MPLS
Internet-Draft
Updates: 6790 (if approved)
Intended status: Standards Track
Expires: August 26, 2021

B. Decraene, Ed.
Orange
C. Filsfils
Cisco Systems, Inc.
W. Henderickx
Nokia
T. Saad
V. Beeram
Juniper Networks
L. Jalil
Verizon

February 22, 2021

Using Entropy Label for Network Slice Identification in MPLS networks. draft-decraene-mpls-slid-encoded-entropy-label-id-01

Abstract

This document defines a solution to encode a slice identifier in MPLS in order to distinguish packets that belong to different slices, to allow enforcing per network slice policies (.e.g, Qos).

The slice identification is independent of the topology. It allows for QoS/DiffServ policy on a per slice basis in addition to the per packet QoS/DiffServ policy provided by the MPLS Traffic Class field.

In order to minimize the size of the MPLS stack and to ease incremental deployment the slice identifier is encoded as part of the Entropy Label.

This document also extends the use of the TTL field of the Entropy Label in order to provide a flexible set of flags called the Entropy Label Control field.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] leverages the source-routing paradigm. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header. In the SR-MPLS data plane [RFC8660], the SR header is instantiated through a label stack.

This document defines a solution to encode a slice identifier in MPLS in order to provide QoS on a per slice basis. It allows for QoS/

DiffServ policy on a per slice basis in addition to the per packet QoS/DiffServ policy provided by the MPLS Traffic Class field. The slice identification is independent of the topology and the QoS of the network, thus enabling scalable network slicing.

This document encodes the slice identifier in a portion of the MPLS Entropy Label (EL) defined in [RFC6790]. This has advantages in SR-MPLS networks as it avoids the use of additional label which would increase the size of the label stack. This also reuses the data plane processing of the Entropy Label on the egress LSR, the signaling of the Entropy Label capability from the egress to the ingress [I-D.ietf-isis-mpls-elc] [I-D.ietf-ospf-mpls-elc], and the signaling capability of transit routers to read this label [RFC8491] which allows for an easier and faster incremental deployment.

2. Entropy Label Control field

[RFC6790] defines the MPLS Entropy Label. [RFC6790] section 4.2 defines the use of the Entropy Label Indicator (ELI) followed by the Entropy Label (EL) and the MPLS header fields (Label, TC, S, TTL) in each. [RFC6790] also specifies that the TTL field of the EL must be set to zero by the ingress LSR.

Following the procedures of [RFC6790] EL is never used for forwarding and its TTL is never looked at nor decremented:

- An EL capable Egress LSR performs a lookup on the ELI and as a result pop two labels: ELI and EL.

- An EL non-capable Egress LSR performs a lookup on the ELI and as a result must drop the packet as specified in [RFC3031] for the handling of an invalid incoming label.

Hence essentially the TTL field of the EL behaves as a reserved field which must be set to zero when sent and ignored when received.

This documents extends the TTL field of the EL and calls it the Entropy Label Control (ELC) field. The ELC is a set of eight flags: ELC0 for bit 0, ELC1 for bit 1,..., ELC7 for bit 7.

Given that the MPLS header is very compact (32 bits) with no reserved bits and that MPLS is used within a trusted administrative domain, the semantic of these bits is not standardized but defined on a per administrative domain basis. This allows for increased re-use and flexibility of this scarce resource. As a consequence, an application using one of those buts MUST allow the choice of the bit by configuration by the network operator.

3. Slice Identifier

Each network slice in an MPLS domain is uniquely identified by a Slice Identifier (SLID). This section proposes to encode the SLID in a portion of the MPLS Entropy Label defined in [RFC6790].

The number of bits to be used for encoding the SLID in the EL is governed by a local policy and uniform within a network slice policy domain.

3.1. Ingress LSR

When an ingress LSR classifies that a packet belongs to the slice and that the egress has indicated via signaling that it can process EL for the tunnel, the ingress LSR pushes an Entropy Label with the:

- SLID encoded in the most significant bits of the Entropy Label.
- the entropy information encoded in the remaining lower bits of the Entropy Label as described in section 4.2 of [RFC6790].
- SPI bit (SLID Presence Indicator) set in one bit of the ELC field.

The choice of the ELC field used for SPI, and the number of bits to be used for encoding the SLID MUST be configurable by the network operator.

The slice classification method is outside the scope of this document.

3.2. Transit LSR

Any router within the SR domain that forwards a packet with the SPI bit set MUST use the SLID to select a slice and apply per-slice policies.

There are many different policies that could define a slice for a particular application or service. The most basic of these is bandwidth-allocation, an implementation complying with this specification SHOULD support the bandwidth-allocation slice as defined in the next section.

3.3. Bandwidth-Allocation Slice

A per-slice policy is configured at each interface of each router in the SR domain, with one traffic shaper per SLID. The bit rate of each shaper is configured to reflect the bandwidth allocation of the per-slice policy.

If shapers are not available, or desirable, an implementation MAY configure one scheduling queue per SLID with a guaranteed bandwidth equal to the bandwidth-allocation for the slice. This option allows a slice to consume more bandwidth than its allocation when available.

Per-slice shapers or queues effectively provides a virtual port per slice. This solution MAY be complemented with a per-virtual-port hierarchical DiffServ policy. Within the context of one specific slice, packets are further classified into children DiffServ queues which hang from the virtual port. The Traffic Class value in the MPLS header SHOULD be used for queue selection.

3.4. Backward Compatibility

The Entropy Label usage described in this document is consistent with [RFC6790] as ingress LSRs freely chooses the EL of a given flow, and transit LSRs treat the EL as an opaque set of bits.

As per [RFC6790] an ingress LSR that does not support this extension has the SPI bit cleared, and thus does not enable the SLID semantic of the Entropy bits. Hence, SLID-aware transit LSRs will not classify these packets into a slice.

3.5. Benefits

From a Segment Routing architecture perspective, this network slice identifier for SR-MPLS is inline with the network slice identifier for SRv6 proposed in [I-D.filsfils-spring-srv6-stateless-slice-id].

From an SR-MPLS perspective, using the EL to carry the network slice identifier has multiple benefits:

- This limits the number of labels pushed on the MPLS stack compared to using a pair of labels (ELI+EL) for flow entropy plus two or three labels for the slice indicator and the slice identifier. This is beneficial for the ingress LSR which may have limitations with regards to the number of labels pushed, for the transit LSR which may have limitations with regards to the label stack depth to be examined during transit in order to read both the entropy and the SLID. This presents additional benefit to network operators by reducing the packet overhead for traffic carried through the network;

- This avoids defining new extensions for the signaling of the egress capability to support the slice indicator and the slice identifier;

- This improves incremental deployment as all egress LSRs supporting EL can be sent the slice identifier from day one, allowing slice classification on transit LSRs.

4. End to end absolute loss measurements

This section describes the usage of a ELC flag to enable packet loss measurements, as described in section 3.1 of [RFC8321], for SR-MPLS networks.

TBD

5. Programmed sampling of packets

This section describes the usage of a ELC flag to detect end to end packet loss.

TBD

6. Changes / Authors Notes

[RFC Editor: Please remove this section before publication]

00: Initial version.

01: New co-author

7. References

7.1. Normative References

[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>.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

7.2. Informative References

[I-D.filsfils-spring-srv6-stateless-slice-id]
Filsfils, C., Clad, F., Camarillo, P., and K. Raza, "Stateless and Scalable Network Slice Identification for SRv6", draft-filsfils-spring-srv6-stateless-slice-id-02 (work in progress), January 2021.

[I-D.ietf-isis-mpls-elc]
Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS", draft-ietf-isis-mpls-elc-13 (work in progress), May 2020.

[I-D.ietf-ospf-mpls-elc]
Xu, X., Kini, S., Psenak, P., Filsfils, C., Litkowski, S., and M. Bocci, "Signaling Entropy Label Capability and Entropy Readable Label Depth Using OSPF", draft-ietf-ospf-mpls-elc-15 (work in progress), June 2020.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

[RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <https://www.rfc-editor.org/info/rfc8491>.

Authors' Addresses

Bruno Decraene (editor)
Orange

Email: bruno.decraene@orange.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018, CA 95134
Belgium

Email: wim.henderickx@nokia.com

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 22, 2021

R. Gandhi, Ed.
Z. Ali
C. Filsfils
F. Brockners
Cisco Systems, Inc.
B. Wen
V. Kozak
Comcast

February 18, 2021

MPLS Data Plane Encapsulation for In-situ OAM Data
draft-gandhi-mpls-ioam-sr-06

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the data packet while the packet traverses a path between two nodes in the network. This document defines how IOAM data fields are transported with MPLS data plane encapsulation using new Generic Associated Channel (G-ACh).

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 22, 2021.

Copyright Notice

Table of Contents

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than being sent within the probe packets specifically

dedicated to OAM or Performance Measurement (PM). The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data], and can be used for various use-cases for OAM and PM. The IOAM data fields are further updated in [I-D.ietf-ippm-ioam-direct-export] for direct export use-cases and in [I-D.ietf-ippm-ioam-flags] for Loopback and Active flags.

This document defines how IOAM data fields are transported with MPLS data plane encapsulations using new Generic Associated Channel (G-ACh).

2. Conventions

2.1. Requirement Language

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

2.2. Abbreviations

Abbreviations used in this document:

ECMP	Equal Cost Multi-Path
E2E	Edge-To-Edge
G-ACh	Generic Associated Channel
HbH	Hop-by-Hop
IOAM	In-situ Operations, Administration, and Maintenance
MPLS	Multiprotocol Label Switching
OAM	Operations, Administration, and Maintenance
PM	Performance Measurement
POT	Proof-of-Transit
PSID	Path Segment Identifier
PW	PseudoWire
SR	Segment Routing

SR-MPLS Segment Routing with MPLS Data plane

3. MPLS Extensions for IOAM Data Fields

3.1. IOAM Generic Associated Channel

The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. The IOAM data fields are carried in the MPLS header as shown in Figure 1. More than one trace options can be present in the IOAM data fields. G-ACh [RFC5586] provides a mechanism to transport OAM and other control messages over MPLS data plane. The IOAM G-ACh header [RFC5586] with new IOAM G-ACh type is added immediately after the MPLS label stack in the MPLS header as shown in Figure 1, before the IOAM data fields.

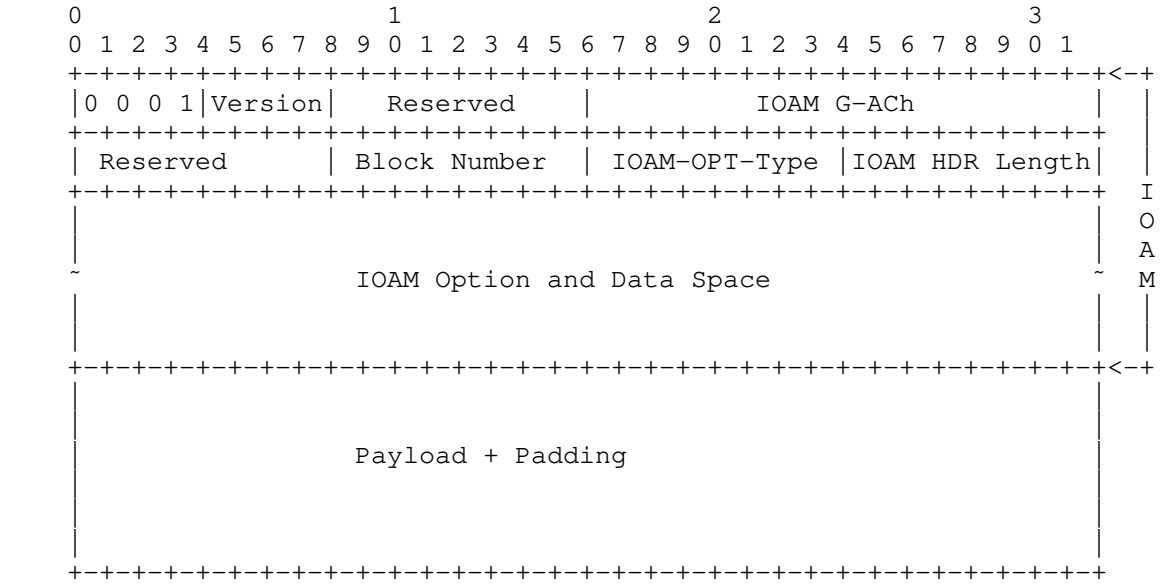


Figure 1: IOAM Generic Associated Channel with IOAM Data Fields

The IOAM data fields are encapsulated using the following fields in the MPLS header:

IP Version Number 0001b: The first four octets are IP Version Field part of a G-ACh header, as defined in [RFC5586].

Version: The Version field is set to 0, as defined in [RFC4385].

IOAM G-ACh: Generic Associated Channel (G-ACh) Type (value TBA3) for IOAM [RFC5586].

Reserved: Reserved Bits MUST be set to zero upon transmission and ignored upon receipt.

Block Number: The Block Number can be used to aggregate the IOAM data collected in data plane, e.g. compute measurement metrics for each block of a flow. It is also used to correlate the IOAM data on different nodes.

IOAM-OPT-Type: 8-bit field defining the IOAM Option type, as defined in Section 8.1 of [I-D.ietf-ippm-ioam-data].

IOAM HDR LEN: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

IOAM Option and Data Space: IOAM option header and data is present as defined by the IOAM-OPT-Type field, and is defined in Section 5 of [I-D.ietf-ippm-ioam-data].

3.2. IOAM Indicator Labels

An IOAM Indicator Label is used to indicate the presence of the IOAM data fields in the MPLS header. There are two IOAM types defined in this document: Edge-to-Edge (E2E) and Hop-by-Hop (HbH) IOAM. If only edge nodes need to process IOAM data then E2E IOAM Indicator Label is used so that intermediate nodes can ignore it. If both edge and intermediate nodes need to process IOAM data then HbH IOAM Indicator Label is used. Different IOAM Indicator Labels allow to optimize the IOAM processing on intermediate nodes by checking if IOAM data fields need to be processed.

4. Edge-to-Edge IOAM

4.1. Edge-to-Edge IOAM Indicator Label

The E2E IOAM Indicator Label is used to indicate the presence of the E2E IOAM data fields in the MPLS header as shown in Figure 2.

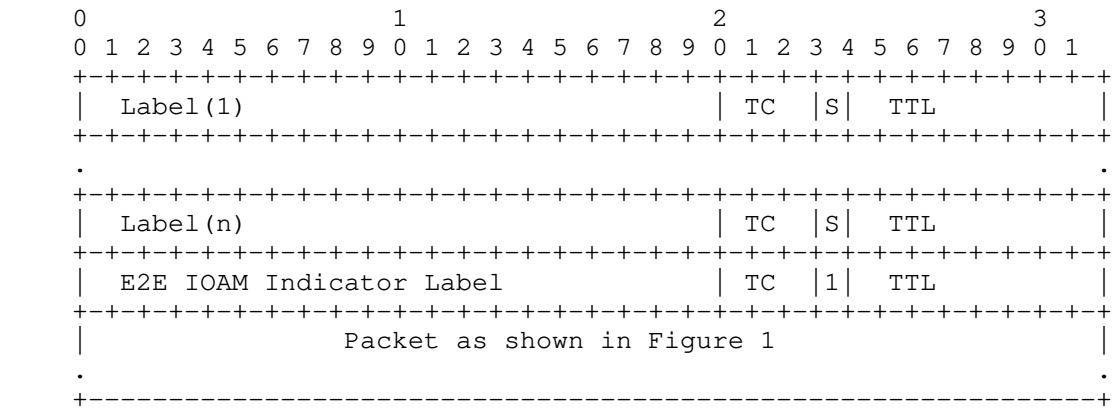


Figure 2: MPLS Encapsulation for E2E IOAM

The E2E IOAM data fields carry the Option-Type(s) that require processing on the encapsulating and decapsulating nodes only. The IOAM Option-Type carried can be IOAM Edge-to-Edge Option-Type [I-D.ietf-ippm-ioam-data]. The E2E IOAM data fields SHOULD NOT carry any IOAM Option-Type that require IOAM processing on the intermediate nodes as it will not be processed by them.

4.2. Procedure for Edge-to-Edge IOAM

The E2E IOM procedure is summarized as following:

- The encapsulating node inserts the E2E IOAM Indicator Label and one or more IOAM data fields in the MPLS header.

- The intermediate nodes do not process IOAM data fields.

- The decapsulating node "punts the timestamped copy" of the received packet as is including the IOAM data fields when the node recognizes the IOAM Indicator Label. The copy of the packet is punted with receive timestamp to the slow path for IOAM data fields processing. The receive timestamp is required by the various E2E OAM use-cases, including streaming telemetry. Note that it is not necessarily punted to the control-plane.

- The decapsulating node processes the IOAM data fields using the procedures defined in [I-D.ietf-ippm-ioam-data]. An example of IOAM processing is to export the data fields, send data fields via streaming telemetry, etc.

- The decapsulating node also pops the IOAM Indicator Label and the IOAM data fields from the received packet. The decapsulated

packet is forwarded downstream or terminated locally similar to the regular data packets.

4.3. Edge-to-Edge IOAM Indicator Label Allocation

The E2E IOAM Indicator Label is used to indicate the presence of the E2E IOAM data fields in the MPLS header. The E2E IOAM Indicator Label can be allocated using one of the following three methods:

- Label assigned by IANA with value TBA1 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].
- Label allocated by a Controller from the global table of the decapsulating node. The Controller provisions the label on both encapsulating and decapsulating nodes.
- Label allocated by the decapsulating node and signalled or advertised in the network. The signaling and/or advertisement extension for this is outside the scope of this document.

5. Hop-by-Hop IOAM

5.1. Hop-by-Hop IOAM Indicator Label

The HbH IOAM Indicator Label is used to indicate the presence of the HbH IOAM data fields in the MPLS header as shown in Figure 3.

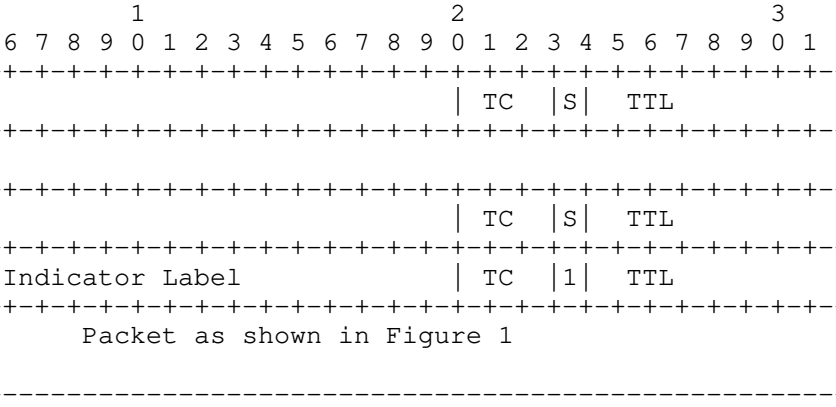


Figure 3: MPLS Encapsulation for HbH IOAM

The HbH IOAM data fields carry the Option-Type(s) that require processing at the intermediate and/or encapsulating and decapsulating nodes. The IOAM Option-Type carried can be IOAM Pre-allocated Trace Option-Type, IOAM Incremental Trace Option-Type and IOAM Proof of

Transit (POT) Option-Type, as well as Edge-to-Edge Option-Type [I-D.ietf-ippm-ioam-data].

5.2. Procedure for Hop-by-Hop IOAM

The HbH IOAM procedure is summarized as following:

- The encapsulating node inserts the HbH IOAM Indicator Label and one or more IOAM data fields in the MPLS header.

- The intermediate node enabled with HbH IOAM functions processes the data packet including the IOAM data fields as defined in [I-D.ietf-ippm-ioam-data] when the node recognizes the HbH IOAM Indicator Label present in the MPLS header. The intermediate node may 'punt the timestamped copy' of the received data packet including the IOAM data fields as required by the IOAM data fields processing. The copy of the packet is punted with receive timestamp to the slow path for IOAM processing.

- The intermediate node forwards a copy of the processed data packet downstream.

- The decapsulating node "punts the timestamped copy" of the received data packet as is including the IOAM data fields when the node recognizes the IOAM Indicator Label. The copy of the packet is punted with receive timestamp to the slow path for IOAM data fields processing. The receive timestamp is required by the various E2E OAM use-cases, including streaming telemetry. Note that it is not necessarily punted to the control-plane.

- The decapsulating node processes the IOAM data fields using the procedures defined in [I-D.ietf-ippm-ioam-data]. An example of IOAM processing is to export the data fields, send data fields via streaming telemetry, etc.

- The decapsulating node also pops the IOAM Indicator Label and the IOAM data fields from the received packet. The decapsulated packet is forwarded downstream or terminated locally similar to the regular data packets.

5.3. Hop-by-Hop IOAM Indicator Label Allocation

The HbH IOAM Indicator Label is used to indicate the presence of the HbH IOAM data fields in the MPLS header. The HbH IOAM Indicator Label can be allocated using one of the following three methods:

- Label assigned by IANA with value TBA2 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].

- Label allocated by a Controller from the network-wide global table. The Controller provisions the labels on all nodes participating in IOAM functions along the data traffic path.

- Labels allocated by the intermediate and decapsulating nodes and signalled or advertised in the network. The signaling and/or advertisement extension for this is outside the scope of this document.

6. Considerations for IOAM Indicator Label

6.1. Considerations for ECMP

The encapsulating node needs to make sure the IOAM data fields do not start with a well-known IP Version Number (e.g. 0x4 for IPv4 and 0x6 for IPv6) as that can alter the hashing function for ECMP that uses the IP header. This is achieved by using the IOAM G-ACh with IP Version Number 0001b after the MPLS label stack [RFC5586].

Note that the hashing function for ECMP that uses the labels from the MPLS header may now include the IOAM Indicator Label.

When entropy label [RFC6790] is used for hashing function for ECMP, the procedure defined in this document does not alter the hashing function.

6.2. Node Capability

The decapsulating node that has to pop the IOAM Indicator Label, data fields, and perform the IOAM function may not be capable of supporting it. The encapsulating node needs to know if the decapsulating node can support the IOAM function. The signaling extension for this capability exchange is outside the scope of this document.

The intermediate node that is not capable of supporting the IOAM functions defined in this document, can simply skip the IOAM processing of the MPLS header.

6.3. MSD Considerations

The SR path computation needs to know the Maximum SID Depth (MSD) that can be imposed at each node/link of a given SR path [RFC8664]. This ensures that the SID stack depth of a computed path does not exceed the number of SIDs the node is capable of imposing. The MSD used for path computation MUST include the IOAM Indicator Label.

6.4. Nested MPLS Encapsulation

The data packets with IOAM data fields carry only one IOAM Indicator Label in the MPLS header. Any intermediate node that adds additional MPLS encapsulation in the MPLS header may further update the IOAM data fields in the header without inserting another IOAM Indicator Label. When a packet is received with a HbH IOAM Indicator Label, the nested MPLS encapsulating node can add a HbH and/or E2E IOAM Option-Type. However, when a packet is received with an E2E IOAM Indicator Label, the nested MPLS encapsulating node SHOULD NOT add a HbH IOAM Option-Type, as intermediate nodes will not process it.

7. MPLS Encapsulation with Control Word and Another G-ACh for IOAM Data Fields

The IOAM data fields, including IOAM G-ACh header are added in the MPLS encapsulation immediately after the MPLS header. Any Control Word [RFC4385] or another G-ACh [RFC5586] MUST be added after the IOAM data fields in the packet as shown in the Figure 4 and Figure 5, respectively. This allows the intermediate nodes to easily access the HbH IOAM data fields located immediately after the MPLS header. The decapsulating node can remove the MPLS encapsulation including the IOAM data fields and then process the Control Word or another G-ACh following it.

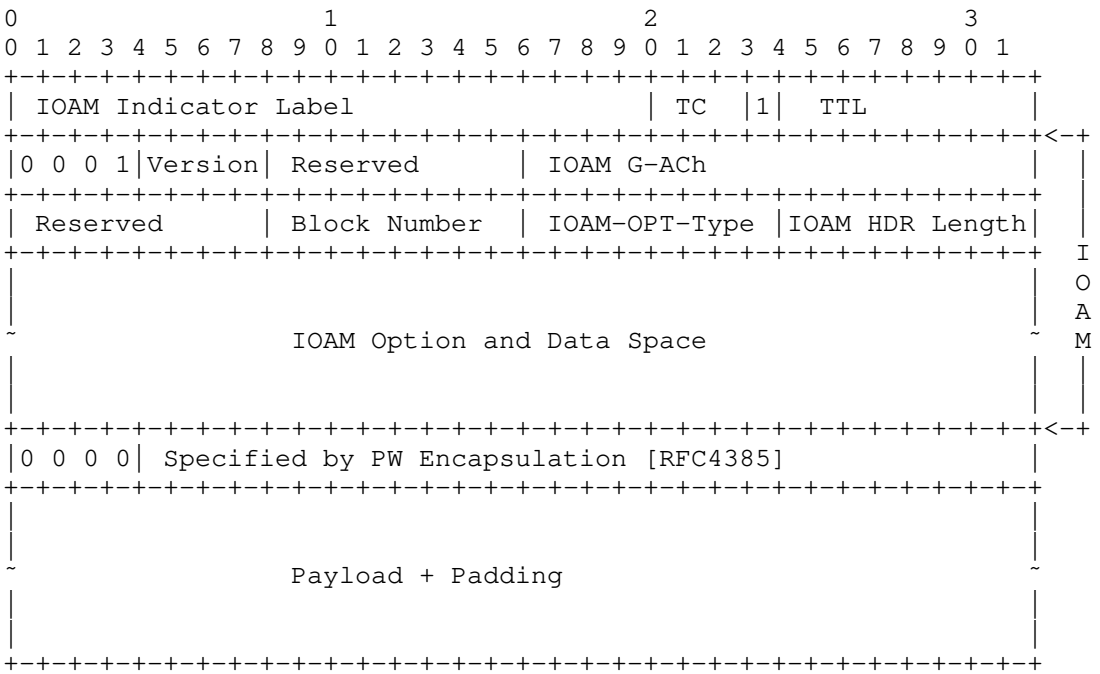


Figure 4: Example MPLS Encapsulation with Generic PW Control Word with IOAM

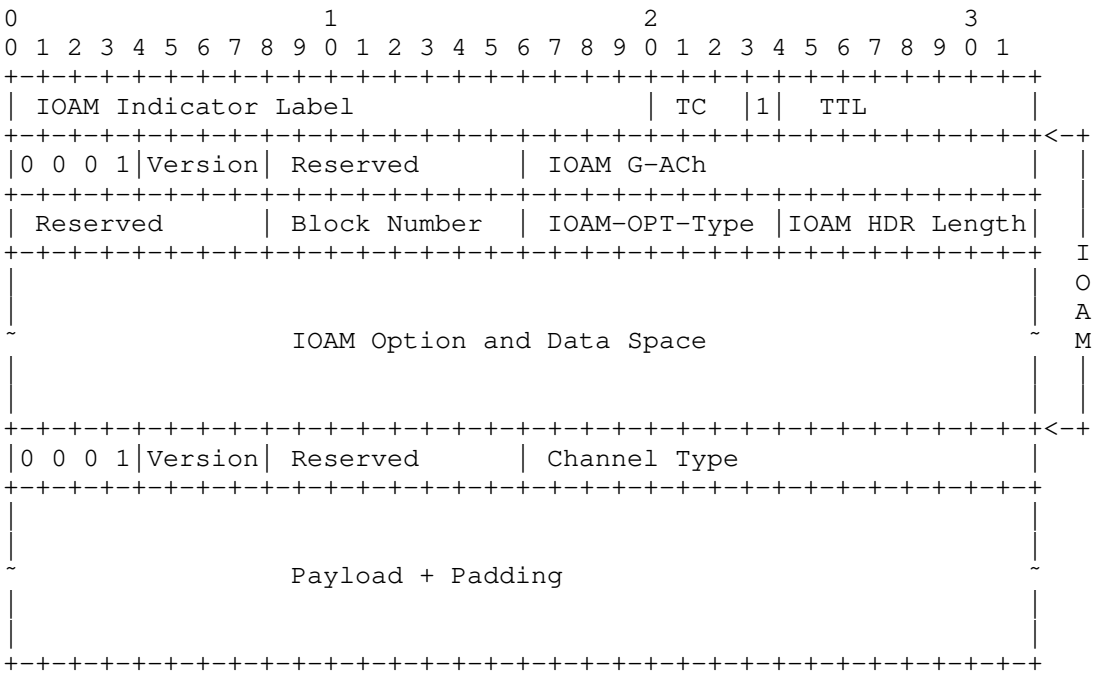


Figure 5: Example MPLS Encapsulation with Another G-ACh with IOAM

8. Example MPLS Encapsulations

8.1. Example SR-MPLS Encapsulation with IOAM

Segment Routing (SR) technology leverages the source routing paradigm [RFC8660]. A node steers a packet through a controlled set of instructions, called segments, by pre-pending the packet with an SR header. In the SR with MPLS data plane (SR-MPLS), the SR header is instantiated through a label stack.

An example of data packet with SR-MPLS encapsulation containing Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] and E2E IOAM data fields is shown in Figure 6. The PSID allows to identify the path associated with the data traffic being monitored for IOAM on the decapsulating node.

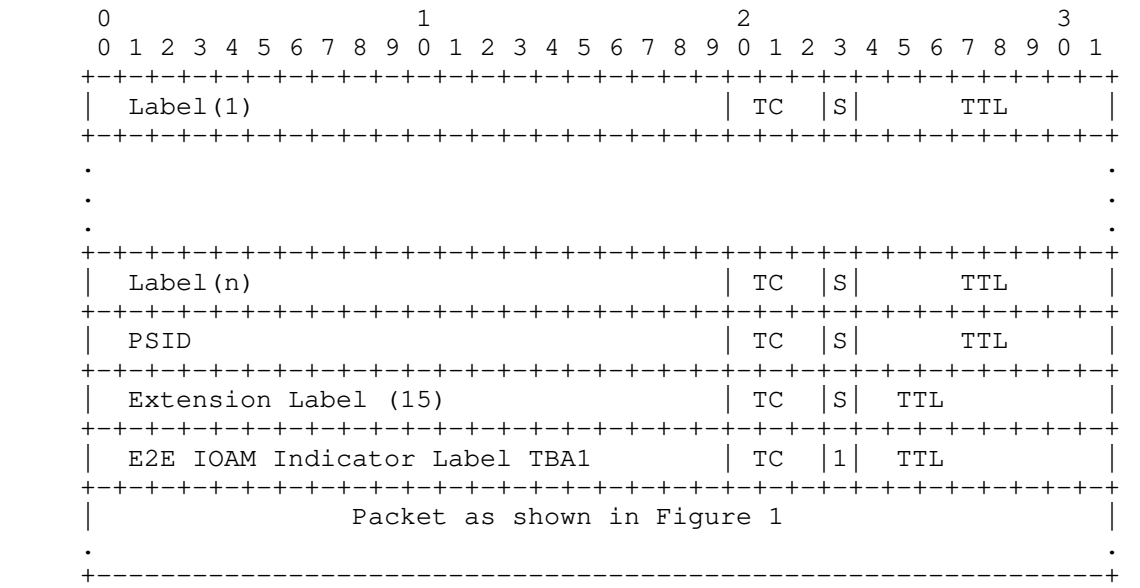


Figure 6: Example SR-MPLS Encapsulation with E2E IOAM Data Fields

9. Security Considerations

The security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

Routers that support G-ACh are subject to the same security considerations as defined in [RFC4385] and [RFC5586].

10. IANA Considerations

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

Value	Description	Reference
TBA1	E2E IOAM Indicator Label	This document
TBA2	HbH IOAM Indicator Label	This document

Table 1: IOAM Indicator Label Values

IANA maintains G-ACh Type Registry (see <https://www.iana.org/assignments/g-ach-parameters/g-ach-parameters.xhtml>). IANA is requested to allocate a value for IOAM G-ACh Type from "MPLS Generalized Associated Channel (G-ACh) Types (including Pseudowire Associated Channel Types)" registry.

Value	Description	Reference
TBA3	IOAM G-ACh Type	This document

Table 2: IOAM G-ACh Type

11. References

11.1. Normative References

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

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

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

[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>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <https://www.rfc-editor.org/info/rfc4385>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <https://www.rfc-editor.org/info/rfc5586>.

[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>.

11.2. Informative References

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

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

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

Acknowledgements

The authors would like to thank Patrick Khordoc, Shwetha Bhandari and Vengada Prasad Govindan for the discussions on IOAM. The authors would also like to thank Tarek Saad, Loa Andersson, Greg Mirsky, Stewart Bryant, Xiao Min, and Cheng Li for providing many useful comments. The authors would also like to thank Mach Chen, Andrew Malis, Matthew Bocci, and Nick Delregno for the MPLS-RT reviews.

Contributors

Sagar Soni
Cisco Systems, Inc.

Email: sagsoni@cisco.com

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

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

Email: fbrockne@cisco.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Voitek Kozak
Comcast

Email: Voitek_Kozak@comcast.com

Network Working Group
Internet-Draft
Updates: 8287 (if approved)
Intended status: Standards Track
Expires: August 27, 2021

N. Nainar, Ed.
Z. Ali
C. Pignataro
Cisco
F. Iqbal
Arista Networks
D. Rathi
S. Hegde
Juniper Networks

February 23, 2021

LSP Ping/Traceroute for Prefix SID in Presence of Multi-Algorithm/Multi-Topology Networks
draft-iqbal-spring-mpls-ping-algo-02

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. The machinery defined in [RFC8287] works well in single topology, single algorithm deployments where each Prefix SID is only associated with a single IP prefix. In multi-topology networks, or networks deploying multiple algorithms for the same IP Prefix, MPLS echo request needs to carry additional information in the Target FEC Stack sub-TLVs to properly validate IGP Prefix SID.

This document updates [RFC8287] by modifying IPv4 and IPv6 IGP-Prefix Segment ID FEC sub-TLVs to also include algorithm identification while maintaining backwards compatibility. This document also introduces new Target FEC Stack sub-TLVs for Prefix SID validation in multi-topology networks.

This Internet-Draft will expire on August 27, 2021.

Table of Contents

1. Introduction

[RFC8287] defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix SID and IGP-Adjacency SID with an MPLS data plane. [RFC8287] proposes 3 Target FEC Stack Sub-TLVs to carry this information. [I-D.ietf-lsr-flex-algo] introduces the concept of Flexible Algorithm that allows IGPs (ISIS, OSPFv2 and OSPFv3) to compute constraint-based path over an MPLS network. The constraint-based paths enables the IGP of a router to associate one or more Segment Routing Prefix-SID with a particular Flexible Algorithm, and steer packets along the constraint-based paths. Multiple Flexible Algorithms are assigned to the same IPv4/IPv6 Prefix while each utilizing a different MPLS Prefix SID label. Similarly, operators may deploy same IP prefix across multiple topologies in the network using IGP Multi-topology ID (MT-ID). As Flexible-Algorithm based deployments in particular, and multi-topology networks in general, become more common, existing OAM machinery requires updates to correctly diagnose network faults.

Segment Routing architecture [RFC8402] defines the context for IGP Prefix SID as a unique tuple comprised of prefix, topology, and algorithm>. Existing MPLS Ping/Traceroute machinery for SR Prefix SIDs, defined in [RFC8287], carries prefix, prefix length, and IGP protocol. To correctly identify and validate a Prefix-SID, the validating device also requires algorithm and topology identification to be supplied in the FEC Stack sub-TLV. This document extends SR-IGP IPv4 and IPv6 Prefix SID FECs to validate a particular algorithm in a single-topology network, while maintaining backwards compatibility with existing implementations of [RFC8287]. It also introduces new Target FEC Stack sub-TLVs to perform MPLS Ping and Traceroute for IGP Prefix SIDs in multi-topology, multi-algorithm deployments.

2. Conventions

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

The term "Must Be Zero" (MBZ) is used in object descriptions for reserved fields. These fields MUST be set to zero when sent and ignored on receipt.

Since this document refers to the MPLS Time to Live (TTL) far more frequently than the IP TTL, the authors have chosen the convention of using the unqualified "TTL" to mean "MPLS TTL" and using "IP TTL" for the TTL value in the IP header.

3. Motivation

In presence of multiple algorithms, a single IGP Prefix may be associated with zero or more IGP Prefix SIDs in addition to the default (Shortest Path First) Prefix SID. Each Prefix SID will have a distinct Prefix SID label and may possibly have a distinct set of next-hops based on associated constraint-based path calculation criteria. This means that to reach the same destination, an non-default algorithm IGP-Prefix SID may take a different path than default IGP Prefix SID algorithm.

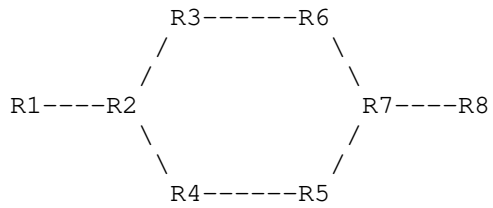


Figure above, which is a simplification of the diagram used in [RFC8287] illustrates this point through an example. Node Segment IDs for R1, R2, R3, R4, R5, R6, R7, and R8 for the default algorithm are 5001, 5002, 5003, 5004, 5005, 5006, 5007, and 5008, respectively. Nodes R1, R2, R4, R5, R7, and R8 also participate in Flexible Algorithm 128. Their corresponding Node Segment IDs for the algorithm are 5801, 5802, 5804, 5805, 5807, and 5808, respectively.

Now consider an MPLS LSP Traceroute request to validate the path to reach node R8 through Flexible Algorithm 128. The TTL of the first echo request packet expires at node R2 with incoming label 5808. Node R2 attempts to validate IGP-Prefix SID Target FEC stack sub-TLV from the echo request. However, this TFS sub-TLV does not contain information identifying the algorithm. As a result, R2 will attempt validation with default algorithm which expects the echo packet to arrive with Prefix SID label 5008. The validation fails, and node R2 responds with error code 10 resulting in a false negative.

Carrying algorithm identification in the Target FEC Stack sub-TLV of MPLS echo request will help avoid such false negatives. It will also

help detect forwarding deviations such as when the packet for a particular destination is incorrectly forwarded to a device that is participating in the default algo but does not participate in a given Flexible Algorithm.

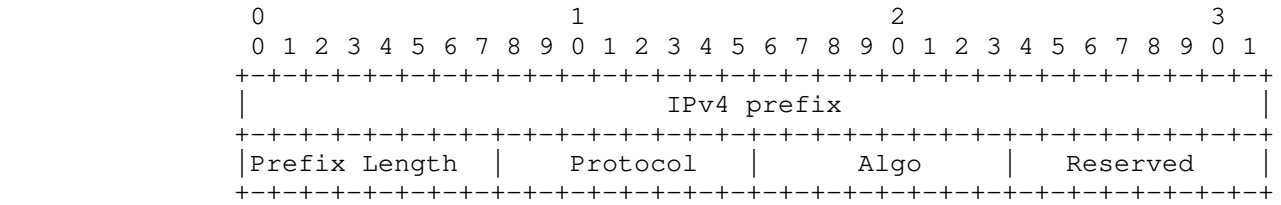
The above problem statement can also be extended to apply in Multi-Topology networks. In such networks, the Target FEC Stack sub-TLV MUST carry Multi-Topology ID (MT-ID) in addition to prefix, its length, IGP identification, and algorithm.

4. Algorithm Identification for IGP-Prefix SID 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]. This section updates IPv4 IGP-Prefix Segment ID Sub-TLV and IPv6 IGP-Prefix Segment ID Sub-TLV to also include an additional field identifying the algorithm.

4.1. IPv4 IGP-Prefix Segment ID Sub-TLV

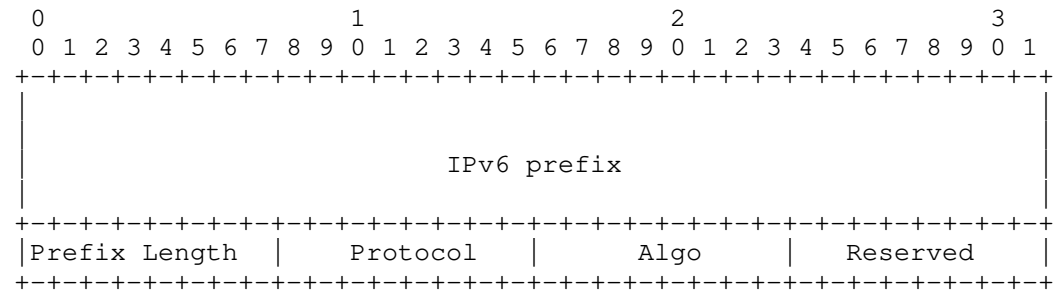
The Sub-TLV format for IPv4 IGP-Prefix Segment ID MUST be set as shown in the below TLV format:



Algo field MUST be set to 0 if the default algorithm is used. Algo field is set to 1 if Strict Shortest Path First (Strict-SPF) algorithm is used. For Flex-Algo, the Algo field MUST be set with the algorithm value (values can be 128-255).

4.2. IPv6 IGP-Prefix Segment ID Sub-TLV

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



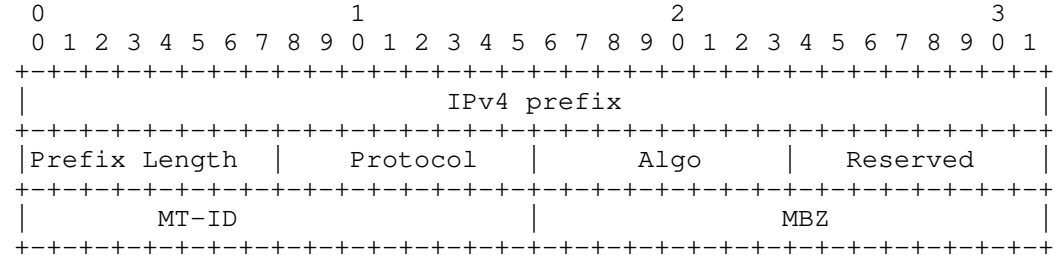
Algo field MUST be set to 0 if the default algorithm is used. Algo field is set to 1 if Strict Shortest Path First (Strict-SPF) algorithm is used. For Flex-Algo, the Algo field MUST be set with the algorithm value (values can be 128-255).

5. Multi-topology Support for IGP Prefix SID

IGP Prefix SID TLVs defined above assume a single-topology network for path validation. For Multi-Topology networks, this section introduces new Multi-Topology IGP IPv4 Prefix SID and Multi-Topology IGP IPv6 Prefix SID sub-TLVs in the Target FEC Stack TLV of MPLS echo request. These sub-TLVs carry MT-ID for OSPF and IS-IS protocols as specified in [RFC4915] and [RFC5120] respectively.

5.1. Multi-topology IPv4 IGP-Prefix Segment ID Sub-TLV

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

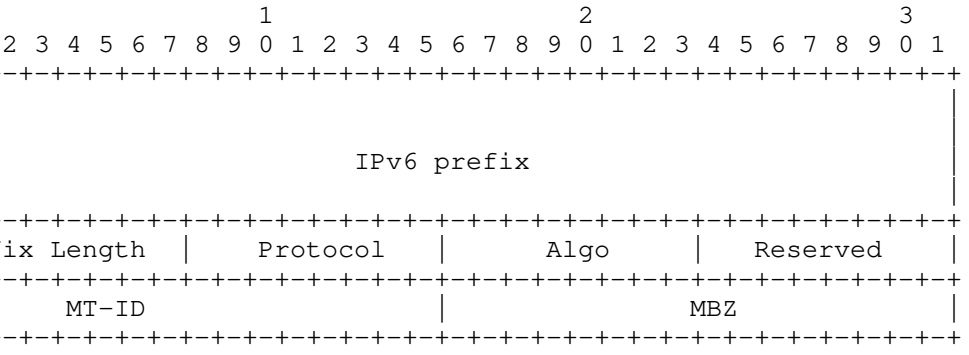


MT-ID identifies the Multi-Topology ID associated with the Prefix SID. MT-ID is set in trailing 12 bits of the field when the Protocol is set to IS-IS. Leading 4-bits of the MT-ID MUST be all zeroes for IS-IS. MT-ID is set to trailing 8 bits when the protocol is specified as OSPF. The leading octet MUST be set to all zeroes for OSPF. MBZ MUST be set to all zeroes.

The Protocol field MUST be set 1 if the responder MUST perform FEC validation using OSPF as the IGP protocol and MT-ID is an OSPF Multi-Topology ID. Protocol is set to 2 if the responder MUST perform FEC validation using IS-IS as the IGP protocol, and the MT-ID is IS-IS Multi-Topology ID. Protocol MUST not be set to 0 when using Multi-Topology IPv4 IGP Prefix SID sub-TLV.

5.2. Multi-Topology IPv6 IGP-Prefix Segment ID Sub-TLV

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



MT-ID identifies the Multi-Topology ID associated with the Prefix SID. MT-ID is set in trailing 12 bits of the field when the Protocol is set to IS-IS. Leading 4-bits of the MT-ID MUST be all zeroes for IS-IS. MT-ID is trailing 8 bits when the protocol is specified as OSPF. The leading octet MUST be set to all zeroes for OSPF. MBZ MUST be set to all zeroes.

The Protocol field MUST be set 1 if the responder MUST perform FEC validation using OSPF as the IGP protocol and MT-ID is an OSPF Multi-Topology ID. Protocol is set to 2 if the responder MUST perform FEC validation using IS-IS as the IGP protocol, and the MT-ID is IS-IS Multi-Topology ID. Protocol MUST not be set to 0 when using Multi-Topology IPv6 IGP Prefix SID sub-TLV.

6. Procedures

The below section describes LSP Ping and Traceroute procedures beyond the text specified in LSP

6.1. Single-Topology Networks

An array of network operators may deploy flexible algorithms in their network for constraint-based shortest paths, without deploying multi-topology. The updated FEC definitions for IGP Prefix SID allows operator to achieve LSP Ping and Traceroute in these networks while maintaining backwards compatibility with existing devices in the network. Below text highlights the handling procedures and initiator and responder for the updated FEC definitions.

6.1.1. Initiator Node Procedures

A node initiating LSP echo request packet for the Node Segment ID MUST identify and include the algorithm associated with the IGP Prefix SID in the Target FEC Stack sub-TLV. If the initiating node is not aware of the algorithm, the default algorithm (id 0) of Shortest Path First is assumed.

6.1.2. Responder Node Procedures

This section updates the procedures defined in Section 7.4 of [RFC8287] for IPv4/IPv6 IGP Prefix SID FEC. If the algorithm is 0, the procedures from [RFC8287] do not require any change. For any other algorithm value, if the responding node is validating the FEC stack, it MUST also validate the IGP Prefix SID advertisement for the algorithm defined in Algo field.

If the responding node is including IGP Prefix SID FEC in the FEC stack due to FEC Stack Change operation, it MUST also include algorithm associated with the Prefix SID.

6.2. Multi-Topology Networks

In presence of Multi-Topology networks, the operators can use the new Multi-Topology IGP IPv4/IPv6 Prefix SID FEC definitions to achieve path validation and fault isolation. Below text describes handling procedures for Multi-Topology networks for initiator and responder. The procedures defined in [RFC8287] are still applicable and the text below updates them instead of replacing them.

6.2.1. Initiator Node Procedures

A node initiating LSP echo request packet for Single-Topology network MAY use Multi-Topology IGP IPv4/IPv6 Prefix SID defined above. A node initiating LSP echo request for Multi-Topology networks MUST use Multi-Topology IGP IPv4/IPv6 Prefix SID defined above. The node MUST identify and include both the IGP MT-ID and the algorithm associated with the IGP prefix SID in addition to prefix, prefix length, and the

protocol. If the initiating node is not aware of the algorithm, the default algorithm (id 0) of Shortest Path First is assumed. The protocol MUST be set to 1 if the responding node is running OSPF, and 2 if the responding node is running IS-IS.

6.2.2. Responding Node Procedures

This section updates the procedures defined in Section 7.4 of [RFC8287] for Multi-Topology IPv4/IPv6 IGP Prefix SID FEC. Upon reception of the sub-TLV, responding node MUST validate that Protocol field is not 0 to correctly parse MT-ID. In addition to procedures defined in [RFC8287], if responding node is validating the FEC Stack, it MUST validate the IGP Prefix SID advertisement for the algorithm and the MT-ID described in the incoming FEC sub-TLV.

If the responding node is including Multi-Topology IGP Prefix SID FEC in the FEC stack due to a FEC Stack Change operation, it MUST also include the algorithm and MT-ID associated with the Prefix SID, and set the Protocol to 1 or 2, based on the corresponding IGP.

7. IANA Considerations

7.1. New Target FEC tack Sub-TLV

IANA is requested to assign two new Sub-TLVs from "Sub-TLVs for TLV Types 1, 16 and 21" sub-registry from the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" (IANA-MPLS-LSP-PING) registry.

Sub-Type	Sub-TLV Name	Reference
TBD1	Multi-topology IPv4 IGP-Prefix Segment ID	This document
TBD2	Multi-topology IPv6 IGP-Prefix Segment ID	This document

7.2. Algorithm in the Segment ID Sub-TLV

IANA is requested to create a new "Algorithm in the Segment ID Sub-TLV" registry under the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" registry. The initial entries are requested as below:

Value	Meaning	Reference
0	Default Algorithm	This document
1	Strict Shortest Path First (Strict-SPF)	This document

8. Security Considerations

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

9. Acknowledgements

TBA.

10. Contributors

TBA

11. References

11.1. 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>.

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

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

11.2. Informative References

[I-D.ietf-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[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-22 (work in progress), May 2019.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[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 (editor)
Cisco Systems, Inc.

Email: naikumar@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Carlos Pignataro
Cisco Systems, Inc.

Email: cpignata@cisco.com

Faisal Iqbal
Arista Networks

Email: faisal.ietf@gmail.com

Deepti
Juniper Networks Inc.

Email: deeptir@juniper.net

Shraddha
Juniper Networks Inc.

Email: shraddha@juniper.net

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

K. Kompella
V. Beeram
T. Saad
Juniper Networks
I. Meilik
Broadcom

February 22, 2021

Multi-purpose Special Purpose Label for Forwarding Actions
draft-kompella-mpls-mspl4fa-00

Abstract

A Slice Selector is packet metadata that dictates the packet's forwarding handling in order to conform to its slice requirements. There are multiple proposals for carrying slice selectors in MPLS networks. One of the more practical proposals is the "Global Identifier for Slice Selector" (GISS). Global uniqueness requires the GISS label be identified as such, via a special purpose label (ideally a base special purpose label (bSPL)). However, bSPLs are a precious commodity, and there are many requests for them. This document serves two purposes: to define a bSPL for carrying a GISS, and to show how this bSPL can consolidate many current requests for special purpose labels while carrying associated data compactly and efficiently.

Status of This Memo

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

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

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

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

Table of Contents

1. Introduction

Network slicing is an important ongoing effort both for network design, as well as for standardization, in particular at the IETF [I-D.nsdt-teas-ns-framework]. A key issue is identifying which slice a packet belongs to, by means of a "slice selector" carried in the packet header. [I-D.bestbar-teas-ns-packet] describes several such methods for MPLS networks, of which the Global Identifier for Slice Selector (GISS) is one of the more practical solutions. This document shows how to realize the GISS using a base special purpose label (bSPL).

Base Special Purpose Labels are a precious commodity; there are only 16 such values, of which 8 have already been allocated. There are currently five requests for bSPLs that the authors are aware of; this document proposes another use case for a bSPL, in all consuming nearly all the remaining values. Therefore, this document also suggests a method whereby a single bSPL can be used for all the purposes currently documented. This leads to perhaps the more valuable long-term contribution of this document: an approach to the definition and use of bSPLs (and SPLs in general) whereby a single value can be used for multiple purposes, and provide a flexible and efficient means of carrying associated data.

1.1. Terminology

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

1.2. Slice Selector

In MPLS networks, a GISS is a data plane construct identifying packets belonging to a slice aggregate (the set of packets that belong to the slice). The GISS dictates forwarding actions for the slice aggregate: QoS behavior and next hop selection. The purpose of the GISS is detailed in [I-D.bestbar-teas-ns-packet]. To embed a GISS in a label stack, one must preface it with a bSPL identifying it as such. For reasons that will become apparent, this bSPL is called the Forwarding Actions Indicator (FAI).

2. Multi-purpose bSPL: the Forwarding Actions Indicator

This document proposes the use of a single bSPL to tell routers one or more forwarding actions they should take on a packet, e.g.:

- to treat a packet according to its slice, given its GISS;
- to load balance a packet, given its entropy;
- whether or not to perform fast reroute on a failure [I-D.kompella-mpls-nffrr];
- whether or not the packet has a Flow ID;
- to update statistics based on the path identifier [I-D.hegde-spring-traffic-accounting-for-sr-paths];

- to view/update OAM metadata;
  [I-D.cheng-mpls-inband-pm-encapsulation],
  [I-D.gandhi-mpls-ioam-sr], other approaches.

- to reassemble a fragmented packet
  [I-D.zzhang-tsvwg-generic-transport-functions];

- and perhaps other functions in the future.

This bSPL is called the "Forwarding Actions Indicator" (FAI). The FAI uses the label's TC bits and TTL field to inform the forwarding plane of the required actions. Each of these actions may have associated data, the Forwarding Actions Data (FAD). The FAD may be carried in the Label Stack (LS FAD) or in the payload (PL FAD).

2.1. The FAI bSPL

The design of the bSPL hinges on a key insight: for labels not at the top of the label stack, the only bits that a forwarding engine looks at are the label value field (to compute entropy and identify SPLs) and the End of Stack (S) bit (to know when the label stack ends). [If you know of a forwarding engine that looks at other bits of labels below the top of stack, please contact the authors.] This means that for a bSPL that will never appear at the top of stack, the TC bits and the TTL bits can be used to carry additional information. Furthermore, for the LS FAD, the entire 4-octet label word, the S bit excepted, can be used to carry data. We use this technique to make the FAI bSPL multipurpose, and to make the FAD words compact and efficient.

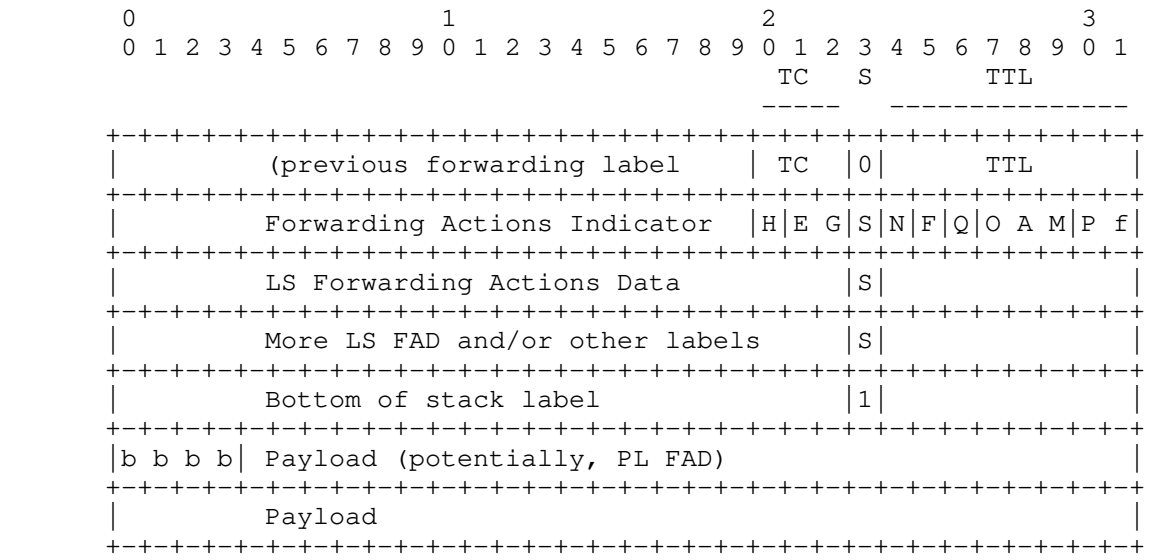


Figure 1: Format for FAI, LS FAD and PL FAD

The FAI's label value MUST be the IANA allocated value. The S bit MUST be reflect whether the label stack ends at this label or not.

The TC and TTL bits are used as flags, defined as follows:

H: if set, the FAI is followed by a Forwarding Actions Header (FAH).

EG: this is a 2-bit flag indicating whether the LS FAD carries Entropy and/or GISS information.

S: MUST be set if the FAI is the end of stack, and clear otherwise.

N: If set, do not do fast reroute (NFFRR).

F: If set, the LS has a Flow ID.

Q: if set, the payload contains Opaque data.

OAM: a 3-bit field that specifies what type(s) of OAM is carried in the in the label stack and/or payload.

P: If set, the PL FAD contains a Path Identifier.

F: If set, the payload contains a Fragmentation Header.

The EG field is used as follows:

00: No Entropy or GISS present

01: LS FAD 0 contains 16 bits of Entropy in the high order 16 bits and 15 bits of GISS in the low order 16 bits (S bit excepted).

10: LS FAD 0 contains 20 bits of Entropy in the high order 20 bits and 11 bits of GISS in the low order 12 bits (S bit excepted).

11: LS FAD 0 contains the 31-bit Entropy; LS FAD 1 contains the 31-bit GISS. In LS FAD 0, the S bit MUST be 0; the packet forwarding engine may choose to use this as part of the Entropy, as it doesn't affect the outcome. In LS FAD 1, the S bit may be 0 or 1.

Here's how the LS FAD is parsed. One must keep track of the S bit to know when the stack ends. It is an error if the label stack ends while there are more PL FAD words to process.

1. Set NL ("next label") to the first 4-octet word of the LS FAD. Set PL ("payload") to the first 4-octet word of the payload.

2. Process H: if set, (TBD); otherwise, NL is unchanged.

3. Process EG:

    1. If EG is 00, NL is unchanged.

    2. If EG is 01 or 10, NL contains both GISS and Entropy. Increment NL.

    3. If EG is 11, NL contains GISS; NL+1 contains Entropy. Increment NL by 2.

4. Process N. NL is unchanged.

5. Process F:

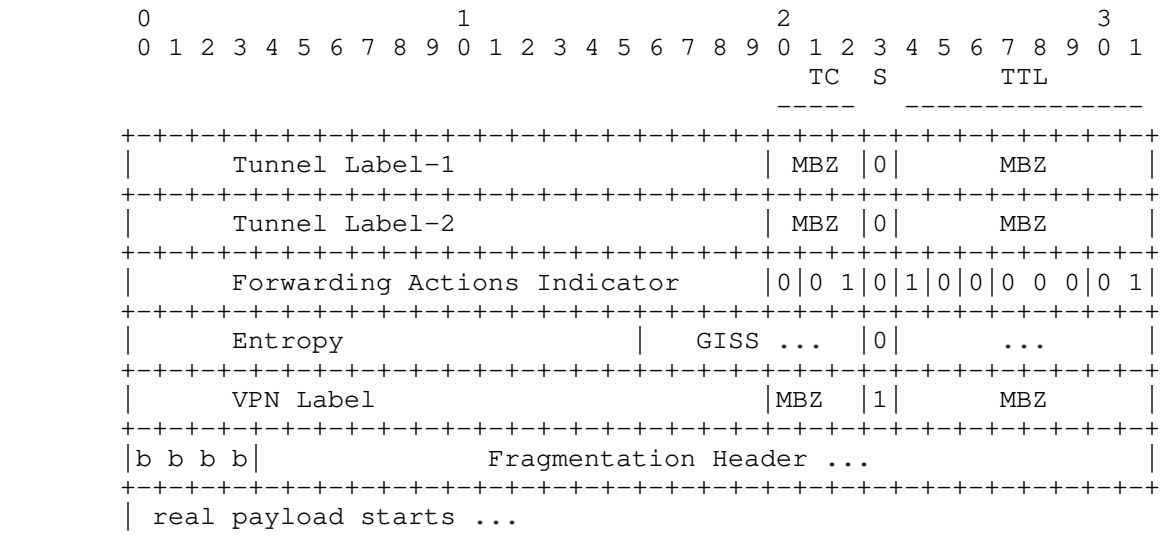
    1. If F is set, NL contains the Flow ID; increment NL.

6. Process Q:

    1. If set, (TBD); otherwise, NL is unchanged.

7. NL now points at next label in the stack.

A similar procedure applies to parsing the PL; details will be forthcoming when the OAM field is better defined.



H = 0; ignore.
EG = 01: LS FAD 0 contains Entropy + GISS.
N = 1: NFFRR is set.
F = 0: No Flow ID.
Q = 0: ignore.
OAM = 0; ignore.
P = 0: no Path Identifier in payload.
F = 1: Fragmentation Header is present.

Figure 2: Example of FAI + LS FAD + PL FAD

The real payload starts after the Fragmentation Header.

3. Issues to be Resolved

3.1. Preventing FAI From Reaching Top of Stack

As was said earlier, the FAI MUST NOT be at the top of stack, since its TC and TTL bits have been repurposed. There are two ways to prevent this. If an LSR X pops a label and encounters an FAI, X can pop the FAI and all LS FAD words. To do that, it must be able to parse the FAI to determine how many LS FAD words there exist. This can be used in conjunction with Section 3.2. However, there are cases when it is desired to preserve the FAI+FAD until the egress. In this case, X should push an explicit NULL (label value 0 or 2) onto the stack above the FAI, with the correct TC and TTL values.

Other options will be pursued.

3.2. Repeating the FAI at "Readable Stack Depth"

For LSRs which cannot parse the entire label stack, or would prefer not to unless needed, it is possible to repeat the FAI at "readable stack depth", say every 4 labels. In the above case, the FAI+LS FAD can be repeated every 4 labels; or a truncated version, just the FAD with GE set to 00 can be used. Only the last FAI would contain the full information, reducing the burden on the label stack. However, in this case, LSRs that don't process the whole stack may not load balance less effectively, and potentially not adhere to the slice service level objectives.

Other options will be described in future versions of this document.

3.3. First Nibble Issues

The first nibble of the first word of the payload SHOULD NOT be 0x4 or 0x6, as legacy LSRs may use the heuristic that this indicates a payload of IPv4/IPPv6. iOAM data has a first nibble of 0x1. However, if there is no iOAM data, the first nibble of the Path Identifier, if any, else that of the Fragmentation Header, MUST NOT be 0x4 or 0x6. However, it is inefficient to have to address this issue for every type of PL FAD, as it may be the first word in the payload. A future version of this document will propose an alternative solution.

It is unclear when a Control Word may be present as the first word of the payload; this is sometimes signaled and sometimes configured. When it is present, the above issue is moot.

4. Contributors

Many thanks to Colby Barth, Chandra Ramachandran and Srihari Sangli for their contributions to this draft.

5. Acknowledgments

We'd like to acknowledge the helpful discussions with Swamy SRK.

6. IANA Considerations

If this draft is deemed useful and adopted as a WG document, the authors request the allocation of a bSPL for the FAI. We suggest the early allocation of label 8 for this.

7. Security Considerations

A malicious or compromised LSR can insert the FAI and associated data into a label stack, preventing (for example) FRR from occurring. If so, protection will not kick in for failures that could have been protected, and there will be unnecessary packet loss. Similarly, inserting or removing a Fragmentation Header means that a packet's contents cannot be accurately reconstructed. Inserting or changing a GISS means that the packet will be misclassified, perhaps leaving or entering a high-value slice and causing damage.

8. References

8.1. Normative References

[I-D.bestbar-teas-ns-packet]
Saad, T., Beeram, V., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., and X. Liu, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-01 (work in progress), December 2020.

[I-D.cheng-mpls-inband-pm-encapsulation]
Cheng, W., Min, X., Zhou, T., Dong, X., and Y. Peleg, "Encapsulation For MPLS Performance Measurement with Alternate Marking Method", draft-cheng-mpls-inband-pm-encapsulation-04 (work in progress), September 2020.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-05 (work in progress), January 2021.

[I-D.hegde-spring-traffic-accounting-for-sr-paths]
Hegde, S., "Traffic Accounting for MPLS Segment Routing Paths", draft-hegde-spring-traffic-accounting-for-sr-paths-02 (work in progress), October 2018.

[I-D.kompella-mpls-nffrr]
Kompella, K. and W. Lin, "No Further Fast Reroute", draft-kompella-mpls-nffrr-01 (work in progress), November 2020.

[I-D.zzhang-tsvwg-generic-transport-functions]
Zhang, Z., Bonica, R., and K. Kompella, "Generic Transport Functions", draft-zzhang-tsvwg-generic-transport-functions-00 (work in progress), November 2020.

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

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

8.2. Informative References

[I-D.nsdt-teas-ns-framework] Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States

Email: kireeti.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States

Email: vbeeram@juniper.net

Tarek Saad
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States

Email: tsaad@juniper.net

Israel Meilik
Broadcom

Email: israel.meilik@broadcom.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2021

Z. Li
J. Dong
Huawei Technologies

April 14, 2021

Carrying Virtual Transport Network Identifier in MPLS Packet
draft-li-mpls-enhanced-vpn-vtn-id-01

Abstract

A Virtual Transport Network (VTN) is a virtual network which has a customized network topology and a set of dedicated or shared network resources allocated from the underlying network infrastructure. Multiple VTNs can be created by network operator for using as the underlay for one or a group of VPNs services to provide enhanced VPN (VPN+) services. In packet forwarding, some fields in the data packet needs to be used to identify the VTN the packet belongs to, so that the VTN-specific processing can be executed.

This document proposes a mechanism to carry the VTN-ID in an MPLS packet to identify the VTN the packet belongs to. The procedure for processing the VTN ID is also specified.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 16, 2021.

Table of Contents

1. Introduction

Virtual Private Networks (VPNs) provide different groups of users with logically isolated connectivity over a common shared network infrastructure. With the introduction of 5G, new service types may require connectivity services with advanced characteristics comparing to traditional VPNs, such as strict isolation from other services or guaranteed performance. These services are refered to as "enhanced VPNs" (VPN+). [I-D.ietf-teas-enhanced-vpn] describes a framework and candidate component technologies for providing VPN+ services.

The enhanced properties of VPN+ require integration between the overlay connectivity and the characteristics provided by the underlay network. To meet the requirement of enhanced VPN services, a number of Virtual Transport Networks (VTNs) need to be created, each consists of a subset of the underlay network topology and a set of network resources allocated from the underlay network to meet the requirement of one or a group of VPN+ services. In the network, traffic of different VPN+ services may to be processed separately based on the topology and the network resources associated with the corresponding VTN.

For network scenraios where a large number of VTNs need to be created and maintained, [I-D.dong-teas-enhanced-vpn-vtn-scalability] describes the scalability considerations for VTN. One approach to improve the data plane scalability is introducing a dedicated VTN Identifier (VTN-ID) in data packets to identify the VTN the packets belong to, so that VTN-specific packet processing can be performed by network nodes.

This document proposes a mechanism to carry the VTN Identifier (VTN-ID) and the related information in MPLS [RFC3031] data packets, so that the packet will be processed by network nodes using the set of network resources allocated to the corresponding VTN. The procedure for processing the VTN-ID is also specified. The forwarding path of the MPLS LSP is determined using the MPLS label stack in the packet, and the set of local network resources used for processing the packet is determined by the VTN-ID. The mechanism introduced in this document is applicable to both MPLS networks with RSVP-TE [RFC3209] or LDP [RFC5036] LSPs, and MPLS networks with Segment Routing (SR) [RFC8402] [RFC8660].

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 BCP14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Carrying VTN Information in MPLS Packet

This document defines a new VTN header which is used to carry the VTN-ID and other VTN related information. In an MPLS packet, The VTN header follows the MPLS label stack, and precedes the header and payloads in the upper layer. The format of VTN header is shown as below:

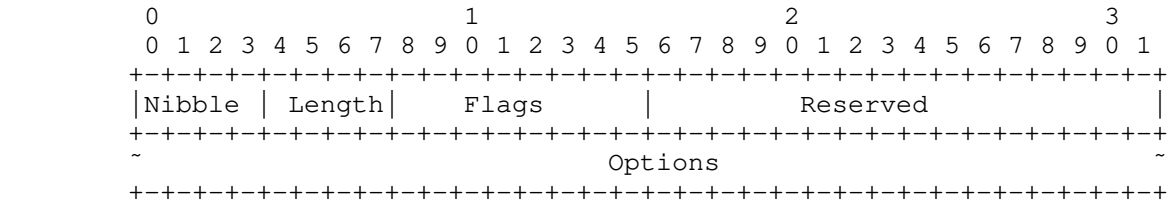


Figure 1. The format of MPLS VTN Header

Where:

- Nibble: The first 4-bit field is set to the binary value 0010. This is to ensure that the VTN header will not be interpreted as an IP header or the ACH of pseudowire packet.

- Length: Indicate the length of the MPLS VTN header in 32-bit words.

- Flags: 8-bit Flags field. All the flags are reversed for future use. This field SHOULD be set to zero on transmission and MUST be ignored on receipt.

- Reserved: 16-bit field reserved for future use.

A new VTN-ID Option is defined in this document, other option types may be defined in future documents. The format of the VTN-ID Option is shown as below:

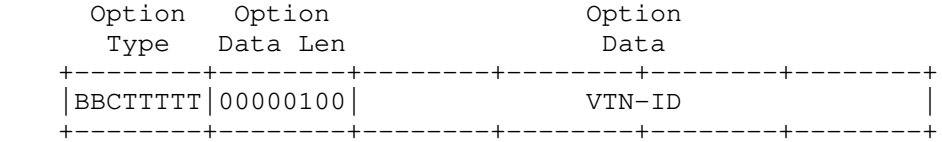


Figure 2. The format of VTN-ID Option

Option Type: 8-bit identifier of the type of option. The type of VTN-ID option is to be assigned by IANA. The highest-order bits of the type field are defined as below:

- BB 00 The highest-order 2 bits are set to 00 to indicate that a node which does not recognize this type will skip over it and continue processing the header.

- C 1 The third highest-order bit are set to 1 to indicate this option may change en route.

Opt Data Len: 8-bit unsigned integer indicates the length of the option Data field of this option, in octets. The value of Opt Data Len of the VTN-ID option SHOULD be set to 4.

Option Data: 4-octet identifier which uniquely identifies a VTN within a network domain.

A new MPLS special-purpose label or extended special-purpose label is defined as the VTN Header Indicator (VHI), its value is to be assigned by IANA. The VHI label is used to indicate the existence of the VTN Header after the MPLS label stack in the packet. The position of the VHI label in the MPLS label stack is not limited.

The benefit of introducing the MPLS VTN header to carry the VTN-ID and the related information is that it provides the flexibility to encode information which cannot be accomodated in an MPLS label (20-bit), and the length of the header can be variable.

4. Procedures

4.1. VTN Header Insertion

When the ingress node of an LSP receives a packet, according to traffic classification or mapping policy, the packet is steered into one of the VTNs in the network, then a VTN header SHOULD be inserted into the packet, and the VTN-ID which the packet is mapped to SHOULD be carried in the VTN header. The ingress node SHOULD also encapsulates the packet with an MPLS label stack which are used to determine the path traversed by the LSP. The VHI label SHOULD be inserted in the label stack to identify the existence of the VTH header.

4.2. VTN based Packet Forwarding

On receipt of a MPLS packet which carries the VHL and the VTN header, network nodes which support the mechanism defined in this document SHOULD scan the label stack to figure out the existence of the VHL. If there is a VHL in the label stack, then the network node SHOULD parse the VTN header and use the VTN-ID to identify the VTN the packet belongs to, and use the local resources allocated to the VTN to process and forward the packet. The forwarding behavior is based on both the top MPLS label and the VTN-ID. The top MPLS label is used for the lookup of the next-hop, and the VTN-ID can be used to determine the set of network resources allocated by the network nodes for processing and sending the packet to the next-hop.

There can be different approaches used for allocating network resources on each network node to the VTNs. For example, on one interface, a subset of forwarding plane resource (e.g. bandwidth and the associated buffer/queuing/scheduling resources) allocated to a particular VTN can be considered as a virtual layer-2 sub-interface with dedicated bandwidth and the associated resources. In packet forwarding, the top MPLS label of the received packet is used to identify the next-hop and the outgoing Layer 3 interface, and the VTN-ID is used to further identify the virtual sub-interface which is associated with the VTN on the outgoing interface.

Network nodes which do not support the mechanism in this document SHOULD ignore the VHL and the VTN header, and forward the packet only based on the top MPLS label.

The egress node of the MPLS LSP SHOULD pop the VHL together with other LSP labels, and decapsulate the VTN header.

5. Capability Advertisement and Negotiation

Before inserting the VTN header into an MPLS packet, the ingress node MAY need to know whether the nodes along the LSP can process the VTN header properly according to the mechanisms defined in this document. This can be achieved by introducing the capability advertisement and negotiation mechanism for the VTN header. The ingress node also need to know whether the egress node of the LSP can remove the VTN header properly before parsing the upper layer and send the packet to the next hop. The capability advertisement and negotiation mechanism will be described in a future version of this document.

6. IANA Considerations

IANA is requested to assign a new special-purpose label from the "Special-Purpose MPLS Label Values" or "Extended Special-Purpose MPLS Label Values" registry.

Value	Description	Reference
TBD	VTN Header Indicator	this document

IANA is requested to assign a new option type of the MPLS VTN extension header:

Value	Description	Reference
TBD	VTN-ID	this document

7. Security Considerations

TBD

8. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

9. Acknowledgements

TBD.

10. References

10.1. Normative References

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

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

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

10.2. Informative References

[I-D.dong-teas-enhanced-vpn-vtn-scalability] Dong, J., Li, Z., Qin, F., and G. Yang, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-01 (work in progress), November 2020.

[I-D.ietf-teas-enhanced-vpn] Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-06 (work in progress), July 2020.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[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>.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[TS23501] "3GPP TS23.501", 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: lizhenbin@huawei.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: jie.dong@huawei.com

MPLS Working Group
Internet-Draft
Updates: 8595 (if approved)
Intended status: Standards Track
Expires: August 25, 2021

Y. Liu
G. Mirsky
ZTE Corporation

February 21, 2021

MPLS-based Service Function Path(SFP) Consistency Verification
draft-lm-mpls-sfc-path-verification-02

Abstract

This document describes extensions to MPLS LSP ping mechanisms to support verification between the control/management plane and the data plane state for SR-MPLS service programming and MPLS-based NSH SFC.

This document defines the signaling of the Generic Associated Channel (G-ACh) over a Service Function Path (SFP) with an MPLS forwarding plane using the basic unit defined in RFC 8595. The document updates RFC 8595 in respect to SFF's handiling TTL expiration. The document also describes the processing of the G-ACh by the elements of the SFP.

Table of Contents

1. Introduction

Service Function Chain (SFC) defined in [RFC7665] as an ordered set of service functions (SFs) to be applied to packets and/or frames, and/or flows selected as a result of classification.

SFC can be achieved through a variety of encapsulation methods, such as NSH [RFC8300], SR service programming [I-D.ietf-spring-sr-service-programming] and MPLS-based NSH SFC [RFC8595].

This document describes extensions to MPLS LSP ping [RFC8029] mechanisms to support verification between the control/management plane and the data plane state for both SR-MPLS service programming and MPLS-based NSH SFC.

An MPLS LSP ping is a component of the MPLS Operation, Administration, and Maintenance (OAM) toolset. OAM packets used to monitor a specific Service Function Path (SFP) can be transported

over a Generic Associated Channel (G-ACh). This document defines the signaling of the G-ACh over an SFP with an MPLS forwarding plane using the basic unit defined in [RFC8595]. The document updates [RFC8595] in respect to SFF's handiling TTL expiration. The document also describes the processing of the G-ACh by the elements of the SFP.

2. Conventions used in this document

2.1. Requirements Language

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

2.2. Terminology and Acronyms

SFC: Service Function Chain

SFF: Service Function Forwarder

SF: Service Function

SFI: Instance of an SF

SFP: Service Function Path

RSP: Rendered Service Path

SFC-MPLS: SFC over an MPLS forwarding plane introduced in [RFC8595]

SR-SFC: SFC achieved by SR service programming [I-D.ietf-spring-sr-service-programming]

NSH-SR: SFC based on the integration of Network Service Header (NSH) and SR for SFC [I-D.ietf-spring-nsh-sr]

SPL: Special-Purpose Label

bSPL: Base SPL

eSPL: Extended SPL

GAL: Generic Associated Channel Label

ELI: Entropy Label Indicator

OAM: Operation, Administration, and Maintenance

G-ACh: Generic Associated Channel

GAL: Generic Associated Channel Label

3. MPLS-based SFP Consistency Verification

MPLS echo request and reply messages [RFC8029] can be extended to support the verification of the consistency of an MPLS-based Service Function Path (SFP).

SR-MPLS/MPLS can be used to realize an SFP. Two methods have been defined:

- [I-D.ietf-spring-sr-service-programming] describes how to achieve service function chaining in SR-enabled MPLS and IPv6 networks. In an SR-MPLS network, each SF is associated with an MPLS label. As a result, an SFP can be encoded as a stack of MPLS labels and pushed on top of the packet.

- [RFC8595] provides another method to realize SFC in an MPLS network by means of using a logical representation of the Network Service Header (NSH) in an MPLS label stack. This method, throughout this document, is referred to as SFC over an MPLS data plane (SFC-MPLS). When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used, which can be present one or more times in the label stack. This unit comprises two MPLS labels, one carries a label to provide a context within the SFC scope (the SFC Context Label), and the other carries a label to show which SF is to be enacted (the SF Label). SFC forwarding can be achieved by label swapping, label stacking, or the mix of both. When an SFF receives a packet containing an MPLS label stack, it examines the top basic unit of the MPLS label stack for SFC, {SPI, SI} or {context label, SFI index}, to determine where to send the packet next.

In MPLS Label Switched Paths (LSPs), MPLS LSP ping [RFC8029] is used to check the correctness of the data plane functioning and to verify the data plane against the control plane.

The proposed extension of MPLS LSP ping allows verification of the correlation between the control/management (if data model-based central controller used) plane and the data plane state in SR-MPLS/ MPLS-based SFC.

As for NSH-SR, OAM defined for NSH in [draft-ietf-sfc-multi-layer-oam] can be re-used and it is out of the scope of this document.

4. LSP Ping in SFC-MPLS

In SFC-MPLS, SFFs are responsible for MPLS echo request processing. there're two reasons:

- In SFC-MPLS, the packet forwarding decision is made by SFFs based on the basic unit. SFs are not aware of the FEC of the basic unit.

- Generally, except for the designed specific functions, the packet processing functions supported by SFs are limited. SFs may not support control and/or management protocols operated over the G-ACh defined in [RFC5586], e.g., MPLS OAM protocols like LSP ping. Such packets may be mishandled.

To support that processing, the basic unit can use the mechanism described in Section 4.1.

4.1. Special-purpose Label in SFC-MPLS Environment

When an SFC-MPLS is used, an SFF needs to identify an OAM packet with the SFP scope. To achieve that, this specification first defines the use of a base special-purpose label (bSPL) [RFC3032] or an extended special-purpose label (eSPL) [RFC7274] (referred to in this document as SPL Unit) with the basic unit defined in [RFC8595]. And based on that, the use of Generic Associated Channel Label (GAL) [RFC5586] with the basic unit in the SFC-MPLS environment.

Special-purpose label (SPL), whether bSPL or eSPL, has special significance in the data and control planes. An ability to use an SPL in the basic unit allows for a closer functional match between the NSH-based SFC and SFC-MPLS. For example, Entropy Label Indicator (ELI) [RFC6790] with the basic unit can be used as the Flow ID TLV [I-D.ietf-sfc-nsh-tlv] to allow an SFF to balance SFC flows among SFs of the same type. An SPL MAY be used with the basic unit in SFC-MPLS, as displayed in Figure 1. Note that an SPL unit MAY be present in one or more basic units when MPLS label stacking is used to carry the SFC information.

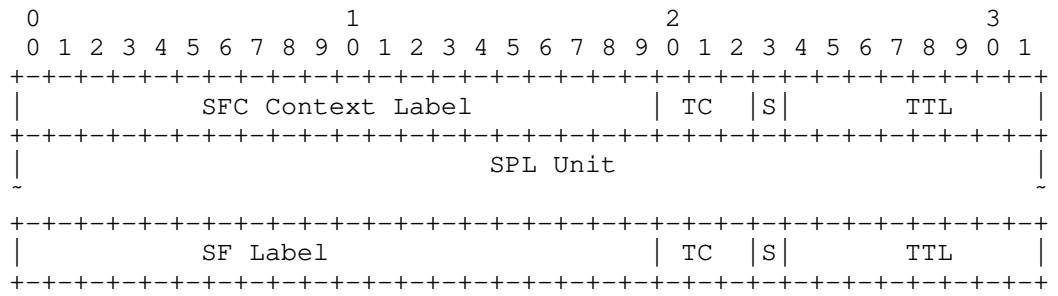


Figure 1: Special-purpose Label Unit with the Basic Unit of MPLS Label Stack for SFC

4.1.1. G-ACh over SFC-MPLS

SFC-MPLS environment could include instances of an SF (SFI) or SFC proxies that cannot properly process control and/or management protocol messages that are exchanged between nodes over the G-ACh associated with the particular SFP. To support OAM over G-ACh, it is beneficial to avoid handing over a test packet to the SFI or SFC proxy. Hence, this specification defines that if the Generic Associated Channel Label (GAL) immediately follows the SFC Context label [RFC8595], then the packet is recognized as an SFP OAM packet.

Below are the processing rules of an SFP OAM packet by an SFF:

- An SFF MUST NOT pass the packet to a local SFI or SFC proxy.
- The SFF MUST decrement SF Label entry's TTL value. If the resulting value equals zero, the SFF MUST pass the SFP OAM packet to the control plane for processing. An implementation that supports this specification MUST provide control to limit the rate of SFP OAM packets passed to the control plane for processing.
- If the TTL value is not zero, the SFP OAM packet is processed as defined in Section 6, Section 7, and Section 8 [RFC8595], according to the type of MPLS forwarding used in the SFP.

4.2. SFC Basic Unit FEC Sub-TLV

Unlike standard MPLS forwarding, based on a single label, packet forwarding defined in [RFC8595] is based on the basic unit of MPLS label stack for SFC(SFC Context Label+SF Label). A new SFC Basic Unit FEC sub-TLV with Type value (TBA1) is defined in this document. The SFC Basic Unit FEC sub-TLV MAY be used to carry the corresponding FEC of the basic unit.

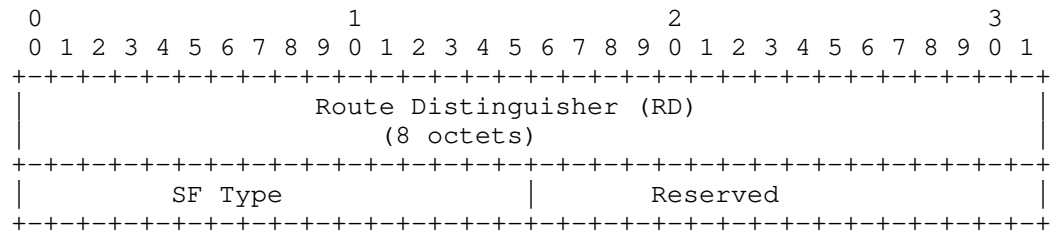


Figure 2: SFC Basic Unit sub-TLV

The format of the basic unit sub-TLV is shown in Figure 2 and includes the following fields:

Route Distinguisher (RD): 8 octets field in SFIR Route Type specific NLRI [I-D.ietf-bess-nsh-bgp-control-plane].

SF Type: 2 octets. It is defined in [I-D.ietf-bess-nsh-bgp-control-plane] and indicates the type of SF, such as DPI, firewall, etc.

Note: [I-D.ietf-bess-nsh-bgp-control-plane] covers the BGP control plane of MPLS-SFC as well.

A node that receives an LSP ping with the Target FEC Stack TLV and the SFC Basic Unit FEC Sub-TLV included will check if it is its Route Distinguisher and whether it advertised that Service Function Type. If the validation is not passed, the SFF will generate an MPLS echo reply with an error code as defined in [RFC8029].

4.3. SFC Basic Unit Nil FEC Sub-TLV

[RFC8029] is based on the premise that one label corresponds to one FEC sub-TLV. For example, in [RFC8029] section 4.4 step 4, before the FEC validation process of an intermediate node first the node should determine FEC-stack-depth from the Downstream Detailed Mapping TLV, and then if the number of FECs in the FEC stack is greater than or equal to FEC-stack-depth, FEC validation is triggered.

In SFC-MPLS OAM, since one basic unit is related to only one FEC sub-TLV, there may be situations that the label stack in Downstream Detailed Mapping TLV contains two labels, but there is only one FEC in the FEC stack.

The SFC Basic Unit Nil Sub-TLV(TBA2) is introduced in this document to ensure that the proper validation can still be performed.

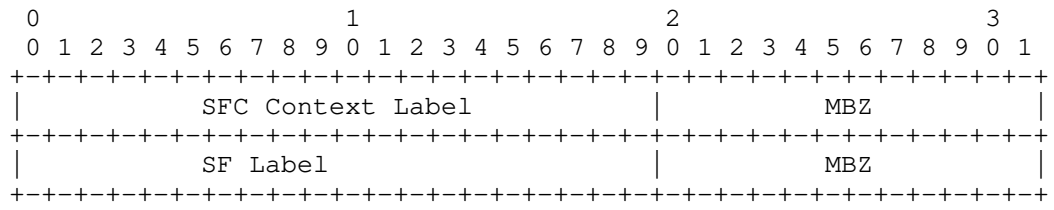


Figure 3: SFC Basic Unit Nil sub-TLV

SFC Context Label and SF Label are the actual label values inserted in the label stack; the MBZ fields MUST be zero when sent and ignored on receipt.

The SFC Basic Unit Nil sub-TLV, when present, MUST be immediately followed by an SFC Basic Unit sub-TLV. During FEC validation, an SFF should skip the SFC Basic Unit Nil sub-TLV and use the following SFC Basic Unit sub-TLV to validate the FEC of the basic unit.

4.4. Theory of Operation

An MPLS SFC validation request is an MPLS echo request with an SFC validation TLV, and the echo request is sent with a label stack corresponding to the SFP being tested. To trace SFC-MPLS, the Generic Associated Channel Label (GAL), which immediately follows the SFC Context label is also included.

If FEC validation is required, the SFC Basic Unit sub-TLV SHOULD be carried in the FEC stack of the request packet, and the SFC Basic Unit Nil sub-TLV MAY also be carried. A Downstream Detailed Mapping TLV MAY be included in the MPLS echo request of the SFP.

Sending an SFC echo request to the control plane is triggered by one of the following packet processing exceptions: IP TTL expiration, MPLS TTL expiration, or the receiver is SFP's egress SFF.

As described in Section 4.1.1, the packet with GAL is recognized by the SFF as an SFP OAM packet. The SFF then decrements the SF Label entry's TTL value. If the resulting value equals zero, the SFF passes the SFP OAM packet to the control plane for processing. The system that supports this specification then generates a reply message.

In "traceroute" mode the TTL of the SF Label is set successively to 1, 2, and so on. After all SFFs on the SFP send back MPLS echo reply, the sender collects information about all traversed SFFs and

SFs on the rendered service path (RSP). But the TTL processing in SR-MPLS is defined in Section 6 of [RFC8595], as follows:

> If an SFF decrements the TTL to zero, it MUST NOT send the packet and MUST discard the packet

and it excludes TTL expiration as the exception mechanism. As a result, tracing a path of an SFC-MPLS-based service chain is problematic. To support the tracing of an SFC, it must be changed to allow punting an OAM packet to the control plane though under throttling control. Hence, this document updates Section 6 of [RFC8595] to state that:

> If an SFF decrements the TTL to zero, an OAM packet MAY be sent to the control plane given it does not exceed the configured rate intended to protect the system from the possible denial-of-service attack.

5. LSP Ping in SR-SFC

In SR service programming, the packet forwarding decision is made based on every single SID/label. The SR proxy SHOULD process the OAM packet for the SF when the SF is not capable of doing so.

If only the SFP connectivity check is required, the current LSP Ping for SR-MPLS [RFC8287] is sufficient.

If operators want to check more information about the SFP(service segment related SF type, SR proxy type, etc.), new FEC sub-TLVs for the service segment should be defined. Details of the new FEC sub-TLVs will be added in the further version.

6. Security Considerations

This specification defines the processing of an SFP OAM packet. Such packets could be used as an attack vector. A system that supports this specification MUST provide control to limit the rate of SFP OAM packets sent to the control plane for processing.

This document defines additional MPLS LSP Ping sub-TLVs and follows the mechanisms defined in [RFC8029]. All the security considerations defined in [RFC8029] will be applicable for this document.

7. IANA Considerations

This document requests assigning two new sub-TLVs from the "sub-TLVs for TLV Types 1, 16, and 21" sub-registry of the "Multi-Protocol

Label Switching(MPLS) Label Switched Paths (LSPs) Ping Parameters" registry according to Table 1

Value	Description	Reference
TBA1	SFC Basic Unit	This document
TBA2	SFC Basic Unit Nil	This document

Table 1: Sub-TLV Values

8. References

8.1. Normative References

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", draft-ietf-bess-nsh-bgp-control-plane-18 (work in progress), August 2020.

[I-D.ietf-spring-nsh-sr]
Guichard, J. and J. Tantsura, "Integration of Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", draft-ietf-spring-nsh-sr-04 (work in progress), December 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

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

[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>.

[RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <https://www.rfc-editor.org/info/rfc5586>.

[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>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <https://www.rfc-editor.org/info/rfc8595>.

8.2. Informative References

[I-D.ietf-sfc-nsh-tlv] Wei, Y., Elzur, U., Majee, S., and C. Pignataro, "Network Service Header Metadata Type 2 Variable-Length Context Headers", draft-ietf-sfc-nsh-tlv-04 (work in progress), January 2021.

[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>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Liu Yao
ZTE Corporation
Nanjing
China

Email: liu.yao71@zte.com.cn

Greg Mirsky
ZTE Corporation

Email: gregory.mirsky@ztetx.com

Routing area
Internet-Draft
Intended status: Standards Track
Expires: October 18, 2021

S. Hegde
K. Arora
M. Srivastava
Juniper Networks Inc.
S. Ninan
Individual Contributor
N. Kumar
Cisco Systems, Inc.
April 16, 2021

PMS/Head-end based MPLS Ping and Traceroute in Inter-domain SR Networks
draft-ninan-mpls-spring-inter-domain-oam-02

Abstract

Segment Routing (SR) architecture leverages source routing and tunneling paradigms and can be directly applied to the use of a Multiprotocol Label Switching (MPLS) data plane. A network may consist of multiple IGP domains or multiple ASes under the control of same organization. It is useful to have the LSP Ping and traceroute procedures when an SR end-to-end path spans across multiple ASes or domains. This document describes mechanisms to facilitae LSP ping and traceroute in inter-AS/inter-domain SR networks in an efficient manner with simple OAM protocol extension which uses dataplane forwarding alone for sending Echo-Reply.

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 October 18, 2021.

Table of Contents

1. Introduction

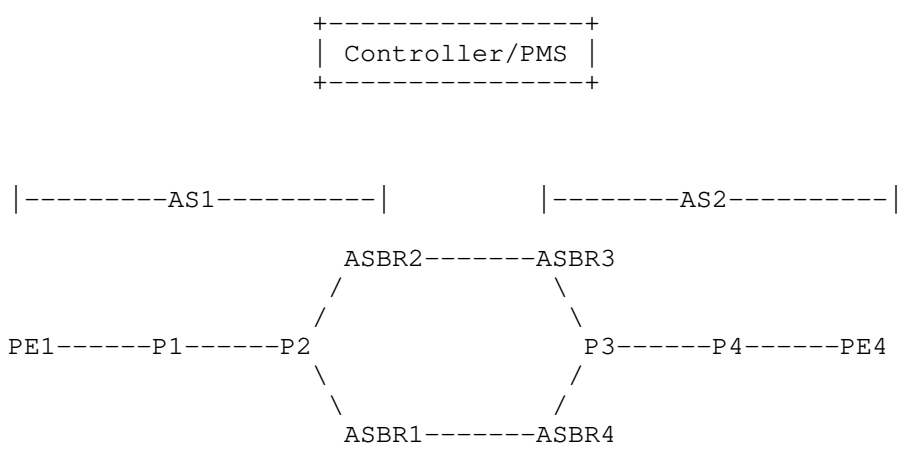


Figure 1: Inter-AS Segment Routing topology

Many network deployments have built their networks consisting of multiple Autonomous Systems either for ease of operations or as a result of network mergers and acquisitions. Segment Routing can be deployed in such scenarios to provide end to end paths, traversing multiple Autonomous systems(AS). These paths consist of Segment Identifiers(SID) of different type as per [RFC8402].

[RFC8660] specifies the forwarding plane behaviour to allow Segment Routing to operate on top of MPLS data plane. [I-D.ietf-spring-segment-routing-central-epe] describes BGP peering SIDs, which will help in steering packet from one Autonomous system to another. Using above SR capabilities, paths which span across multiple Autonomous systems can be created.

For example Figure 1 describes an inter-AS network scenario consisting of ASes AS1 and AS2. Both AS1 and AS2 are Segment Routing enabled and the EPE links have EPE labels configured and advertised via [I-D.ietf-idr-bgpls-segment-routing-epe]. Controller or head-end can build end-to-end Traffic-Engineered path consisting of Node-SIDs, Adjacency-SIDs and EPE-SIDs. It is advantageous for operations to be able to perform LSP ping and traceroute procedures on these inter-AS SR paths. LSP ping/traceroute procedures use ip connectivity for Echo-reply to reach the head-end. In inter-AS networks, ip connectivity may not be there from each router in the path.For example in Figure 1 P3 and P4 may not have ip connectivity for PE1.

[RFC8403] describes mechanisms to carry out the MPLS ping/traceroute from a PMS. It is possible to build GRE tunnels or static routes to each router in the network to get IP connectivity for the reverse path. This mechanism is operationally very heavy and requires PMS to be capable of building huge number of GRE tunnels, which may not be feasible.

It is not possible to carry out LSP ping and Traceroute functionality on these paths to verify basic connectivity and fault isolation using existing LSP ping and Traceroute mechanism([RFC8287] and [RFC8029]). This is because, there exists no IP connectivity to source address of ping packet, which is in a different AS, from the destination of Ping/Traceroute.

[RFC7743] describes a Echo-relay based solution based on advertising a new Relay Node Address Stack TLV containing stack of Echo-relay ip addresses. That mechanism requires the return ping packet to reach the control plane on every relay node.

This document describes a new mechanism which is efficient and simple and can be easily deployed in SR networks. This mechanism uses Return path TLV [RFC7110] to convey the reverse path. Three new sub-TLVs for Return path TLV are defined, that faciliate encoding segment routing label stack. The TLV can either be derived by a smart application or controller which has a full topology view. This document also proposes mechanisms to derive the Return path dynamically during traceroute procedures.

2. Inter domain networks with multiple IGPs

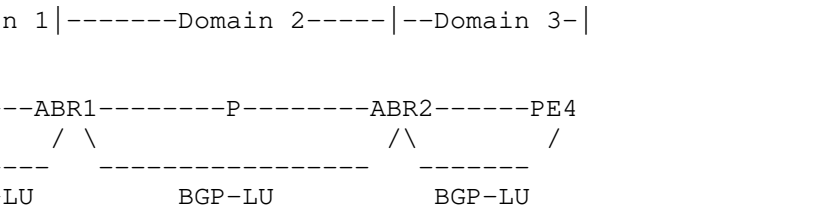


Figure 2: Inter-domain networks with multiple IGPs

When the network consists of large number of nodes, the nodes are seggregated into multiple IGP domains. The connectivity to the remote PEs can be achieved using BGP-LU [RFC3107] or by stacking the labels for each domain as described in [RFC8604]. It is useful to support mpls ping and traceroute mechanisms for these networks. The

procedures described in this document for constructing Return path TLV and its use in echo-reply is equally applicable to networks consisting of multiple IGP domains that use BGP-LU or label stacking.

3. Return Path TLV

Segment Routing networks statically assign the labels to nodes and PMS/Head-end may know the entire database. The reverse path can be built from PMS/Head-end by stacking segments for the reverse path. Return path TLV as defined in [RFC7110] is used to carry the return path. While using the procedures described in this document, the reply mode MUST be set to 5 and Return Path TLV MUST be included in the Echo-Request message. The procedures decribed in [RFC7110] are applicable for constructing the Return Path TLV. This document define three new sub-TLVs to encode the Segment Routing path

The type of segment that the head-end chooses to send in the Return Path TLV is governed by local policy.

Some networks may consist of pure IPV4 domains and Pure IPv6 domains. Handling end-to-end MPLS OAM for such networks is out of scope for this document. It is recommended to use dual stack in such cases and use end-to-end IPv6 addresses for MPLS ping and trace route procedures.

4. Segment sub-TLV

[I-D.ietf-spring-segment-routing-policy] defines various types of segments. The segments applicable to this documents have been re-defined here. One or more segment sub-TLV can be included in the Return Path TLV. The segment sub-TLVs included in a Return Path TLV MAY be of different types.

Below types of segment sub-TLVs are applicable for the Reverse Path Segment List TLV.

Type 1: SID only, in the form of MPLS Label

Type 3: IPv4 Node Address with optional SID

Type 4: IPv6 Node Address with optional SID for SR MPLS

4.1. Type 1: SID only, in the form of MPLS Label

The Type-1 Segment Sub-TLV encodes a single SID in the form of an MPLS label. The format is as follows:

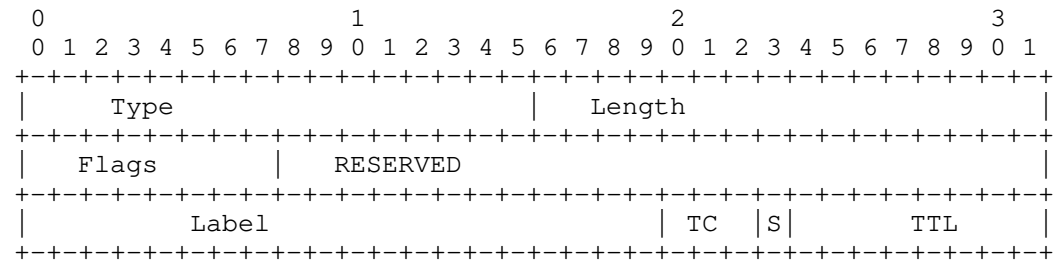


Figure 3: Type 1 Segment sub-TLV

where:

Type: TBD1(to be assigned by IANA from the registry "Sub-TLV Target FEC stack TLV").

Length is 8.

Flags: 1 octet of flags as defined in Section Section 4.4.

RESERVED: 3 octets of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

Label: 20 bits of label value.

TC: 3 bits of traffic class

S: 1 bit of bottom-of-stack.

TTL: 1 octet of TTL.

The following applies to the Type-1 Segment sub-TLV:

The S bit SHOULD be zero upon transmission, and MUST be ignored upon reception.

If the originator wants the receiver to choose the TC value, it sets the TC field to zero.

If the originator wants the receiver to choose the TTL value, it sets the TTL field to 255.

If the originator wants to recommend a value for these fields, it puts those values in the TC and/or TTL fields.

The receiver MAY override the originator's values for these fields. This would be determined by local policy at the receiver. One possible policy would be to override the fields only if the fields have the default values specified above.

4.2. Type 3: IPv4 Node Address with optional SID for SR-MPLS

The Type-3 Segment Sub-TLV encodes an IPv4 node address, SR Algorithm and an optional SID in the form of an MPLS label. The format is as follows:

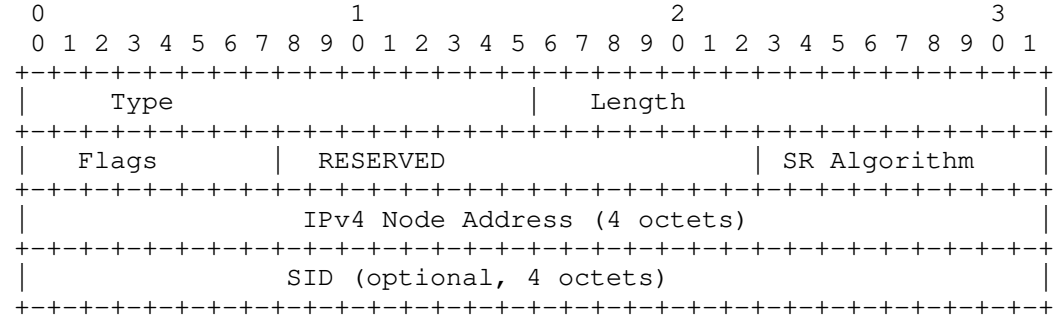


Figure 4: Type 3 Segment sub-TLV

where:

Type: TBD3(to be assigned by IANA from the registry "Sub-TLV Target FEC stack TLV").

Length is 8 or 12.

Flags: 1 octet of flags as defined in Section Section 4.4.

SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section Section 4.4is present. SR Algorithm is used by the receiver to derive the Label. When A-Flag is not encoded, this field SHOULD be unset on transmission and MUST be ignored on receipt.

RESERVED: 2 octets of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

IPv4 Node Address: a 4 octet IPv4 address representing a node.

SID: 4 octet MPLS label.

The following applies to the Type-3 Segment sub-TLV:

The IPv4 Node Address MUST be present.

The SID is optional and specifies a 4 octet MPLS SID containing label, TC, S and TTL as defined in Section Section 4.1.

If length is 8, then only the IPv4 Node Address is present.

If length is 12, then the IPv4 Node Address and the MPLS SID are present.

4.3. Type 4: IPv6 Node Address with optional SID for SR MPLS

The Type-4 Segment Sub-TLV encodes an IPv6 node address, SR Algorithm and an optional SID in the form of an MPLS label. The format is as follows:

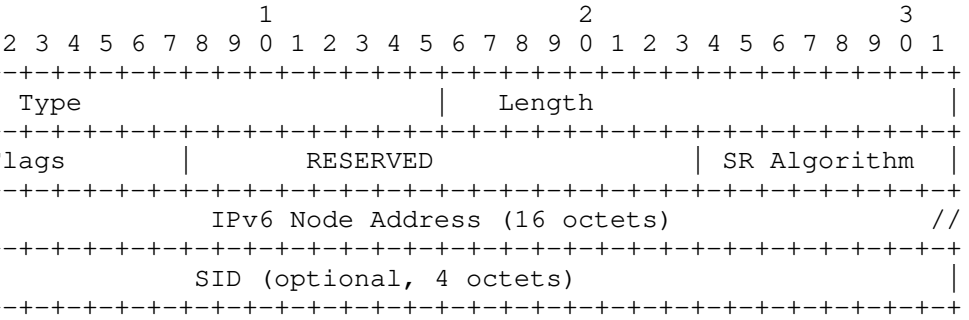


Figure 5: Type 4 Segment sub-TLV

where:

Type: TBD4(to be assigned by IANA from the registry "Sub-TLV Target FEC stack TLV").

Length is 20 or 24.

Flags: 1 octet of flags as defined in Section Section 4.4.

SR Algorithm: 1 octet specifying SR Algorithm as described in section 3.1.1 in [RFC8402], when A-Flag as defined in Section Section 4.4 is present. SR Algorithm is used by the receiver to derive the label. When A-Flag is not encoded, this field SHOULD be unset on transmission and MUST be ignored on receipt.

RESERVED: 2 octets of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.

IPv6 Node Address: a 16 octet IPv6 address representing a node.

SID: 4 octet MPLS label.

The following applies to the Type-4 Segment sub-TLV:

The IPv6 Node Address MUST be present.

The SID is optional and specifies a 4 octet MPLS SID containing label, TC, S and TTL as defined in Section Section 4.1 .

If length is 20, then only the IPv6 Node Address is present.

If length is 24, then the IPv6 Node Address and the MPLS SID are present.

4.4. Segment Flags

The Segment Types described above MAY contain following flags in the "Flags" field (codes to be assigned by IANA from the registry "Return path sub-TLV Flags")

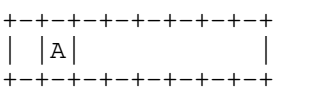


Figure 6: Flags

where:

A-Flag: This flag indicates the presence of SR Algorithm id in the "SR Algorithm" field applicable to various Segment Types.

Unused bits in the Flag octet SHOULD be set to zero upon transmission and MUST be ignored upon receipt.

The following applies to the Segment Flags:

A-Flag is applicable to Segment Types 3, 4. If A-Flag appears with any other Segment Type, it MUST be ignored.

5. SRv6 Dataplane

SRv6 dataplane is not in the scope of this document and will be addressed in a separate document.

6. Detailed Procedures

6.1. Sending an Echo-Request

In the inter-AS scenario when there is no reverse path connectivity, LSP ping initiator MUST add a Return Path TLV in the Echo-request message. The reverse Segment List MUST correspond to the return path from the egress. The Return Path TLV must contain the Segment Routing Path in the reverse direction encoded as an ordered list of Segments. The first Segment MUST correspond to the top Segment in MPLS header that the responder MUST use while sending the Echo-reply.

6.2. Receiving an Echo-Request

As described in [RFC7110], when Reply mode is set to 5 (Reply via Specified Path), The Echo-Request MUST contain the Return path TLV. Absence of Reply path TLV is treated as malformed Echo-Request.When a Return Path TLV is received, and the responder that supports processing it, it MUST use the Segments in Return Path TLV to build the echo-reply. The responder MUST follow the normal FEC validation procedures as described in [RFC8029] and [RFC8287] and this document does not suggest any change to those procedures. When the Echo-reply has to be sent out the Return Path TLV is used to construct the MPLS packet to send out.

6.3. Sending an Echo-Reply

The Echo-Reply message is sent as MPLS packet with a MPLS label stack. The Echo-Reply message MUST be constructed as described in the [RFC8029]. An MPLS packet is constructed with Echo-reply in the payload. The top label MUST be constructed from the first Segment from the Return Path TLV. The remaining labels MUST follow the order from the Return Path TLV. The responder MAY check the reachability of the top label in its own LFIB before sending the Echo-Reply. In certain scenarios the head-end may choose to send Type 2/Type 3 segments consisting of IPV4 address and SID. In such cases that node sending the echo-reply MUST derive the MPLS labels from SIDs based on SRGB and encode the echo-reply with MPLS labels.

The return code MUST be set as described in [RFC7110]. The Return Path TLV MUST be included in Echo-Reply indicating the specified return path that the echo reply message is required to follow.

7. Detailed Example

An example topology is given in Figure 1 . This will be used in below sections to explain LSP Ping and Traceroute procedures. The PMS/Head-end has complete view of topology. PE1, P1, P2, ASBR1 and ASBR2 are in AS1. Similarly ASBR3, ASBR4, P3, P4 and PE4 are in AS2.

AS1 and AS2 have Segment Routing enabled. IGPs like OSPF/ISIS are used to flood SIDs in each Autonomous System. The ASBR1, ASBR2, ASBR3, ASBR4 advertise BGP EPE SIDs for the inter-AS links. Topology of AS1 and AS2 are advertised via BGP-LS to the controller/PMS or Head-end node. The EPE-SIDs are also advertised via BGP-LS as described in [I-D.ietf-idr-bgpls-segment-routing-epe]

The description in the document uses below notations for Segment Identifiers(SIDs).

Node SIDs : N-PE1, N-P1, N-ASBR1 etc.

Adjacency SIDs : Adj-PE1-P1, Adj-P1-P2 etc.

EPE SIDS : EPE-ASBR2-ASBR3, EPE-ASBR1-ASBR4, EPE-ASBR3-ASBR2 etc.

Let us consider a traffic engineered path built from PE1 to PE4 with Segment List stack as below. N-P1, N-ASBR1, EPE-ASBR1-ASBR4, N-PE4 for following procedures. This stack may be programmed by controller/PMS or Head-end router PE1 may have imported the whole topology information from BGP-LS and computed the inter-AS path.

7.1. Procedures for Segment Routing LSP ping

To perform LSP ping procedure on an SR-Path from PE1 to PE4 consisting of label stacks [N-P1,N-ASBR1,EPE-ASBR1-ASBR4, N-PE4], The remote end(PE4) needs IP connectivity to head end(PE1) for the Segment Routing ping to succeed, because Echo-reply needs to travel back to PE1 from PE4. But in typical deployment scenario there will be no ip route from PE4 to PE1 as they belong to different ASes.

PE1 adds Return Path from PE4 to PE1 in the MPLS Echo-request using multiple Segments in "Return Path TLV" as defined above. An example return path Segment List for PE1 to PE4 for LSP ping is [N-ASBR4, EPE-ASBR4-ASBR1, N-PE1]. An implementation may also build a Return Path consisting of labels to reach its own AS. Once the label stack is popped-off the Echo-reply message will be exposed.The further packet forwarding will be based on ip lookup. An example Return Path for this case could be [N-ASBR4, EPE-ASBR4-ASBR1].

On receiving MPLS Echo-request PE4 first validates FEC in the Echo-request. PE4 then builds label stack to send the response from PE4 to PE1 by copying the labels from "Return Path TLV". PE4 builds the Echo-reply packet with the MPLS label stack constructed and imposes MPLS headers on top of Echo-reply packet and sends out the packet towards PE1. This Segment List stack can successfully steer reply back to Head-end node(PE1).

7.2. Procedures for Segment Routing LSP Traceroute

As described in the procedures for LSP ping, the reverse Segment List may be sent from head-end in which case the LSP Traceroute procedures are similar to LSP ping. The head-end constructs the Return Path TLV and the egress node uses the Return Path to construct the Echo-reply packet header. Head-end/PMS is aware of the return path from every node visited in the network and builds the Return Path segment list for every visited node accordingly.

For Example:

For the same traffic engineered path PE1 to PE4 mentioned in above sections, let us assume there is no reverse path available from the nodes ASBR4 to PE1. During the Traceroute procedure, when PE1 has to visit ASBR4, it builds return Path TLV and includes label to the border-node which has the route to, PE1. In this example the Return Path TLV will contain [EPE-ASBR4-ASBR1]. Further down the traceroute procedure when P3 or P4 node is being visited, PE1 build the Return Path TLV containing [N-ASBR4, EPE-ASBR4-ASBR1]. The Echo-reply will be an MPLS packet with this label stack and will be forwarded to PE1.

8. Building Reverse Path Segment List TLV dynamically

In some cases, the head-end may not have complete visibility of Inter-AS topology. In such cases, it can rely on downstream routers to build the reverse path for mpls traceroute procedures. For this purpose, new return code is defined which implies the Return Path TLV in the Echo-reply corresponds to the return path to be used in next Echo-Request.

Value	Meaning
0x0006	Use Return Path TLV in Echo-Reply for next Echo-Request.

Figure 7: Return Code

8.1. The procedures to build the reverse path

When an ASBR receives an echo-request from another AS, and ASBR is configured to build the return path dynamically, ASBR MUST build a Return Path TLV that should be used in next Echo-request and add it in echo-reply. ASBR MUST locally decide the outgoing interface for the echo-reply packet. Generally, remote ASBR will choose interface on which the incoming OAM packet was receieved to send the echo-reply out. Return Path TLV is built by adding two segment sub TLVs. The top segment sub TLV consists of the ASBR's Node SID and second segment consists of the EPE SID in the reverse direction to reach the AS from which the OAM packet was received.The type of segment chosen to build Return Path TLV is implementation dependent. In cases where the AS is configured with different SRGBs, the Node SID of the ASBR should be represented using type 3 segment so that all the nodes inside the AS can correctly translate the Node-SID to a label.

Irrespective of which type of segment is included in the Return Path TLV, the responder of echo-request always translates the Return Path TLV to a label stack and builds MPLS header for the the echo-reply packet. This procedure can be applied to an end-to-end path consisting of multiple ASes. Each ASBR at the border adds its Node-SID and EPE-SID on top of existing segments in the Return Path TLV.

8.2. Details with example

Let us consider a traffic engineered path built from PE1 to PE4 with a label stack as below. N-P1, N-ASBR1, EPE-ASBR1-ASBR4, N-PE4 for the following procedures. This traceroute doesn't need any changes to the Return Path TLV till it leaves AS1. The same Return Path TLV that is received is included in the Echo-Reply. But this traceroute requires changes to Return Path TLV once it starts probing AS2 routers. ASBR4 should form this Return Path TLV using its own Node SID(N-ASBR4) and EPE SID (EPE-ASRB4-ASBR1) labels and set the return code to 6. Then PE1 should use this Return Path TLV in subsequent echo-requests. In this example, when the subsequent echo-request reaches P3, it should use this Return Path TLV for sending the echo-reply. The same Return Path TLV is enough for any router in AS2 to send the reply. Because the first label(N-ASBR4) can direct echo-reply to ASBR4 and second one (EPE-ASBR4-ASBR1) to direct echo-reply to AS1. Once echo reply reaches AS1, normal IP forwarding helps it to reach PE1 or the head-end.

9. Security Considerations

TBD

10. IANA Considerations

Sub-TLVs for TLV Types 1, 16, and 21

SID only in the form of mpls label : TBD (Range 32768-65535)

IPv4 Node Address with optional SID for SR-MPLS : TBD (Range 32768-65535)

IPv6 Node Address with optional SID for SR-MPLS : TBD (Range 32768-65535)

11. Contributors

1.Carlos Pignataro

Cisco Systems, Inc.

cpignata@cisco.com

2. Zafar Ali

Cisco Systems, Inc.

zali@cisco.com

12. Acknowledgments

Thanks to Bruno Decreane for suggesting use of generic Segment sub-TLV. Thanks to Adrian Farrel for careful review and comments. Thanks to Mach Chen for suggesting to use Return Path TLV.

13. References

13.1. Normative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-11 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", draft-ietf-spring-segment-routing-central-epe-10 (work in progress), December 2017.

[RFC7110] Chen, M., Cao, W., Ning, S., Jounay, F., and S. Delord, "Return Path Specified Label Switched Path (LSP) Ping", RFC 7110, DOI 10.17487/RFC7110, January 2014, <https://www.rfc-editor.org/info/rfc7110>.

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

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

13.2. Informative References

[I-D.ietf-idr-bgpls-segment-routing-epe] Previdi, S., Talaulikar, K., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-19 (work in progress), May 2019.

[I-D.ietf-mpls-interas-lspping] Nadeau, T. and G. Swallow, "Detecting MPLS Data Plane Failures in Inter-AS and inter-provider Scenarios", draft-ietf-mpls-interas-lspping-00 (work in progress), March 2007.

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

[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>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <https://www.rfc-editor.org/info/rfc3107>.

[RFC7743] Luo, J., Ed., Jin, L., Ed., Nadeau, T., Ed., and G. Swallow, Ed., "Relayed Echo Reply Mechanism for Label Switched Path (LSP) Ping", RFC 7743, DOI 10.17487/RFC7743, January 2016, <https://www.rfc-editor.org/info/rfc7743>.

[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>.

[RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <https://www.rfc-editor.org/info/rfc8403>.

[RFC8604] Filsfils, C., Ed., Previdi, S., Dawra, G., Ed., Henderickx, W., and D. Cooper, "Interconnecting Millions of Endpoints with Segment Routing", RFC 8604, DOI 10.17487/RFC8604, June 2019, <https://www.rfc-editor.org/info/rfc8604>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

Authors' Addresses

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Kapil Arora
Juniper Networks Inc.

Email: kapilaro@juniper.net

Mukul Srivastava
Juniper Networks Inc.

Email: msri@juniper.net

Samson Ninan
Individual Contributor

Email: samson.cse@gmail.com

Nagendra Kumar
Cisco Systems, Inc.

Email: naikumar@cisco.com

Internet
Internet-Draft
Intended status: Standards Track
Expires: August 8, 2021

Y. Qu
Futurewei
A. Lindem
S. Litkowski
Cisco Systems
J. Tantsura
Juniper

February 4, 2021

A YANG Model for MPLS MSD
draft-qu-mpls-mpls-msd-yang-00

Abstract

This document defines a YANG data module augmenting the IETF MPLS YANG model to provide support for MPLS Maximum SID Depths (MSDs) as defined RFC 8476 and RFC 8491.

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 8, 2021.

Copyright Notice

Table of Contents

1. Overview

YANG [RFC6020] [RFC7950] is a data definition language used to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g., ReST) and encodings other than XML (e.g., JSON) are being defined. Furthermore, YANG data models can be used as the basis for implementation of other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data module augmenting the IETF MPLS YANG model [RFC8960], which itself augments [RFC8349], to provide operational state for various MSDs[RFC8662].

The augmentation defined in this document requires support for the MPLS base model[RFC8960] which defines basic MPLS configuration and state.

The YANG module in this document 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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. YANG Module for MPLS MSD

This document defines a YANG module for MSD extensions [RFC8476][RFC8491] to MPLS base model as defined in [RFC8960].

```
module: ietf-mpls-msd
  augment /rt:routing/mpls:mpls:
    +--ro msd
       +--ro node-msds* [msd-type]
       |  +--ro msd-type     identityref
       |  +--ro msd-value?   uint8
       +--ro link-msds* [interface]
          +--ro interface    if:interface-ref
          +--ro link-msd* [msd-type]
             +--ro msd-type     identityref
             +--ro msd-value?   uint8
```

```
<CODE BEGINS> file "ietf-mpls-msd@2021-02-04.yang"
module ietf-mpls-msd {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-msd";
  prefix mpls-msd;

  import ietf-routing {
    prefix rt;
    reference "RFC 8349: A YANG Data Model for Routing
               Management (NMDA Version)";
  }

  import ietf-interfaces {
    prefix if;
    reference "RFC 8343: A YANG Data Model for Interface
                         Management (NMDA Version)";
  }

  import ietf-mpls {
    prefix mpls;
    reference "RFC 8960: A YANG Data Model for MPLS Base";
  }

  organization
    "IETF MPLS - MPLS Working Group";
  contact
    "WG Web:   <https://tools.ietf.org/wg/mpls/>
     WG List:  <mailto:mpls@ietf.org>

     Author:    Yingzhen Qu
                <mailto:yingzhen.qu@futurewei.com>
```

```
      Author:    Acee Lindem
                 <mailto:acee@cisco.com>
      Author:    Stephane Litkowski
                 <mailto:slitkows.ietf@gmail.com>
      Author:    Jeff Tantsura
                 <jefftant.ietf@gmail.com>

    ";
  description
    "The YANG module augments the base MPLS model, and it is to
     manage different types of MSDs.

     This YANG model conforms to the Network Management
     Datastore Architecture (NMDA) as described in RFC 8342.

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

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

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

     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 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.";


  reference "RFC XXXX: YANG Data Model for Segment Routing.";

  revision 2021-02-04 {
    description
      "Initial Version";
    reference "RFC XXXX: YANG Data Model for Segment Routing.";
  }

  identity msd-base-type {
    description
      "Base identity for MSD Type";
  }
```

```
identity base-mpls-msd {
  base msd-base-type;
  description
    "Base MPLS Imposition MSD.";
  reference
    "RFC 8491: Singling MSD using IS-IS.";
}

identity erld-msd {
  base msd-base-type;
  description
    "ERLD-MSD is defined to advertise the ERLD.";
  reference
    "RFC 8662: Entropy Label for Source Packet Routing in
                Networking (SPRING) Tunnels";
}

grouping max-sid-depth {
  description
    "Maximum SID Depth (MSD) grouping.";
  list node-msds {
    key "msd-type";
    leaf msd-type {
      type identityref {
      base msd-base-type;
      }
      description
        "MSD-Types";
    }
    leaf msd-value {
      type uint8;
      description
        "MSD value, in the range of 0-255.";
    }
    description
      "Node MSD is the smallest link MSD supported by
       the node.";
  }
  list link-msds {
    key "interface";
    leaf interface {
      type if:interface-ref;
      description
        "Reference to device interface.";
    }
    list link-msd {
      key "msd-type";
      leaf msd-type {
```

```
            type identityref {
              base msd-base-type;
            }
            description
              "MSD-Types";
          }
          leaf msd-value {
            type uint8;
            description
              "MSD value, in the range of 0-255.";
          }
          description
            "List of link MSDs";
        }
        description
          "MSD supported by an individual interface.";
      }
    }

    augment "/rt:routing/mpls:mpls" {
      description
        "This module augments MPLS data model (RFC 8960)
         with MSD.";
      container msd {
        config false;
        description
          "Maximum SID Depth (MSD) operational state.";
        uses max-sid-depth;
      }
    }
  }
    <CODE ENDS>
```

3. Security Considerations

The YANG modules specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

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

Some of the readable data nodes in the modules 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. The exposure of the Link State Database (LSDB) will expose the detailed topology of the network. This may be undesirable since both due to the fact that exposure may facilitate other attacks. Additionally, network operators may consider their topologies to be sensitive confidential data.

4. IANA Considerations

This document registers URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations is requested to be made:

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

This document registers the YANG modules in the YANG Module Names registry [RFC6020].

```
name: ietf-mpls-msd
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls-msd
prefix: mpls-msd
reference: RFC XXXX
```

5. Acknowledgements

This document was produced using Marshall Rose’s xml2rfc tool.

The YANG model was developed using the suite of YANG tools written and maintained by numerous authors.

6. References

6.1. Normative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <https://www.rfc-editor.org/info/rfc8476>.

[RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <https://www.rfc-editor.org/info/rfc8491>.

[RFC8960] Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", RFC 8960, DOI 10.17487/RFC8960, December 2020, <https://www.rfc-editor.org/info/rfc8960>.

6.2. Informative References

[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

Yingzhen Qu
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: yingzhen.qu@futurewei.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513

EMail: acee@cisco.com

Stephane Litkowski
Cisco Systems

EMail: slitkows.ietf@gmail.com

Jeff Tantsura
Juniper

EMail: jefftant.ietf@gmail.com

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

D. Rathi, Ed.
K. Arora
S. Hegde
Juniper Networks Inc.
Z. Ali
N. Nainar
Cisco Systems, Inc.

February 21, 2021

Egress TLV for Nil FEC in Label Switched Path Ping and Traceroute Mechanisms

draft-rathi-mpls-egress-tlv-for-nil-fec-04

Abstract

MPLS ping and traceroute mechanism as described in RFC 8029 and related extensions for SR as defined in RFC 8287 is very useful to precisely validate the control plane and data plane synchronization. All intermediate nodes may not have been upgraded to support validation procedures. A simple mpls ping and traceroute mechanism comprises of ability to traverse any path without having to validate the control plane state. RFC 8029 supports this mechanism with Nil FEC. The procedures described in RFC 8029 are mostly applicable when the Nil FEC is used as intermediate FEC in the label stack. When all labels are represented using Nil FEC, it poses some challenges.

This document introduces new TLV as additional extension to exisiting Nil FEC and describes mpls ping and traceroute procedures using Nil FEC with this additional extensions to overcome these challenges.

Requirements Language

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

Status of This Memo

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

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

Table of Contents

1. Introduction

Segment routing supports the creation of explicit paths using adjacency- sids, node-sids, and anycast-sids. In certain usecases, the TE paths are built using mechanisms described in [I.D-ietf-spring-segment-routing-policy] by stacking the labels that represent the nodes and links in the explicit path. When the SR-TE paths are built by the controller, the head-end routers may not have

the complete database of the network and may not be aware of the FEC associated with labels that are used in the label stack. A very useful Operations And Maintenance (OAM) requirement is to be able to ping and trace these paths. A simple mpls ping and traceroute mechanism comprises of ability to traverse the SR-TE path without having to validate the control plane state.

MPLS ping and traceroute mechanism as described in [RFC8029] and related extensions for SR as defined in [RFC8287] is very useful to precisely validate the control plane and data plane synchronization. It also provides ability to traverse multiple ECMP paths and validate each of the ECMP paths. Use of Target FEC requires all nodes in the network to have implemented the validation procedures. All intermediate nodes may not have been upgraded to support validation procedures. In such cases, it is useful to have ability to traverse the paths using ping and traceroute without having to obtain the Forwarding Equivalence Class (FEC) for each label.

[RFC8029] supports this mechanism with Nil FEC. Nil FEC consists of the label and there is no other associated FEC information. The procedures described in [RFC8029] are mostly applicable when the Nil FEC is used where the Nil FEC is an intermediate FEC in the label stack. When all labels are represented using Nil FEC, it poses some challenges.

Section 2 discusses the problems associated with using all Nil FECs in a MPLS ping/traceroute procedure and Section 3 and Section 4 discusses simple extensions needed to solve the problem.

2. Problem with Nil FEC

The purpose of Nil FEC as described in [RFC8029] is to ensure hiding of transit tunnel information and in some cases to avoid false negatives when the FEC information is not known.

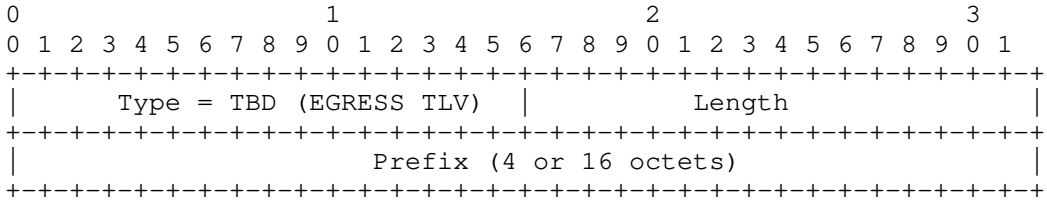
The MPLS ping/traceroute packet consists of only single Nil FEC corresponding to the complete label stack irrespective of number of segments in the label-stack. When router in the label-stack path receives MPLS ping/traceroute packets, there is no definite way to decide on whether its egress or transit since Nil FEC does not carry any information. So there is high possibility that the packet may be mis-forwarded to incorrect destination but the ping/traceroute might still show success.

To avoid this problem, there is a need to add additional information in the MPLS ping/traceroute packet along with Nil FEC that will help to do needed validation on each router of the label-stack path and sends proper information to ingress router on success and failure.

Thus it will be useful to add egress information in ping/traceroute packet that will help in validating Nil-FEC on each receiving router on label-stack path to ensure the correct destination.

3. Egress TLV

The Egress object is a TLV that MAY be included in an MPLS Echo Request message. Its an optional TLV and should appear before FEC-stack TLV in the MPLS Echo Request packet. In case multiple Nil FEC is present in Target FEC Stack TLV, Egress TLV should be added corresponding to the ultimate egress of the label-stack. It can be use for any kind of path with Egress TLV added corresponding to the end-point of the path. The format is as specified below:



Type : TBD

Length : variable based on IPV4/IPV6 prefix. Length excludes the length of Type and length field. Length will be 4 octets for IPv4 and 16 octets for IPv6.

Prefix : This field carries the valid IPv4 prefix of length 4 octets or valid IPv6 Prefix of length 16 octets. It can be obtained from egress of Nil FEC corresponding to last label in the label-stack or SR-TE policy endpoint field [I.D-ietf-idr-segment-routing-te-policy].

4. Procedure

This section describes aspects of LSP Ping and Traceroute operations that require further considerations beyond [RFC8029].

4.1. Sending Egress TLV in MPLS Echo Request

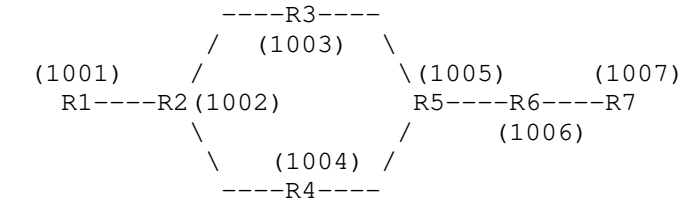
As stated earlier, when the sender node builds a Echo Request with target FEC Stack TLV, Egress TLV SHOULD appear before Target FEC-stack TLV in MPLS Echo Request packet.

Ping

When the sender node builds a Echo Request with target FEC Stack TLV that contains a single NiL FEC corresponding to the last segment of the SR-TE path, sender node MUST add a Egress TLV with prefix obtained from SR-TE policy endpoint field [I.D-ietf-idr-segment-routing-te-policy] to indicate the egress for this Nil FEC in the Echo Request packet. In case endpoint is not specified or is equal to 0, sender MUST use the prefix corresponding to last segment of the SR-TE path as prefix for Egress TLV.

Traceroute

When the sender node builds a Echo Request with target FEC Stack TLV that contains a single NiL FEC corresponding to complete segment-list of the SR-TE path, sender node MUST add a Egress TLV with prefix obtained from SR-TE policy endpoint field [I.D-ietf-idr-segment-routing-te-policy] to indicate the egress for this Nil FEC in the Echo Request packet. In case of multiple Nil FEC, Egress TLV SHOULD be added with prefix that indicate endpoint for last Nil-FEC corresponding to respective segment in label-stack. In case endpoint is not specified or is equal to 0, sender MUST use the prefix corresponding to the last segment endpoint of the SR-TE path i.e. ultimate egress as prefix for Egress TLV.



Consider the SR-TE policy configured with label-stack as 1002, 1004 , 1007 and end point as X on ingress router R1 to reach egress router R3. Segment 1007 belongs to R3 that has prefix X configured on it locally.

In Ping Echo Request, with target FEC Stack TLV that contains a single NiL FEC corresponding to 1007, should add Egress TLV for endpoint X with type as EGRESS-TLV, length depends on if X is IPv4 or IPv6 address and prefix as X.

In Traceroute Echo Request, with target FEC Stack TLV that contains a single NiL FEC corresponding to complete label-stack (1002, 1004, 1007) or multiple Nil-FEC corresponding to each label in label-stack, should add single Egress TLV for endpoint X with type as EGRESS-TLV, length depends on if X is IPv4 or IPv6 address and prefix as X or endpoint of segment 1007. In case X is not present or is set to 0, sender should use endpoint of segment 1007 as prefix for Egress TLV.

4.2. Receiving Egress TLV in MPLS Echo Request

No change in the processing for Nil FEC as defined in [RFC8029] in Target FEC stack TLV Node that receives an MPLS echo request.

Additional processing done for Egress TLV on receiver node as follows:

1. If the Label-stack-depth is greater than 0 and the Target FEC Stack sub-TLV at FEC-stack-depth is Nil FEC, set Best-return-code to 8 ("Label switched at stack-depth") and Best-return-subcode to Label-stack-depth to report transit switching in MPLS Echo Reply message.

2. If the Label-stack-depth is 0 and the Target FEC Stack sub-TLV at FEC-stack-depth is Nil FEC then do the look up for an exact match of the EGRESS TLV prefix to any of locally configured interfaces or loopback addresses.

2a. If EGRESS TLV prefix look up succeeds, set Best-return-code to 3 ("Replying router is an egress for the FEC at stack-depth") and Best-return-subcode to 1 to report egress ok in MPLS Echo Reply message.

2b. If EGRESS TLV prefix look up fails, set the Best-return-code to 10, "Mapping for this FEC is not the given label at stack-depth" and Best-return-subcode to 1.

5. Backward Compatibility

The extension proposed in this document is backward compatible with procedures described in [RFC8029].

6. Security Considerations

TBD

7. IANA Considerations

7.1. New TLV

IANA need to assign new value for EGRESS TLV in the "Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters" TLV registry [IANA].

EGRESS TLV : (TBD)

8. Acknowledgements

TBD.

9. References

9.1. Normative References

[I.D-ietf-idr-segment-routing-te-policy]
Filsfils, C., Ed., Previdi, S., Ed., Talaulikar , K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09, work in progress, may 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-segment-routing-te-policy-09>.

[I.D-ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar , K., Bogdanov, A., Mattes, P., and D. Voyer, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08, work in progress, July 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-segment-routing-policy-08>.

[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>.

9.2. Informative References

[IANA] IANA, "Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Ping Parameters", <http://www.iana.org/assignments/mpls-lsp-ping-parameters>.

Authors' Addresses

Deepti N. Rathi (editor)
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: deeptir@juniper.net

Kapil Arora
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: kapilaro@juniper.net

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Nagendra Kumar Nainar
Cisco Systems, Inc.

Email: naikumar@cisco.com

intarea
Internet-Draft
Intended status: Standards Track
Expires: October 23, 2021

Z. Zhang
R. Bonica
K. Kompella
Juniper Networks
G. Mirsky
ZTE

April 21, 2021

Generic Delivery Functions
draft-zzhang-intarea-generic-delivery-functions-01

Abstract

Some functionalities (e.g., fragmentation/reassembly and Encapsulating Security Payload) provided by IPv6 can be viewed as delivery functions independent of IPv6 or even IP entirely. This document proposes to provide those functionalities at different layers (e.g., MPLS, BIER or even Ethernet) independent of IP.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 23, 2021.

Table of Contents

1. Introduction

Consider an operator providing Ethernet services such as EVPN. The Ethernet frames that a Provider Edge (PE) device receives from a Customer Edge (CE) device may have a larger size than the PE-PE path MTU (PMTU) in the provider network. This could be because

1. the provider network is built upon virtual connections (e.g., pseudowires) provided by another infrastructure provider, or

2. the customer network uses jumbo frames while the provider network does not, or

3. the provider-side overhead for transporting customer packets across the network pushes past the PMTU.

In any case, the provider cannot simply require its customers to change their MTU.

To get those large frames across the provider network, currently, the only workaround is to encapsulate the frames in IP (with or without GRE) and then fragment the IP packets. Even if MPLS is used for service delimiting, IP is used for transportation (MPLS over IP/GRE). This may not be desirable in certain deployment scenarios, where MPLS is the preferred transport or IP encapsulation overhead is deemed excessive.

IPv6 fragmentation and reassembly are based on the IPv6 Fragmentation header below [RFC8200]:

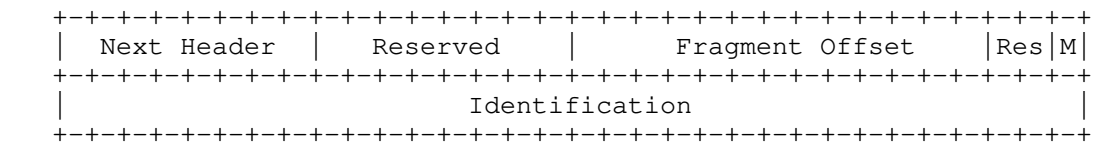


Figure 1: IPv6 Fragmentation Header

This document proposes adapting this header for use in non-IP contexts since the fragmentation/reassembly function is actually independent of IPv6 except for the following aspects:

- The fragment header is identified as such by the "previous" header.
- The "Next Header" value is from the "Internet Protocol Numbers" registry.
- The "Identification" value is unique in the (source, destination) context provided by the IPv6 header.

The "Identification" field, in conjunction with the IPv6 source and destination addresses identifies fragments of the original packet for the purpose of reassembly.

Therefore, the fragmentation/reassembly function can be applied at other layers as long as a) the fragment header is identified as such; and b) the context for packet identification is provided. Examples of such layers include MPLS, BIER, and Ethernet (if IEEE determines it is so desired).

For the same consideration, the IP Encapsulating Security Payload (ESP) [RFC4303] could also be applied at other layers if ESP is desired there. For example, if for whatever reason the Ethernet service provider wants to provide ESP between its PEs, it could do so without requiring IP encapsulation if ESP is applied at non-IP layers.

We refer to these as Generic Delivery Functions (GDFs), which could be achieved at a shim layer between a source and destination delivery points, for example:

- Source and destination IP/Ethernet nodes
- Ingress and egress nodes of MPLS Label Switch Paths (LSPs)

- o BIER Forwarding Ingress Routers (BFIRs) and BIER Forwarding Egress Routers (BFERs)

It is not the intended to apply the GDFs hop by hop between the source and destination delivery points.

The possibility of applying some other IP functions (e.g., Authentication Header [RFC4302]) is for further study.

2. Specifications

2.1. Generic Delivery Function Header

The following Generic Delivery Function Header (GDFH) is defined:

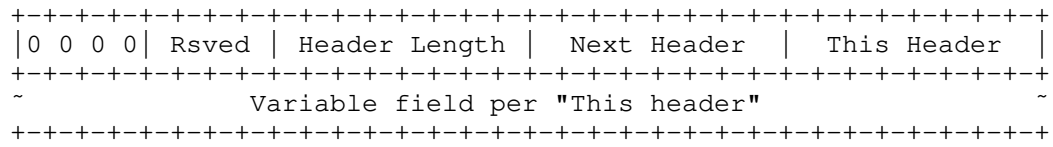


Figure 2: Generic Delivery Function Header

0000 nibble: Prevents the GDFH from being mistaken for an IP header by a router doing deep packet inspection for ECMP hashing purposes.

Header Length: The length of header in the number of 4-octet units.

Next Header: The type of next header. For functions that IETF is concerned with, the "Next Header" values are from the "Internet Protocol Numbers" registry (e.g., the next header could be a TCP or UDP or MPLS header). The next header could be another GDFH, and a value for GDFH will be assigned by IANA from that registry.

This Header: The type of this GDFH header. For example, TBD1 for generic fragmentation, TBD2 for generic ESP. The values are from a space independent of the "Next Header".

The outer header MUST identify that a GDFH follows. Encoding "This Header" in the GDFH allows a single outer header encoding to be used for different GDFHs.

If the outer header is BIER, a TBD value for the "proto" field in the BIER header identifies that a GDFH follows.

If the outer header is MPLS, the label preceding the GDFH indicates that a GDFH follows (see Section 2.5).

If the outer header is Ethernet (if IEEE would decide to provide the generic delivery functions on top of Ethernet directly), then a new Ethertype would be assigned by IEEE.

2.2. Generic Fragmentation Header

For generic fragmentation/reassembly functionality, the GDFH takes the following Generic Fragmentation Header (GFH) format:

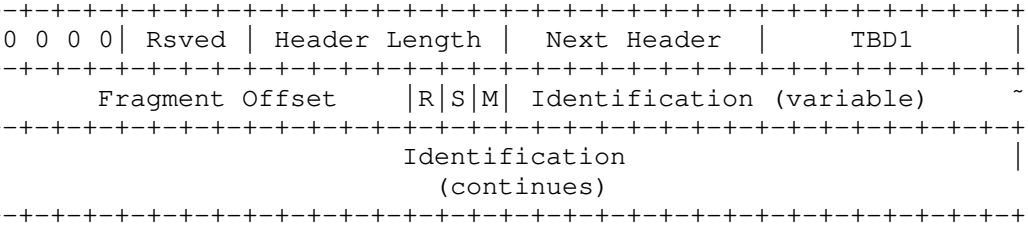


Figure 3: Generic Fragmentation Header

The "Fragment Offset" and "M" flag bit fields are as defined for the IPv6 Fragmentation Header in Section 4.5 [RFC8200].

R: The "R" flag bit is reserved. It MUST be set to 0 on transmit and ignored on receive.

Identification: at least 4-octet long. // would 2-octet be ok as a minimum?

S: If the "S" flag bit is clear, the context for the Identification field is provided by the outer header, and only the source-identifying information in the outer header is used.

If the "S" flag bit is set, the variable Identification field encodes both source-identifying information (e.g., the IP address of the node adding the GFH) and an identification number unique within that source.

When a GFH is used together with other GDFHs, the GFH SHOULD be the first GDFH.

If the outer header is BIER and the "S" flag bit is clear, the "BFIR-id" field in the BIER header provides the context for the "Identification" field.

If the outer header is MPLS, the "S" flag bit MAY be clear if the label preceding the GFH identifies the sending node in addition to indicating that a GFH follows (see Section 2.5).

2.3. Payload Type Header

While originally it is not the intention to provide a way to identify the payload type after an MPLS label stack, it has been pointed out that the GFH now provides the payload-identifying functionality as a by-product - even when fragmentation is not needed, a GFH can be inserted, with the Fragmentation Offset, the M-bit and Identification fields set to 0, and the Next Header set appropriately.

If the payload-identifying functionality is deemed as desired, a dedicated header type could be assigned for this purpose, with a smaller header compared to GFH.

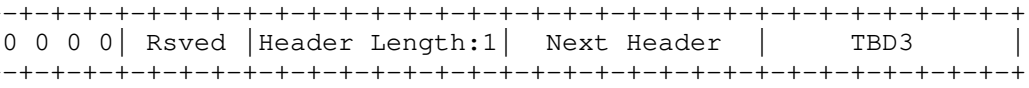


Figure 4: Generic Payload Type Header

2.4. Generic ESP/Authentication Header

To be specified in future revisions.

2.5. GDFH in an MPLS Data Plane

When GDFH is used in a network with MPLS data plane, the BoS label indicates that a GDFH immediately follows. The label can be either a special purpose label or a regular label signaled via BGP or IGP. The details will be provided in future revisions.

3. Security Considerations

To be provided.

4. Acknowledgements

The authors thank Stewart Bryant and Tony Przygienda for their valuable comments and suggestions.

5. References

5.1. Normative References

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

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

[RFC8362] Lindem, A., Roy, A., Goethals, D., Reddy Vallem, V., and F. Baker, "OSPFv3 Link State Advertisement (LSA) Extensibility", RFC 8362, DOI 10.17487/RFC8362, April 2018, <https://www.rfc-editor.org/info/rfc8362>.

5.2. Informative References

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net

Ron Bonica
Juniper Networks

Email: rbonica@juniper.net

Kireeti Kompella
Juniper Networks

Email: kireeti@juniper.net

Gregory Mirsky
ZTE

Email: gregory.mirsky@ztetx.com