

3.5.2 Local Name Space (LNS) Identifier

A Local Name Space (LNS) is specified in the metadata header. A GNS value of 0xF in the MINT header indicates the presence of an LNS. This is defined as the "Non-uniform Mode" of deployment.

A switch pipeline MUST parse the GNS field in the MINT header. The parsing result will dictate the name space ID that the hop needs to comply with.

The advantage of using the non-uniform mode is having a flexible metadata stack. This allows each hop to include the most relevant data for that hop.

The disadvantage is more complex parsing by a collector.

3.5.3 Device ID

A 28-bit unique identifier for the device inserting the metadata. If a GNS other than 0xF is present, then the device ID can be expanded to a 32 bit value. This is to support including an IPv4 loopback address as a Device ID.

3.6 MINT Network Overhead

A common problem associated with inserting metadata on a per packet per flow basis is the amount of traffic overhead on the network. MINT is defined to minimize the overhead on the network.

MINT Base Header	: 4 octets
MINT Metadata Header	: 4 octets
MINT Checksum Header	: 4 octets
MINT Fragmentation Header	: 4 octets

Minimum Overhead:

MINT header	: 4 octets
MINT Metadata Header	: 4 octets

Total Min Overhead : 8 octets per packet

3.7 MINT Analytics

There are two kinds of actions considered in this proposal.

(1) Action Bit MAP in MINT Header - This is encoded in the MINT header. Each node in the path MAY use the action bitmap to insert or not insert the metadata based on exceeding a locally-specified threshold. Not inserting the metadata is indicated by setting the field value to -1 (all 1s).

(2) Terminating Node Actions - A terminating node may decide to perform threshold or other actions on the set of metadata in the packet. This information is not encoded in the MINT header.

3.8 MINT Packet Format

The MINT header is treated as a layer 3 extension header. MINT header and metadata stack length is reflected in IP total length field. IPv6 extension headers are ordered. The MINT header MUST be the last extension header in the IPv6 extension header chain. Similarly in case of IPv4 AH/ESP/WESP extension headers, MINT header MUST be the last extension header.

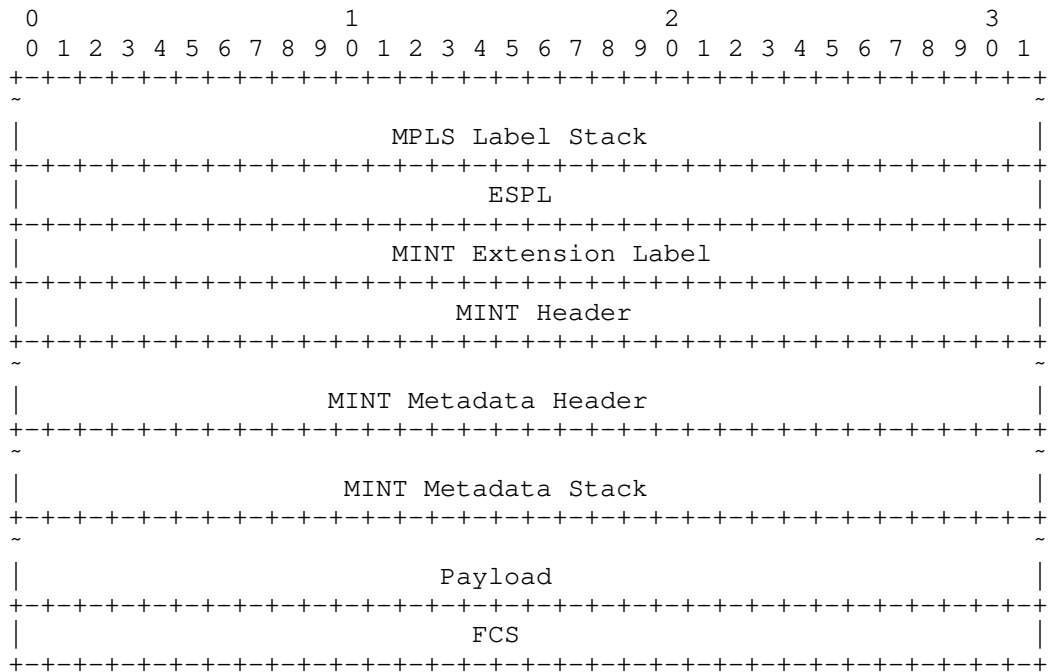


Figure 8 MINT Packet Format

3.8.1 MINT Packet Format with TS Flag Set

In case the Tail Stamp flag is set in the MINT header, the MINT metadata header and metadata stack are inserted at the end of the packet just before the FCS. Each node inserts metadata at the bottom of MINT metadata stack.

One of the key advantages of using TS is to support legacy devices and/or appliances that need to look at the layer 5 data. The IP length and IP header checksum are updated at each hop inserting metadata. This is the same as without the TS flag.

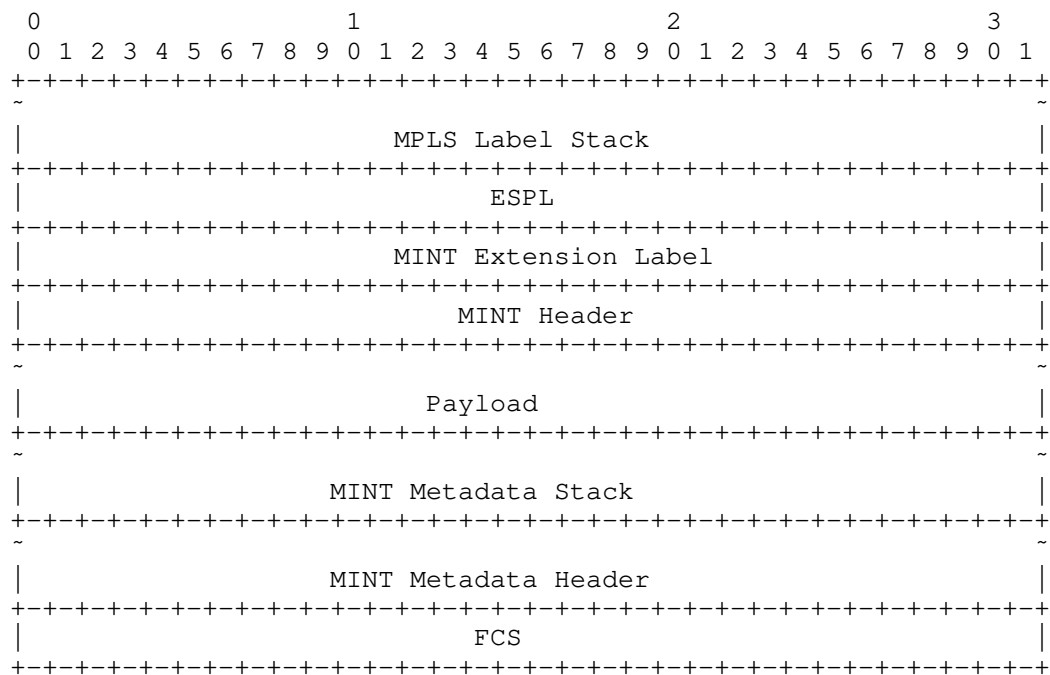


Figure 9 MINT Packet Format with TS

3.9 MINT Load Balancing

As stated in [RFC7325], regarding use of MPLS fields in multipath load balancing:

Special-purpose labels (label values 0-15) MUST NOT be used.
 Extended special-purpose labels (any label following label 15) MUST NOT be used.

Accordingly, the addition of an ESPL and a MINT Extension Label will not cause a different multipath computation from that which is calculated in absence of these labels.

4. IANA Considerations

This document requests the following IANA Actions.

4.1. Extended Special-Purpose MPLS Label Values Registry

This registry defines a new code point to be added to the Extended Special-Purpose MPLS Label Values Registry to identify that a MINT extension label is present.

The following new code point is defined in this draft:

16 MINT Extension Label

5. Security Considerations

A successful attack on an OAM protocol can prevent the detection of failures or anomalies, or create a false illusion of nonexistent ones.

The metadata elements of MINT can be used by attackers to collect information about the network hops.

Adding MINT headers or adding to MINT metadata can be used to consume resources within the path being monitored or by a collector.

Adding MINT headers or adding to MINT metadata can be used to force exceeding the MTU for the path being monitored resulting in fragmentation and/or packet drops.

MINT is expected to be deployed within controlled network domains, containing attacks to that controlled domain. Limiting or preventing monitoring or attacks using IFA requires limiting or preventing unauthorized access to the domain in which MINT is to be used, and preventing leaking IFA metadata beyond the controlled domain.

6. References

6.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2 Informative References

[RFC6790] Kompella, K., "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012, <<http://www.rfc->

editor.org/info/rfc6790>.

[RFC7274] Kompella, K., "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, June 2014, <<http://www.rfc-editor.org/info/rfc7274>>.

[RFC7325] Villamizar, C., "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, August 2014, <<http://www.rfc-editor.org/info/rfc7325>>.

[I-D.draft-kumar-ippm-ifa-01] Kumar, J., "Inband Flow Analyzer", March 2019, <<https://tools.ietf.org/html/draft-kumar-ippm-ifa-01>> (work in progress).

Authors' Addresses

Jai Kumar
Broadcom Inc.
Email: jai.kumar@broadcom.com

John Lemon
Broadcom Inc.
Email: john.lemon@broadcom.com

Yoav Peleg
Broadcom Inc.
Email: yoav.peleg@broadcom.com

Kireeti Kompella
Juniper Networks
Email: kireeti@juniper.net

Network Work group
Internet-Draft
Updates: 8287 (if approved)
Intended status: Standards Track
Expires: July 21, 2019

N. Nainar
C. Pignataro
Cisco Systems, Inc.
F. Iqbal
Individual
A. Vainshtein
ECI Telecom
January 17, 2019

RFC8287 Sub-TLV Length Clarification
draft-nainar-mpls-rfc8287-len-clarification-00

Abstract

RFC8287 defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifier (SIDs) with an MPLS data plane. RFC8287 proposes 3 Target FEC Stack Sub-TLVs. While the standard defines the format and procedure to handle those Sub-TLVs, it does not sufficiently clarify how the length of the Segment ID Sub-TLVs should be computed to include in the Length field of the Sub-TLVs which may result in interoperability issues.

This document updates RFC8287 by clarifying the length of each Segment ID Sub-TLVs defined in RFC8287.

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 21, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Requirements notation	3
4. Length field clarification for Segment ID Sub-TLVs	3
4.1. IPv4 IGP-Prefix Segment ID Sub-TLV	3
4.2. IPv6 IGP-Prefix Segment ID Sub-TLV	3
4.3. IGP-Adjacency Segment ID Sub-TLV	4
5. IANA Considerations	5
6. Security Considerations	5
7. Contributors	5
8. Acknowledgement	5
9. Normative References	5
Authors' Addresses	6

1. Introduction

[RFC8287] defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifier (SIDs) with an MPLS data plane. [RFC8287] proposes 3 Target FEC Stack Sub-TLVs. While the standard defines the format and procedure to handle those Sub-TLVs, it does not sufficiently clarify how the length of the Segment ID Sub-TLVs should be computed to include in the Length field of the Sub-TLVs which may result in interoperability issues.

This document updates [RFC8287] by clarifying the length of each Segment ID Sub-TLVs defined in [RFC8287].

2. Terminology

This document uses the terminologies defined in [RFC8402], [RFC8029], [RFC8287] and so the readers are expected to be familiar with the same.

3. Requirements notation

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].

4. Length field clarification for Segment ID Sub-TLVs

Section 5 of [RFC8287] defines 3 different Segment ID Sub-TLVs that will be included in Target FEC Stack TLV defined in [RFC8029]. The length of each Sub-TLVs MUST be calculated as defined in this section.

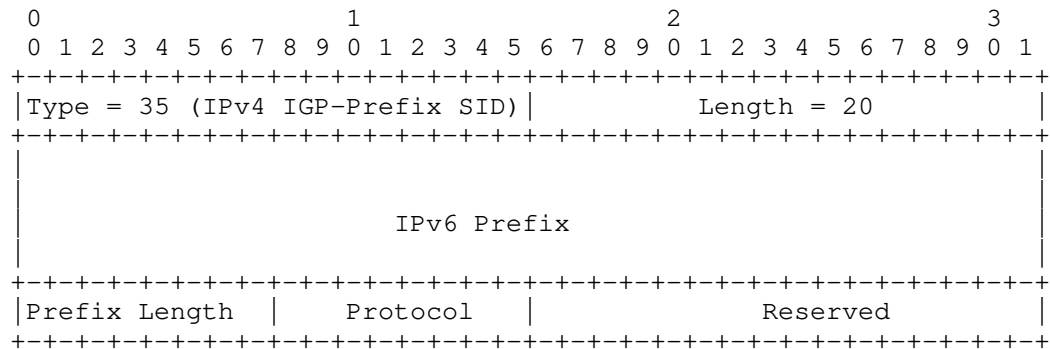
4.1. IPv4 IGP-Prefix Segment ID Sub-TLV

The Sub-TLV length for IPv4 IGP-Prefix Segment ID MUST be set to 8 as shown in the below TLV format:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Type = 34 (IPv4 IGP-Prefix SID)										Length = 8																													
IPv4 prefix																																							
Prefix Length										Protocol										Reserved																			

4.2. IPv6 IGP-Prefix Segment ID Sub-TLV

The Sub-TLV length for IPv6 IGP-Prefix Segment ID MUST be set to 20 as shown in the below TLV format:



4.3. IGP-Adjacency Segment ID Sub-TLV

The Sub-TLV length for IGP-Adjacency Segment ID varies depending on the Adjacency Type and Protocol. In any of the allowed combination of Adjacency Type and Protocol, the sub-TLV length MUST be calculated by including 2 octets of Reserved field. Below is a table that list the length for different combinations.

Protocol	Length for Adj.Type		
	Parallel	IPv4	IPv6
OSPF	20	20	44
ISIS	24	24	48
Any	20	20	44

For example, when the Adj. Type is set to Parallel Adjacency and the Protocol is set to 0, the Sub-TLV will be as below:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type = 36 (IGP-Adjacency SID) | Length = 20 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Adj. Type = 1 | Protocol = 0 | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Local Interface ID (4 octets) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Remote Interface ID (4 octets) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Advertising Node Identifier (4 octets) |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Receiving Node Identifier (4 octets) |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5. IANA Considerations

This document does not introduce any IANA consideration.

6. Security Considerations

This document updates [RFC8287] and does not introduce any security considerations.

7. Contributors

The below individuals contributed to this document:

Zafar Ali, Cisco Systems, Inc.

8. Acknowledgement

The authors would like to thank Michael Gorokhovsky and Manohar Doppalapudi for investigating the interop issue during EANTC test

9. 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>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.

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

Authors' Addresses

Nagendra Kumar Nainar
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

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

Email: cpignata@cisco.com

Faisal Iqbal
Individual
Canada

Email: faisal.iqbal@msn.com

Alexander Vainshtein
ECI Telecom
Israel

Email: vainshtein.alex@gmail.com

MPLS Working Group
Internet-Draft
Intended status: Informational
Expires: September 8, 2019

L. Andersson
Bronze Dragon Consulting
S. Bryant
A. Malis
Huawei Technologies
N. Leymann
Deutsche Telekom
G. Swallow
Independent
March 7, 2019

Deprecating MD5 for LDP
draft-nslag-mpls-deprecate-md5-04

Abstract

When the MPLS Label Distribution Protocol (LDP) was specified circa 1999, there were very strong requirements that LDP should use a cryptographic hash function to sign LDP protocol messages. MD5 was widely used at that time, and was the obvious choices.

However, even when this decision was being taken there were concerns as to whether MD5 was a strong enough signing option. This discussion was briefly reflected in section 5.1 of RFC 5036 [RFC5036] (and also in RFC 3036 [RFC3036]).

Over time it has been shown that MD5 can be compromised. Thus, there is a concern shared in the security community and the working groups responsible for the development of the LDP protocol that LDP is no longer adequately secured.

This document deprecates MD5 as the signing method for LDP messages. The document also selects a future method to secure LDP messages - the choice is TCP-AO. In addition, we specify that the TBD cryptographic mechanism is to be the default TCP-AO security method.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 8, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
1.1. Requirement Language	3
2. Background	3
2.1. LDP in RFC 5036	3
2.2. MD5 in BGP	3
2.3. Prior Art	4
3. Securing LDP	4
4. Security Considerations	5
5. IANA Considerations	5
6. Acknowledgements	5
7. References	5
7.1. Normative References	5
7.2. Informative References	6
Authors' Addresses	6

1. Introduction

RFC 3036 was published in January 2001 as a Proposed Standard, and it was replaced by RFC 5035, which is a Draft Standard, in October 2007. Two decades after LDP was originally specified there is a concern shared by the security community and the IETF working groups that develop the LDP protocol that LDP is no longer adequately secured.

LDP currently uses MD5 to cryptographically sign its messages for security security purposes. However, MD5 is a hash function that is no longer considered adequate to meet current security requirements.

1.1. 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. Background

2.1. LDP in RFC 5036

In Section 5.1 "Spoofing" of RFC 5036 [RFC5036], in list item 2 "Session communication carried by TCP" the following statements are made:

LDP specifies use of the TCP MD5 Signature Option to provide for the authenticity and integrity of session messages.

RFC 2385 [RFC2385] asserts that MD5 authentication is now considered by some to be too weak for this application. It also points out that a similar TCP option with a stronger hashing algorithm (it cites SHA-1 as an example) could be deployed. To our knowledge, no such TCP option has been defined and deployed. However, we note that LDP can use whatever TCP message digest techniques are available, and when one stronger than MD5 is specified and implemented, upgrading LDP to use it would be relatively straightforward.

2.2. MD5 in BGP

There has been a similar discussion among working groups developing the BGP protocol. BGP has already replaced MD5 with TCP-AO. This was specified in RFC 7454 [RFC7454].

To secure LDP the same approach will be followed, TCP-AO will be used for LDP also.

As far as we are able to ascertain, there is currently no recommended, mandatory to implement, cryptographic function specified. We are concerned that without such a mandatory function, implementations will simply fall back to MD5 and nothing will really be changed. The MPLS working group will need the expertise of the

security community to specify a viable security function that is suitable for wide scale deployment on existing network platforms.

2.3. Prior Art

RFC 6952 [RFC6952] discusses a set of routing protocols that all are using TCP for transport of protocol messages, according to guidelines set forth in Section 4.2 of "Keying and Authentication for Routing Protocols Design Guidelines", RFC 6518 [RFC6518].

RFC 6952 takes a much broader approach than this document, it discusses several protocols and also securing the LDP session initialization. This document has a narrower scope, securing LDP session messages only. LDP in initialization mode is addressed in RFC 7349 [RFC7349].

RFC 6952 and this document, basically suggest the same thing, move to TCP-AO and deploy a strong cryptographic algorithm.

All the protocols discussed in RFC 6952 should adopt the approach to securing protocol messages over TCP.

3. Securing LDP

Implementations conforming to this RFC MUST implement TCP-AO to secure the TCP sessions carrying LDP in addition to the currently required TCP MD5 Signature Option.

A TBD cryptographic mechanism must be implemented and provided to TCP-AO to secure LDP messages.

The TBD mechanism is the preferred option, and MD5 SHOULD only to be used when TBD is unavailable.

Note: The authors are not experts on this part of the stack, but it seems that TCP security negotiation is still work in progress. If we are wrong, then we need to include a requirement that such negotiation is also required. In the absence of a negotiation protocol, however, we need to leave this as a configuration process until such time as the negotiation protocol work is complete. On completion of a suitable negotiation protocol we need to issue a further update requiring its use.

Cryptographic mechanisms do not have an indefinite lifetime, the IETF hence anticipates updating default cryptographic mechanisms over time.

The TBD default security function will need to be chosen such that it can reasonably be implemented on a typical router route processor, and which will provide adequate security without significantly degrading the convergence time of a Label Switching Router (LSR).

Without a function that does not significantly impact router convergence we simply close one vulnerability and open another.

Note: As experts on the LDP protocol, but not on security mechanisms, we need to ask the security area for a review of our proposed approach, and help correcting any misunderstanding of the security issues or our misunderstanding of the existing security mechanisms. We also need a recommendation on a suitable security function (TBD in the above text).

4. Security Considerations

This document is entirely about LDP operational security. It describes best practices that one should adopt to secure LDP messages and the TCP based LDP sessions between LSRs.

This document does not aim to describe existing LDP implementations, their potential vulnerabilities, or ways they handle errors. It does not detail how protection could be enforced against attack techniques using crafted packets.

5. IANA Considerations

There are no requests for IANA actions in this document.

Note to the RFC Editor - this section can be removed before publication.

6. Acknowledgements

-

-

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>>.

- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <<https://www.rfc-editor.org/info/rfc2385>>.
- [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>>.
- [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

- [RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, DOI 10.17487/RFC3036, January 2001, <<https://www.rfc-editor.org/info/rfc3036>>.
- [RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, DOI 10.17487/RFC6518, February 2012, <<https://www.rfc-editor.org/info/rfc6518>>.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <<https://www.rfc-editor.org/info/rfc6952>>.
- [RFC7349] Zheng, L., Chen, M., and M. Bhatia, "LDP Hello Cryptographic Authentication", RFC 7349, DOI 10.17487/RFC7349, August 2014, <<https://www.rfc-editor.org/info/rfc7349>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.

Authors' Addresses

Loa Andersson
Bronze Dragon Consulting

Email: loa@pi.nu

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Nicolai Leymann
Deutsche Telekom

Email: N.Leymann@telekom.de

George Swallow
Independent

Email: swallow.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 13, 2019

L. Zheng
G. Zheng
Huawei Technologies
G. Mirsky
ZTE Corp.
R. Rahman
F. Iqbal
Cisco Systems
January 9, 2019

YANG Data Model for LSP-Ping
draft-zheng-mpls-lsp-ping-yang-cfg-10

Abstract

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. RFC 8029 defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in RFC 6020 and RFC 7950, is a data modeling language used to specify the contents of a conceptual data stores that allows networked devices to be managed using NETCONF, as specified in RFC 6241. This document defines a YANG data model that can be used to configure and manage LSP-Ping.

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 13, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Support of Long Running Command with NETCONF	3
2. Scope	3
3. Design of the Data Model	4
3.1. The Configuration of Control Information	4
3.2. The Configuration of Schedule Parameters	5
3.3. Display of Result Information	6
4. Data Hierarchy	7
5. Interaction with other MPLS OAM Tools Models	9
6. LSP-Ping YANG Module	10
7. Examples	21
7.1. Configuration of Control Information	21
7.2. The Configuration of Schedule Parameters	22
7.3. Display of Result Information	23
8. Security Considerations	25
9. IANA Considerations	26
Contributors	26
Acknowledgments	27
12. References	27
12.1. Normative References	27
12.2. Informative References	27
Authors' Addresses	28

1. Introduction

When an LSP fails to deliver user traffic, the failure cannot always be detected by the MPLS control plane. [RFC8029] defines a mechanism that would enable users to detect such failure and to isolate faults. YANG, defined in [RFC6020] and [RFC7950], is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. This document defines a YANG data model that can be used to configure and manage LSP-Ping [RFC8029].

The rest of this document is organized as follows. Section 2 presents the scope of this document. Section 3 provides the design of the LSP-Ping configuration data model in details by containers. Section 4 presents the complete data hierarchy of LSP-Ping YANG model. Section 5 discusses the interaction between LSP-Ping data model and other MPLS tools data models. Section 6 specifies the YANG module and section 7 lists examples which conform to the YANG module specified in this document. Finally, security considerations are discussed in Section 8.

This version of the LSP Ping data model conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

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.

1.2. Support of Long Running Command with NETCONF

LSP Ping is one of the examples of what can be described as "long-running operation". Unlike most of the configuration operations that result in single response execution of an LSP Ping triggers multiple responses from a node under control. The question of implementing the long-running operation in NETCONF is still open and possible solutions being discussed:

1. Consecutive Remote Processing Calls (RPC) to poll for results.
2. Model presented in [RFC4560].
3. The one outlined in [I-D.mahesh-netconf-persistent].

The problem of long-running operation as well can be considered as a case of controlling and obtaining results from a Measurement Agent (MA) as defined in [RFC7594].

2. Scope

The fundamental mechanism of LSP-Ping is defined in [RFC8029]. Extensions of LSP-Ping has been developed over the years. There are extensions for performing LSP ping, for example, over P2MP MPLS LSPs [RFC6425] or for Segment Routing IGP Prefix and Adjacency SIDs with an MPLS data plane [RFC8287]. These extensions will be considered in a later update of this document.

3. Design of the Data Model

This YANG data model is defined to be used to configure and manage LSP-Ping and it provides the following features:

1. The configuration of control information of an LSP-Ping test.
2. The configuration of schedule parameters of an LSP-Ping test.
3. Display of result information of an LSP-Ping test.

The top-level container `lsp-pings` holds the configuration of the control information, schedule parameters and result information for multiple instances of LSP-Ping test.

3.1. The Configuration of Control Information

Container `lsp-pings:lsp-ping:control-parameters` defines the configuration parameters which control an LSP-Ping test. Examples are the `target-fec-type/target-fec` of the echo request packet and the `reply mode` of the echo reply packet. Values of some parameters may be auto-assigned by the system, but in several cases, there is a requirement for configuration of these parameters. Examples of such parameters are source address and outgoing interface.

The data hierarchy for control information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
        +--rw target-fec-type?      target-fec-type
        +--rw (target-fec)?
          +--:(ip-prefix)
            +--rw ip-address?        inet:ip-address
          +--:(bgp)
            +--rw bgp?                inet:ip-address
          +--:(rsvp)
            +--rw tunnel-interface?  string
          +--:(vpn)
            +--rw vrf-name?           uint32
            +--rw vpn-ip-address?     inet:ip-address
          +--:(pw)
            +--rw vcid?               uint32
          +--:(vpls)
            +--rw vsi-name?           string
        +--rw traffic-class?        uint8
        +--rw reply-mode?            reply-mode
        +--rw timeout?               uint32
        +--rw timeout-units?         units
        +--rw interval?              uint32
        +--rw interval-units?        units
        +--rw probe-count?           uint32
        +--rw data-size?              uint32
        +--rw data-fill?              string
        +--rw description?            string
        +--rw source-address?         inet:ip-address
        +--rw ttl?                    uint8
        +--rw (outbound)?
          +--:(interface)
            +--rw interface-name?     string
          +--:(nexthop)
            +--rw nexthop?             inet:ip-address

```

3.2. The Configuration of Schedule Parameters

Container `lsp-pings:lsp-ping:scheduling-parameters` defines the schedule parameters of an LSP-Ping test, which describes when to start and when to end the test. Four start modes and three end modes are defined respectively. To be noted that, the configuration of "interval" and "probe-count" parameter defined in container `lsp-pings:lsp-ping:control-parameters` could also determine when the test ends implicitly. All these three parameters are optional. If the user

does not configure either "interval" or "probe-count" parameter, then the default values will be used by the system. If the user configures "end-test", then the actual end time of the LSP-Ping test is the smaller one between the configuration value of "end-test" and the time implicitly determined by the configuration value of "interval"/"probe-count".

The data hierarchy for schedule information configuration is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
        +--rw (start-test)?
          +--:(now)
            | +--rw start-test-now?          empty
          +--:(at)
            | +--rw start-test-at?           yang:date-and-time
          +--:(delay)
            | +--rw start-test-delay?        uint32
            | +--rw start-test-delay-units?  units
          +--:(daily)
            | +--rw start-test-daily?        yang:date-and-time
        +--rw (end-test)?
          +--:(at)
            | +--rw end-test-at?             yang:date-and-time
          +--:(delay)
            | +--rw end-test-delay?          uint32
            | +--rw end-test-delay-units?    units
          +--:(lifetime)
            | +--rw end-test-lifetime?       uint32
            | +--rw lifetime-units?         units

```

3.3. Display of Result Information

Container `lsp-pings:lsp-ping:result-info` shows the result of the current LSP-Ping test. Both the statistical result e.g. `min-rtt`, `max-rtt`, and per test probe result e.g. `return code`, `return subcode`, are shown.

The data hierarchy for display of result information is presented below:

```

module: ietf-lsp-ping
  +--rw lsp-pings
    +--rw lsp-ping* [lsp-ping-name]
      +--rw lsp-ping-name          string
      +--rw control-parameters
      ...
      +--rw scheduling-parameters
      ...
      +--ro result-info
        +--ro operational-status?    operational-status
        +--ro source-address?        inet:ip-address
        +--ro target-fec-type?       target-fec-type
        +--ro (target-fec)?
          +--:(ip-prefix)
            | +--ro ip-address?       inet:ip-address
          +--:(bgp)
            | +--ro bgp?              inet:ip-address
          +--:(rsvp)
            | +--ro tunnel-interface? string
          +--:(vpn)
            | +--ro vrf-name?         uint32
            | +--ro vpn-ip-address?   inet:ip-address
          +--:(pw)
            | +--ro vcid?             uint32
          +--:(vpls)
            | +--ro vsi-name?         string
        +--ro min-rtt?               uint32
        +--ro max-rtt?               uint32
        +--ro average-rtt?           uint32
        +--ro probe-responses?       uint32
        +--ro sent-probes?           uint32
        +--ro sum-of-squares?        uint32
        +--ro last-good-probe?       yang:date-and-time
        +--ro probe-results
          +--ro probe-result* [probe-index]
            +--ro probe-index         uint32
            +--ro return-code?        uint8
            +--ro return-sub-code?    uint8
            +--ro rtt?               uint32
            +--ro result-type?        result-type

```

4. Data Hierarchy

The complete data hierarchy of LSP-Ping YANG model is presented below.

```

module: ietf-lsp-ping

```

```

+--rw lsp-pings
  +--rw lsp-ping* [lsp-ping-name]
    +--rw lsp-ping-name          string
    +--rw control-parameters
      +--rw target-fec-type?      target-fec-type
      +--rw (target-fec)?
        +--:(ip-prefix)
          +--rw ip-address?      inet:ip-address
        +--:(bgp)
          +--rw bgp?             inet:ip-address
        +--:(rsvp)
          +--rw tunnel-interface? string
        +--:(vpn)
          +--rw vrf-name?        uint32
          +--rw vpn-ip-address?  inet:ip-address
        +--:(pw)
          +--rw vcid?            uint32
        +--:(vpls)
          +--rw vsi-name?        string
      +--rw traffic-class?        uint8
      +--rw reply-mode?           reply-mode
      +--rw timeout?              uint32
      +--rw timeout-units?        units
      +--rw interval?            uint32
      +--rw interval-units?       units
      +--rw probe-count?          uint32
      +--rw data-size?            uint32
      +--rw data-fill?            string
      +--rw description?          string
      +--rw source-address?       inet:ip-address
      +--rw ttl?                  uint8
      +--rw (outbound)?
        +--:(interface)
          +--rw interface-name?   string
        +--:(nexthop)
          +--rw nexthop?          inet:ip-address
    +--rw scheduling-parameters
      +--rw (start-test)?
        +--:(now)
          +--rw start-test-now?    empty
        +--:(at)
          +--rw start-test-at?     yang:date-and-time
        +--:(delay)
          +--rw start-test-delay?   uint32
          +--rw start-test-delay-units? units
        +--:(daily)
          +--rw start-test-daily?   yang:date-and-time
      +--rw (end-test)?

```

```

    +---:(at)
    |   +---rw end-test-at?                yang:date-and-time
    +---:(delay)
    |   +---rw end-test-delay?             uint32
    |   +---rw end-test-delay-units?       units
    +---:(lifetime)
    |   +---rw end-test-lifetime?          uint32
    |   +---rw lifetime-units?             units
+---ro result-info
+---ro operational-status?                operational-status
+---ro source-address?                   inet:ip-address
+---ro target-fec-type?                  target-fec-type
+---ro (target-fec)?
|   +---:(ip-prefix)
|   |   +---ro ip-address?                inet:ip-address
+---:(bgp)
|   +---ro bgp?                          inet:ip-address
+---:(rsvp)
|   +---ro tunnel-interface?              string
+---:(vpn)
|   +---ro vrf-name?                      uint32
|   +---ro vpn-ip-address?                inet:ip-address
+---:(pw)
|   +---ro vcid?                          uint32
+---:(vpls)
|   +---ro vsi-name?                      string
+---ro min-rtt?                          uint32
+---ro max-rtt?                          uint32
+---ro average-rtt?                      uint32
+---ro probe-responses?                  uint32
+---ro sent-probes?                      uint32
+---ro sum-of-squares?                   uint32
+---ro last-good-probe?                  yang:date-and-time
+---ro probe-results
    +---ro probe-result* [probe-index]
        +---ro probe-index                uint32
        +---ro return-code?               uint8
        +---ro return-sub-code?           uint8
        +---ro rtt?                       uint32
        +---ro result-type?               result-type

```

5. Interaction with other MPLS OAM Tools Models

TBA

6. LSP-Ping YANG Module

```
<CODE BEGINS> file "ietf-lsp-ping@2018-11-29.yang"
module ietf-lsp-ping {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-lsp-ping";
  //namespace need to be assigned by IANA
  prefix "lsp-ping";

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types{
    prefix yang;
    reference "RFC 6991: Common YANG Types.";
  }

  organization "IETF Multiprotocol Label Switching Working Group";

  contact
    "WG Web: http://tools.ietf.org/wg/mppls/
    WG List: mppls@ietf.org

    Editor: Greg Mirsky
      gregimirsky@gmail.com
    Editor: Lianshu Zheng
      vero.zheng@huawei.com
    Editor: Guangying Zheng
      zhengguangying@huawei.com
    Editor: Reshad Rahman
      rrahman@cisco.com
    Editor: Faisal Iqbal
      faiqbal@cisco.com";

  description
    "This YANG module specifies a vendor-independent model
    for the LSP Ping.

    This YANG data model is defined to be used to configure and manage
    LSP-Ping and it provides the following features:
    1. The configuration of control information of an LSP-Ping test.
    2. The configuration of schedule parameters of an LSP-Ping test.
    3. Display of result information of an LSP-Ping test.

    Copyright (c) 2018 IETF Trust and the persons identified as
    the document authors. All rights reserved.
    Redistribution and use in source and binary forms, with or
```


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

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

```
reference "draft-zheng-mpls-lsp-ping-yang-cfg";

revision "2018-11-29" {
  description
    "10 version, refine the target fec type,
    as per RFC8029 and update Security Considerations section.";
  reference "draft-zheng-mpls-lsp-ping-yang-cfg";
}

typedef target-fec-type {
  type enumeration {
    enum ip-prefix {
      value "0";
      description "IPv4/IPv6 prefix";
    }
    enum bgp {
      value "1";
      description "BGP IPv4/IPv6 prefix";
    }
    enum rsvp {
      value "2";
      description "Tunnel interface";
    }
    enum vpn {
      value "3";
      description "VPN IPv4/IPv6 prefix";
    }
    enum pw {
      value "4";
      description "FEC 128 pseudowire IPv4/IPv6";
    }
    enum vpls {
      value "5";
      description "FEC 129 pseudowire IPv4/IPv6";
    }
  }
  description "Target FEC type, as defined in RFC 8029";
}
```

```
typedef reply-mode {
  type enumeration {
    enum do-not-reply {
      value "1";
      description "Do not reply";
    }
    enum reply-via-udp {
      value "2";
      description "Reply via an IPv4/IPv6 UDP packet";
    }
    enum reply-via-udp-router-alert {
      value "3";
      description
        "Reply via an IPv4/IPv6 UDP packet with Router Alert";
    }
    enum reply-via-control-channel {
      value "4";
      description
        "Reply via application level control channel";
    }
  }
  description "Reply mode";
}

typedef units {
  type enumeration {
    enum seconds {
      description "Seconds";
    }
    enum milliseconds {
      description "Milliseconds";
    }
    enum microseconds {
      description "Microseconds";
    }
    enum nanoseconds {
      description "Nanoseconds";
    }
  }
  description "Time units";
}

typedef operational-status {
  type enumeration {
    enum enabled {
      value "1";
      description "The Test is active";
    }
  }
}
```

```
    enum disabled {
        value "2";
        description "The test has stopped";
    }
    enum completed {
        value "3";
        description "The test is completed";
    }
}
description "Operational state of an LSP Ping test";
}

typedef result-type {
    type enumeration {
        enum success {
            value "1";
            description "The test probe is successful";
        }
        enum fail {
            value "2";
            description "The test probe has failed";
        }
        enum timeout {
            value "3";
            description "The time of the test probe has expired";
        }
    }
    description "Result of each LSP Ping test probe";
}

container lsp-pings {
    description "Multi-instance of the LSP Ping test";
    list lsp-ping {
        key "lsp-ping-name";
        description "LSP Ping test";
        leaf lsp-ping-name {
            type string {
                length "1..31";
            }
            mandatory "true";
            description "LSP Ping test name";
        }
        container control-parameters {
            description "Control information of the LSP Ping test";
            leaf target-fec-type {
                type target-fec-type;
                description "Specifies the address type of the Target FEC";
            }
        }
    }
}
```

```
choice target-fec {
  case ip-prefix {
    leaf ip-address {
      type inet:ip-address;
      description "IPv4/IPv6 Prefix";
    }
  }
  case bgp {
    leaf bgp {
      type inet:ip-address;
      description "BGP IPv4/IPv6 Prefix";
    }
  }
  case rsvp {
    leaf tunnel-interface {
      type string;
      description "Tunnel interface";
    }
  }
  case vpn {
    leaf vrf-name {
      type uint32;
      description "Layer3 VPN Name";
    }
    leaf vpn-ip-address {
      type inet:ip-address;
      description "Layer3 VPN IPv4 Prefix";
    }
  }
  case pw {
    leaf vcid {
      type uint32;
      description "VC ID";
    }
  }
  case vpls {
    leaf vsi-name {
      type string;
      description "VPLS VSI";
    }
  }
  description "Specifies the type of the Target FEC";
}
leaf traffic-class {
  type uint8;
  description "Specifies the Traffic Class";
}
leaf reply-mode {
```

```
    type reply-mode;
    description "Specifies the Reply Mode";
  }
  leaf timeout {
    type uint32;
    description
      "Specifies the time-out value for a LSP Ping operation.";
  }
  leaf timeout-units {
    type units;
    description "Time-out units";
  }
  leaf interval {
    type uint32;
    default 1;
    description
      "Specifies the interval between transmissions
       of LSP Ping echo request packets (probes)
       as part of the LSP Ping test.";
  }
  leaf interval-units {
    type units;
    default seconds;
    description "Interval units";
  }
  leaf probe-count {
    type uint32;
    default 5;
    description
      "Specifies the number of probes sent in the LSP Ping test.";
  }
  leaf data-size {
    type uint32;
    description
      "Specifies the size of the data portion to
       be transmitted in an LSP Ping operation, in octets.";
  }
  leaf data-fill {
    type string{
      length "0..1564";
    }
    description
      "Used together with the corresponding
       data-size value to determine how to fill the data
       portion of a probe packet.";
  }
  leaf description {
    type string{
```

```
        length "1..31";
    }
    description "A descriptive name of the LSP Ping test";
}
leaf source-address {
    type inet:ip-address;
    description "Specifies the source address";
}
leaf ttl {
    type uint8;
    default 255;
    description "Time to live";
}
choice outbound {
    case interface {
        leaf interface-name {
            type string {
                length "1..255";
            }
            description "Specifies the outgoing interface";
        }
    }
    case nexthop {
        leaf nexthop {
            type inet:ip-address;
            description "Specifies the nexthop";
        }
    }
    description "Specifies the out interface or nexthop";
}
}

container scheduling-parameters {
    description "LSP Ping test schedule parameter";
    choice start-test {
        case now {
            leaf start-test-now {
                type empty;
                description "Start test now";
            }
        }
        case at {
            leaf start-test-at {
                type yang:date-and-time;
                description "Start test at a specific time";
            }
        }
        case delay {
```

```
    leaf start-test-delay {
        type uint32;
        description "Start after a specific delay";
    }
    leaf start-test-delay-units {
        type units;
        default seconds;
        description "Delay units";
    }
}
case daily {
    leaf start-test-daily {
        type yang:date-and-time;
        description "Start test daily";
    }
}
description
    "Specifies when the test begins to start,
    include 4 schedule method: start now(1), start at(2),
    start delay(3), start daily(4).";
}

choice end-test{
    case at {
        leaf end-test-at{
            type yang:date-and-time;
            description "End test at a specific time";
        }
    }
    case delay {
        leaf end-test-delay {
            type uint32;
            description "End after a specific delay";
        }
        leaf end-test-delay-units {
            type units;
            default seconds;
            description "Delay units";
        }
    }
}
case lifetime {
    leaf end-test-lifetime {
        type uint32;
        description "Set the test lifetime";
    }
    leaf lifetime-units {
        type units;
        default seconds;
    }
}
```

```
        description "Lifetime units";
    }
}
description
    "Specifies when the test ends, include 3
    schedule method: end at(1), end delay(2),
    end lifetime(3).";
}
}

container result-info {
    config "false";
    description "LSP Ping test result information";
    leaf operational-status {
        type operational-status;
        description "Operational state of a LSP Ping test";
    }
    leaf source-address {
        type inet:ip-address;
        description "The source address of the test";
    }
    leaf target-fec-type {
        type target-fec-type;
        description "The Target FEC address type";
    }
    choice target-fec {
        case ip-prefix {
            leaf ip-address {
                type inet:ip-address;
                description "IPv4/IPv6 Prefix";
            }
        }
        case bgp {
            leaf bgp {
                type inet:ip-address;
                description "BGP IPv4/IPv6 Prefix";
            }
        }
        case rsvp {
            leaf tunnel-interface {
                type string;
                description "Tunnel interface";
            }
        }
        case vpn {
            leaf vrf-name {
                type uint32;
                description "Layer3 VPN Name";
            }
        }
    }
}
```



```
    }
    leaf vpn-ip-address {
        type inet:ip-address;
        description "Layer3 VPN IPv4 Prefix";
    }
}
case pw {
    leaf vcid {
        type uint32;
        description "VC ID";
    }
}
case vpls {
    leaf vsi-name {
        type string;
        description "VPLS VSI";
    }
}
description "The Target FEC address";
}
leaf min-rtt {
    type uint32;
    description
        "The minimum LSP Ping round-trip-time (RTT)
        received measured in usec.";
}
leaf max-rtt {
    type uint32;
    description
        "The maximum LSP Ping round-trip-time (RTT)
        received measured in usec.";
}
leaf average-rtt {
    type uint32;
    description
        "The current average LSP Ping round-trip-time
        (RTT) measured in usec.";
}
leaf probe-responses {
    type uint32;
    description
        "Number of responses received for the
        corresponding LSP Ping test.";
}
leaf sent-probes {
    type uint32;
    description
        "Number of probes sent for the
```

```
        corresponding LSP Ping test.";
    }
    leaf sum-of-squares {
        type uint32;
        description
            "The sum of the squares of RTT,
            calculated as the sum of the squared
            differences between each RTT and the overall
            mean RTT, for all replies received.";
    }
    leaf last-good-probe {
        type yang:date-and-time;
        description
            "Date and time when the last response
            was received for a probe.";
    }
}

container probe-results {
    description "Result info of test probes";
    list probe-result {
        key "probe-index";
        description "Result info of each test probe";
        leaf probe-index {
            type uint32;
            config false;
            description "Probe index";
        }
        leaf return-code {
            type uint8;
            config false;
            description "The Return Code set in the echo reply";
        }
        leaf return-sub-code {
            type uint8;
            config false;
            description
                "The Return Sub-code set in the echo reply.";
        }
        leaf rtt {
            type uint32;
            config false;
            description "The round-trip-time (RTT) received";
        }
        leaf result-type {
            type result-type;
            config false;
            description "The probe result type";
        }
    }
}
```

```
    }  
  }  
}  
}  
}  
<CODE ENDS>
```

7. Examples

The following examples show the netconf RPC communication between client and server for one LSP-Ping test case.

7.1. Configuration of Control Information

Configure the control-parameters for sample-test-case.

Request from netconf client:

```
<rpc
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <control-parameters>
            <target-fec-type>ip-prefix</target-fec-type>
            <ip-prefix>2001:db8::1:100/64</ip-prefix>
            <reply-mode>reply-via-udp</reply-mode>
            <timeout>1</timeout>
            <timeout-units>seconds</timeout-units>
            <interval>1</interval>
            <interval-units>seconds</interval-units>
            <probe-count>6</probe-count>
            <admin-status>enabled</admin-status>
            <data-size>64</data-size>
            <data-fill>this is a lsp ping test</data-fill>
            <source-address>2001:db8::4</source-address>
            <ttl>56</ttl>
          </control-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.2. The Configuration of Schedule Parameters

Set the scheduling-parameters for sample-test-case to start the test.

Request from netconf client:

```
<rpc
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <scheduling-parameters>
            <start-test-now/>
          </scheduling-parameters>
        </lsp-ping>
      </lsp-pings>
    </config>
  </edit-config>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="102" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

7.3. Display of Result Information

Get the result-info of sample-test-case.

Request from netconf client:

```
<rpc
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
        <lsp-ping>
          <lsp-ping-name>sample-test-case</lsp-ping-name>
          <result-info/>
        </lsp-ping>
      </lsp-pings>
    </filter>
  </get>
</rpc>
```

Reply from netconf server:

```
<rpc-reply
  message-id="103" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
<data>
  <lsp-pings xmlns="urn:ietf:params:xml:ns:yang:ietf-lsp-ping">
    <lsp-ping>
      <lsp-ping-name>sample-test-case</lsp-ping-name>
      <result-info>
        <operational-status>completed</operational-status>
        <source-address>2001:db8::4</source-address>
        <target-fec-type>ip-prefix</target-fec-type>
        <ip-prefix>2001:db8::1:100/64</ip-prefix>
        <min-rtt>10</min-rtt>
        <max-rtt>56</max-rtt>
        <average-rtt>36</average-rtt>
        <probe-responses>6</probe-responses>
        <sent-probes>6</sent-probes>
        <sum-of-squares>8882</sum-of-squares>
        <last-good-probe>2015-07-01T10:36:56</last-good-probe>
        <probe-results>
          <probe-result>
            <probe-index>0</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>10</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>1</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>56</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>2</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>35</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>3</probe-index>
            <return-code>0</return-code>
            <return-sub-code>3</return-sub-code>
            <rtt>38</rtt>
            <result-type>success</result-type>
          </probe-result>
          <probe-result>
            <probe-index>4</probe-index>
            <return-code>0</return-code>
```

```

        <return-sub-code>3</return-sub-code>
        <rtt>36</rtt>
        <result-type>success</result-type>
    </probe-result>
    <probe-result>
        <probe-index>5</probe-index>
        <return-code>0</return-code>
        <return-sub-code>3</return-sub-code>
        <rtt>41</rtt>
        <result-type>success</result-type>
    </probe-result>
</probe-results>
</result-info>
</lsp-ping>
</lsp-pings>
</data>
</rpc-reply>

```

8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

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

TBD

Unauthorized access to any data node of these subtrees can adversely affect the routing subsystem of both the local device and the network. This may lead to corruption of the measurement that may result in false corrective action, e.g., false negative or false positive. That could be, for example, prolonged and undetected

deterioration of the quality of service or actions to improve the quality unwarranted by the real network conditions.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

TBD

Unauthorized access to any data node of these subtrees can disclose the operational state information of VRRP on this device.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

TBD

The LSP ping YANG module inherits all security consideration of [RFC8029].

9. IANA Considerations

The IANA is requested to assign a new namespace URI from the IETF XML registry.

URI:TBA

Contributors

Yanfeng Zhang

Huawei Technologies

zhangyanfeng@huawei.com

Sam Aldrin

Google

aldrin.ietf@gmail.com

Acknowledgments

TBD

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>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [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>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

12.2. Informative References

- [I-D.mahesh-netconf-persistent] Jethanandani, M., "NETCONF and persistent responses", draft-mahesh-netconf-persistent-00 (work in progress), October 2014.
- [RFC4560] Quittek, J., Ed. and K. White, Ed., "Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations", RFC 4560, DOI 10.17487/RFC4560, June 2006, <<https://www.rfc-editor.org/info/rfc4560>>.
- [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>>.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

Authors' Addresses

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Guangying Zheng
Huawei Technologies
China

Email: zhengguangying@huawei.com

Greg Mirsky
ZTE Corp.
USA

Email: gregimirsky@gmail.com

Reshad Rahman
Cisco Systems
Canada

Email: rrahman@cisco.com

Faisal Iqbal
Cisco Systems

Email: faiqbal@cisco.com