PALS Working Group
Internet Draft
Intended status: Standard Track
Expires: April 2019

Italo Busi
Stewart Bryant
Andrew G. Malis
Jie Dong
Huawei

October 22, 2018

Pseudowire (PW) Control Word (CW) Stitching
draft-busi-pals-pw-cw-stitching-01.txt

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

This Internet-Draft will expire on April 22, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Abstract

This document defines the behavior of a new type of Multi-Segment Pseudowire (MS-PW) Switching PE (S-PE) which enhances the S-PE functions defined in [RFC 6073], with the capability to switch an Ethernet pseudowire (PW) segment that uses the PW Control Word (CW) [RFC 4385] with an Ethernet PW segment that does not use the CW.

This new type of S-PE can be deployed in the network one hop away, at the MPLS layer, from a Terminating PE (T-PE) which does not support CW for Ethernet PW encapsulation [RFC 4448]. In this way, all the Ethernet PW packets sent though the MPLS network will have the CW and be protected against incorrect equal-cost-multi-path (ECMP) behavior as described in [I-D ETH-CW].

Table of Contents

1. Introduction.....2
   1.1. Assumptions.....5
2. Terminology.....5
   2.1. Conventions Used in This Document.....6
3. Control Word Stitching procedures.....6
   3.1. CW Stitching Signaling.....7
4. VCCV Stitching Procedures.....8
   4.1. VCCV Stitching for CC Type 3.....9
   4.2. VCCV Stitching for CC Type 4.....10
   4.3. VCCV Stitching Signaling.....12
5. Other Deployment Scenarios.....13
6. Security Considerations.....15
7. IANA Considerations.....16
8. References.....16
   8.1. Normative References.....16
   8.2. Informative References.....16
9. Acknowledgments.....17

1. Introduction

In order to protect Ethernet pseudowire (PW) packets against incorrect equal-cost-multi-path (ECMP) behavior, which may cause out-of-order delivery of the payload Ethernet frames, the use of PW control word (CW) has been recommended in [I-D ETH-CW].

There are cases where service providers have existing deployments where the Provider Edge (PE) device is an old piece of equipment which does not support the CW for Ethernet PW encapsulation. In this case, the CW shall not be used as defined in [RFC 4448].

There are situations where replacing this PE with a new piece of equipment which supports CW for Ethernet PW is not acceptable because of economical or operational (e.g., service disruption time) reasons.

It may be beneficial to give operators an option to deploy (or re-use) another piece of equipment, located one hop away at the MPLS layer from this PE (typically physically co-located), which can add the CW to the Ethernet PW packets received from this PE, before sending them though an MPLS network.

This node should behave as a Switching PE (S-PE) as defined in [RFC 6073] and also be capable of switching an Ethernet pseudowire (PW) segment that uses the control word (CW) with an Ethernet PW segment that does not use the CW.

The reference network is shown in Figure 1 below.

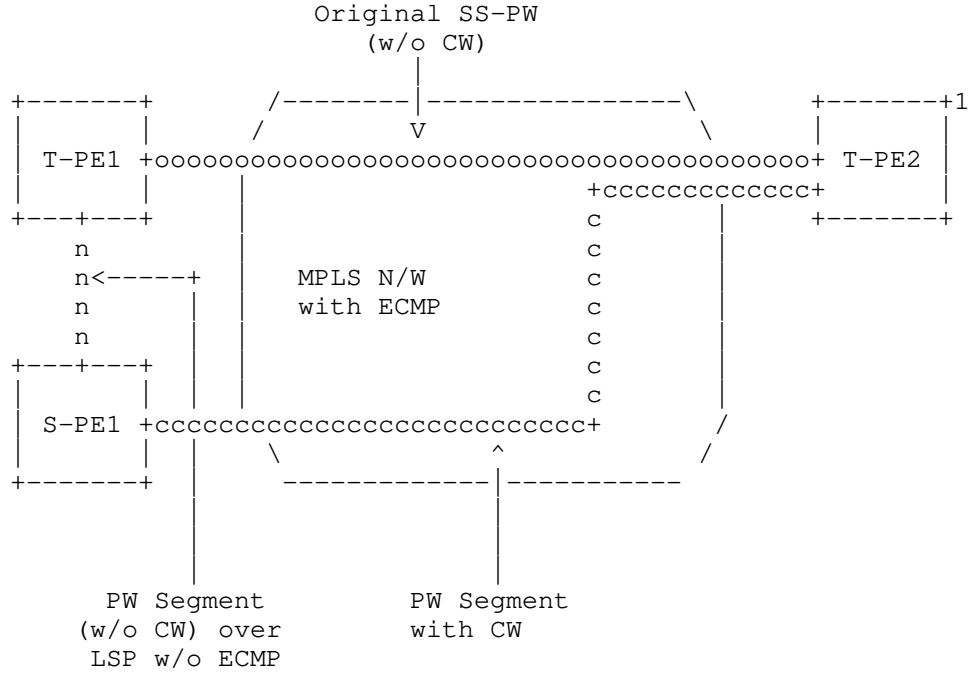


Figure 1 Reference Network

In Figure 1, T-PE1 is a device which is not capable of including a CW in Ethernet PW encapsulation, T-PE2 is a device which is capable to use the CW for Ethernet PW encapsulation, while S-PE1 is the new type of device defined in this document.

S-PE1 can be added to the network with minimum or no service disruption and PW redundancy [RFC 6718] or [RFC 7771] can be used to move the traffic from the old single-segment PW (SS-PW) without the CW to the new multi-segment PW (MS-PW) with the CW on the PW segment that passes through the MPLS network.

The deployment of the S-PE1, either as a new router or as an upgrade of an existing router, does not require any changes/upgrades to other nodes already installed within the network.

It is expected that in new deployments, all the Provider Edge (PE) devices are capable to insert the CW for Ethernet PW encapsulation and therefore the solution described in this document mainly applies to existing deployments where there are old pieces of equipment not being capable to support the CW for Ethernet PW encapsulation.

1.1. Assumptions

This document assumes that T-PE1 operates in the same way regardless of whether the PW is a SS-PW or a MS-PW, as defined in [RFC 6073]:

- T-PE1 signals SS-PW with T-PE2 using T-LDP, as defined in [RFC 4447]
- T-PE1 could be configured to signal a PW segment with S-PE1, as if it were T-PE2 using T-LDP, following the procedures defined in [RFC 6073].
- T-PE1 is capable to set the PW-TTL value (i.e., the TTL value of the PW LSE) for Ethernet PW packets to a proper value that allows the Ethernet frames to be forwarded on the AC on T-PE2 (e.g., PW-TTL>2 in Figure 1): this could be done either via administrative configuration or though T-LDP information.

It is also assumed that if T-PE1 supports Pseudowire Virtual Circuit Connectivity Verification (VCCV), it can support at least CC Type 3 or CC Type 4. The underlying rationale for this assumption is that use of CC Type 2 for MS-PW is not allowed in [RFC 6073].

If T-PE1 supports CC Type 3, it is assumed that it is capable to set the PW-TTL value for the VCCV packets to a proper value that allows the VCCV packets to be recognized by T-PE2 by PW-TTL expiry (e.g., PW-TTL=2): this could be done either via administrative configuration or though T-LDP information.

It is assumed that S-PE1 is manually configured to switch between the two PW segments, following the procedure described in [RFC 6073].

If T-PE2 supports VCCV, it is configured to always advertise support for CC type 1. This would allow simplifying the VCCV switching process since CC type 1 is always used on the PW segment with CW.

2. Terminology

This document re-uses the terminology defined in [RFC 6073] for single-segment pseudo-wire (SS-PW), multi-segment PW (MS-PW), terminating provider edge (T-PE) and switching provider edge (S-PE).

This document uses the acronym PW-TTL to indicate the TTL value in the PW label stack entry (LSE).

2.1. Conventions Used in This Document

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

3. Control Word Stitching procedures

The CW stitching procedure is performed by the S-PE1 on the Ethernet PW packets it is forwarding.

With a reference to Figure 1, it performs the following operations, in the direction from T-PE1 to T-PE2:

1. It pops the MPLS label stack entry (LSE) of the LSP from T-PE1 to S-PE1, if not PHP-ed by the penultimate LSR

2. It swaps the PW label (and decrements the PW-TTL)

3. It adds the CW immediately following the bottom of the label stack

4. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that step 3 is the only addition to the S-PE forwarding rules defined in [RFC 6073].

In this step, the S-PE inserts also the sequence number field in the control word, following the rules defined in [RFC4448].

In the opposite direction, S-PE1 performs the following operations:

1. It pops the MPLS label stack entry (LSE) for the LSP from T-PE2 to S-PE1, if not PHP-ed by the penultimate LSR

2. It swaps the PW label (and decrements the PW-TTL)

3. It removes the CW, which is located immediately following the bottom of the label stack

4. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that step 3 is the only addition to the S-PE forwarding rules defined in [RFC 6073].

In this step, the S-PE MAY also process the sequence number field in the control word, following the rules defined in [RFC4448].

3.1. CW Stitching Signaling

S-PE1 negotiates CW capabilities with T-PE1 and T-PE2 following almost the same procedures defined in [RFC 4447] and [RFC 6073].

The only exception to the procedures defined in [RFC 6073] is that S-PE1, when signaling one PW segment, will always behave as if the CW is supported on the other PW segment.

This allows S-PE1 to negotiate different CW capabilities on different PW segments as well as to enable CW toward any T-PE that support CW insertion.

If the same CW capabilities are negotiated on both PW segments, then S-PE1 will behave as specified in [RFC 6073]. CW stitching, as defined in this document, is enabled if and only if different CW capabilities are negotiated on the two PW segments.

In case the S-PE considers the sequence number field in the control word, it SHALL follow the rules described in section 6.4 of [RFC4447].

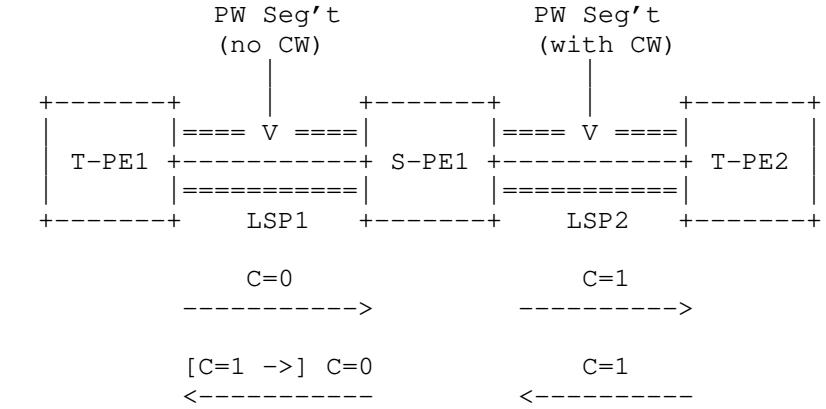


Figure 2 CW Stitching Signaling

Figure 2 shows an example of how CW capabilities are negotiated in the reference network scenario of Figure 1.

T-PE1 will send a T-LDP Label Mapping message with c=0 and T-PE2 will send a T-LDP Label Mapping message with C=1, following the procedures defined in section 6.2 of [RFC 4447] and amended by [I-D ETH-CW].

After S-PE1 receives the T-LDP Label Mapping message (with c=1) from T-PE2, it can send a T-LDP Label Mapping message back to T-PE2 (with c=1), following the procedures defined in section 6.2 of [RFC 4447], and a T-LDP Label Mapping messages to T-PE1 (with c=1), following the procedures of [RFC 6073].

After S-PE1 receives the T-LDP Label Mapping message (with c=0) from T-PE1, it can send a T-LDP Label Mapping message to T-PE2 (with c=1), as if it has received c=1 from T-PE1. It can also send a T-LDP Label Mapping message back to T-PE1 with c=0, following the procedures defined in section 6.2 of [RFC 4447].

If S-PE1 receives the T-LDP Label Mapping message (with c=0) from T-PE1 after having sent a T-LDP Label Mapping message with c=1 to T-PE1, a Label Withdraw message needs to be sent to T-PE1 before sending another Label Mapping message with c=1, as specified in section 6.2 of [RFC 4447].

When the MS-PW is completely setup:

- T-PE1 is configured not to insert CW
- T-PE2 is configures to insert CW
- S-PE1 is configured to stitch the CW between the two PW segments

4. VCCV Stitching Procedures

When CW stitching is enabled, VCCV packets sent on the two PW segments would have different formats. In order to enable end-to-end OAM, S-PE1 needs to be capable to perform VCCV stitching.

Since support of CC Type 1 is REQUIRED by [RFC 5085] for PWs that support the CW, within this document it is RECOMMENDED that its use is always enabled at the T-PEs supporting the CW (e.g., T-PE2) such that, following the rules defined in [RFC 5085], when VCCV is in use, CC Type 1 is always used on the PW segment that support the CW.

Since [RFC 5085] does not define any mandatory CC Types for the PWs that do not support CW, different VCCV stitching procedures need to be defined depending on the CC Type supported by the T-PE not supporting the CW (e.g., T-PE1).

The VCCV stitching procedure is performed by S-PE1 on the VCCV packets it is forwarding.

In the traffic direction from T-PE2 and T-PE1 CC Type 1 is used: S-PE1 can distinguish VCCV and Ethernet PW packets by looking at the first nibble immediately following the bottom of the label stack which identifies either an ACH or a CW:

- Ethernet PW packets are received with the CW: these packets need to be forwarded following the rules defined in section 3.
- VCCV packets targeted at S-PE1 are received with the ACH and the PW-TTL=1: these packets should be processed by S-PE1 and not forwarded.
- Other VCCV packets are received with the ACH and with a PW-TTL value greater than 1: these packets need to be forwarded following the rules defined in the following sections.

In the traffic direction from T-PE1 and T-PE2, the rules used to distinguish VCCV packets from Ethernet PW packets depends from the CC Type used on the PW segment without the CW.

4.1. VCCV Stitching for CC Type 3

In case CC Type 3 is used on the PW segment not using the CW, VCCV stitching needs to translate between CC Type 3 (without the CW) and CC Type 1. It is worth noting that when CC Type 3 is used on PW segments not using the CW, only IP-based CV types can be supported.

In the traffic direction from T-PE1 and T-PE2, S-PE1 can distinguish VCCV and Ethernet PW packets by looking at the PW-TTL value:

- Ethernet PW packets are received with a PW-TTL value exceeding the PW-TTL distance from S-PE1 to T-PE2 (e.g., TTL>2): these packets need to be forwarded following the rules defined in section 3.
- VCCV packets targeted at S-PE1 are received with PW-TTL=1: these packets should be processed by S-PE1 and not forwarded.
- Other VCCV packets are received with a PW-TTL value greater than 1 and not exceeding the PW-TTL distance to T-PE2 (e.g., TTL=2): these packets need to be forwarded following the rules defined in this section.

With a reference to Figure 1, S-PE1 performs the following operations, in the direction from T-PE1 to T-PE2:

1. It pops the MPLS label stack entry (LSE) of the LSP from T-PE1 to S-PE1, if not PHP-ed by the penultimate LSR

2. It swaps the PW label (and decrements the PW-TTL)

3. It adds the ACH immediately following the bottom of the label stack (setting the ACH Channel Type based on the IP version field of the encapsulated IP packet)

4. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that step 3 is the only addition to the S-PE forwarding rules defined in [RFC 6073]: it is also the only step where the forwarding rules of VCCV packets are different from the forwarding rules defined for Ethernet PW packets in section 3.

S-PE1 can understand the IP version field of the encapsulated IP packet by looking at the first nibble immediately following the bottom of the label stack of the received packet.

In the opposite direction, S-PE1 performs the following operations:

1. It pops the MPLS label stack entry (LSE) for the LSP from T-PE2 to S-PE1, if not PHP-ed by the penultimate LSR

2. It swaps the PW label (and decrements the PW-TTL)

3. It removes the ACH, which is located immediately following the bottom of the label stack

4. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that step 3 is the only addition to the S-PE forwarding rules defined in [RFC 6073]: it is also the only step where the forwarding rules of VCCV packets are different from the forwarding rules defined for Ethernet PW packets in section 3.

4.2. VCCV Stitching for CC Type 4

In case CC Type 4 is used on the PW segment not using the CW, VCCV stitching needs to translate between CC Type 4 and CC Type 1. It is

worth noting that in this case both IP-based and ACH-based CV types can be supported.

In the traffic direction from T-PE1 and T-PE2, S-PE1 can distinguish VCCV and Ethernet PW packets by looking at GAL LSE right after the PW LSE:

- Ethernet PW packets are received without a GAL LSE: these packets need to be forwarded following the rules defined in section 3.
- VCCV packets targeted at S-PE1 are received with the GAL LSE and with the PW-TTL=1: these packets should be processed by S-PE1 and not forwarded.
- Other VCCV packets are received with the GAL LSE and with a PW-TTL value greater than 1: these packets need to be forwarded following the rules defined in this section.

With a reference to Figure 1, S-PE1 performs the following operations, in the direction from T-PE1 to T-PE2:

1. It pops the MPLS label stack entry (LSE) of the LSP from T-PE1 to S-PE1, if not PHP-ed by the penultimate LSR
2. It swaps the PW label (and decrements the PW-TTL)
3. It removes the GAL LSE at the bottom of the label stack
4. It sets the S-bit of the PW LSE since the PW LSE becomes the new bottom of the label stack
5. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that steps 3 and 4 are the only additions to the S-PE forwarding rules defined in [RFC 6073]: they are also the only steps where the forwarding rules of VCCV packets are different from the forwarding rules defined for Ethernet PW packets in section 3.

In the opposite direction, S-PE1 performs the following operations:

1. It pops the MPLS label stack entry (LSE) for the LSP from T-PE2 to S-PE1, if not PHP-ed by the penultimate LSR
2. It swaps the PW label (and decrements the PW-TTL)
3. It inserts the GAL LSE at the bottom of the label stack

4. It clears the S-bit of the PW LSE since the PW LSE is no longer at the bottom of the label stack

5. It pushes the MPLS LSE for the LSP to T-PE2, unless this LSP is a single-hop PHP-ed LSP.

It is worth noting that steps 3 and 4 are the only additions to the S-PE forwarding rules defined in [RFC 6073]: they are also the only steps where the forwarding rules of VCCV packets are different from the forwarding rules defined for Ethernet PW packets in section 3.

4.3. VCCV Stitching Signaling

S-PE1 negotiates VCCV capabilities with T-PE1 and T-PE2 following almost the same procedures defined in [RFC 5085] and [RFC 6073].

If the same CW capabilities are negotiated on both PW segments, then S-PE1 will behave as specified in [RFC 6073]. VCCV stitching, as defined in this document, is enabled if and only if different CW capabilities are negotiated on the two PW segments.

If S-PE1 supports VCCV stitching for CC Type 3, and it knows the PW-TTL distance to both T-PE1 and T-PE2:

- If T-PE1 advertises support for CC Type 3, S-PE1 advertises support for CC Type 1 to T-PE2

- If T-PE2 advertises support for CC Type 1, S-PE1 behaves toward T-PE1 if it supports CC Type 3 and T-PE2 has advertised support for CC Type 3, following the procedure defined in [RFC 6073]

If S-PE1 supports VCCV stitching for CC Type 4:

- If T-PE1 advertises support for CC Type 4, S-PE1 advertises support for CC Type 1 to T-PE2

- If T-PE2 advertises support for CC Type 1, S-PE1 behaves toward T-PE1 as if it supports CC Type 4 and T-PE2 has advertised support for CC Type 4, following the procedure defined in [RFC 6073]

CV types are advertised based on S-PE1 capabilities as per [RFC 6073] with the following additional rule:

- S-PE1 can advertise support for ACH-based CV types if and only if it supports VCCV stitching for CC Type 4

This rule ensures that only IP-based CV types are negotiated between T-PE1, T-PE2 and S-PE1 when VCCV stitching for CC Type 3 is used.

If T-PE1 supports CC Type 4 and S-PE1 supports VCCV stitching for CC Type 4, then VCCV stitching for CC Type 4 is used and both IP-based and ACH-based CV capabilities can be negotiated depending on T-PE1, T-PE2 and S-PE1 CV capabilities.

If T-PE1 does not support CC Type 4, it will advertise support only for IP-based CV types and therefore only IP-based CV capabilities can be negotiated depending on T-PE1, T-PE2 and S-PE1 CV capabilities.

If S-PE1 does not support VCCV stitching for CC Type 4, it will advertise support only for IP-based CV types and therefore only IP-based CV capabilities can be negotiated depending on T-PE1, T-PE2 and S-PE1 CV capabilities.

5. Other Deployment Scenarios

The solution described in this document is quite generic and can be used in different deployment scenarios, in addition to the reference network outline in Figure 1, without requiring any change to the behavior of the S-PE, as defined in this document.

A possible deployment scenario is shows in Figure 3 where both T-PEs are not capable to insert the CW:

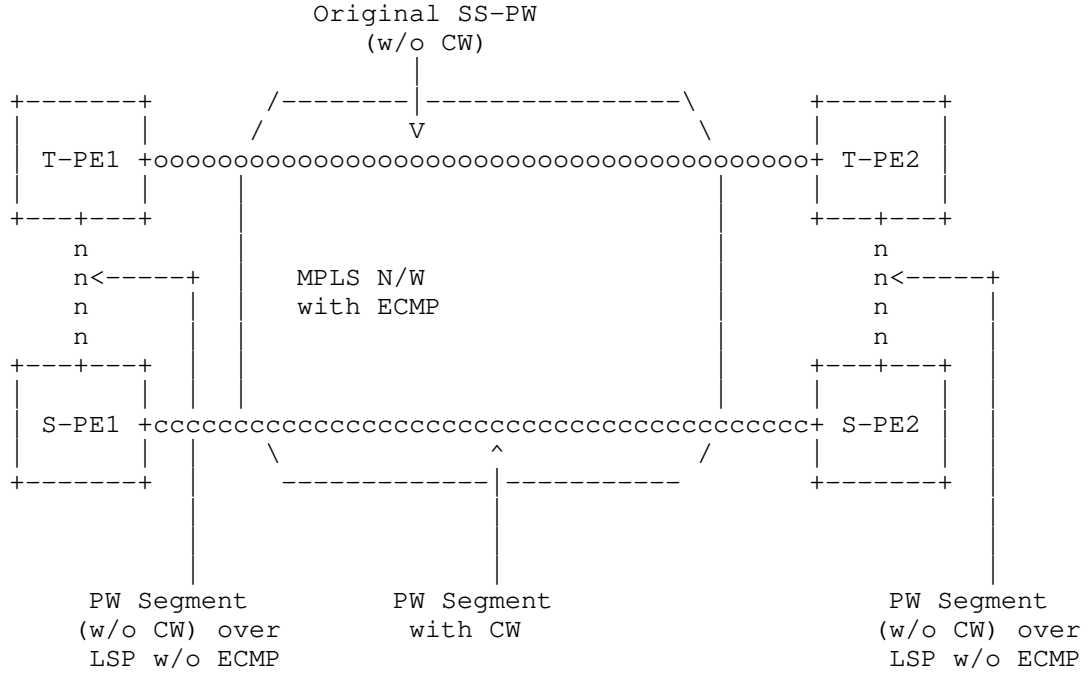


Figure 3 Reference network with two T-PEs not capable to insert CW

In this scenario, two S-PEs needs to be deployed: S-PE1 in front of T-PE1 and S-PE2 in front of T-PE2.

S-PE1 and S-PE2 operate as defined in this document: these operations are the same even if one or both the PW segments switched by one S-PE are terminated at a T-PE or at another S-PE.

An even more generic deployment scenario is shows in Figure 3:

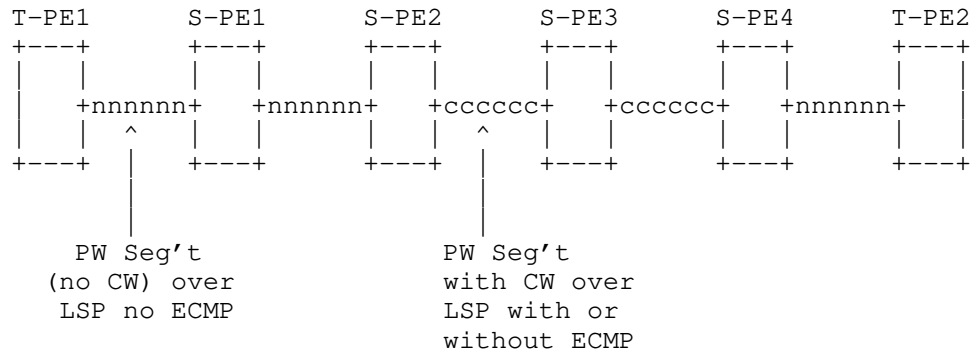


Figure 4 More generic Reference network

In this case a MS-PW can be setup with some PW segments using the CW and other not using the CW.

S-PE1 and S-PE3 operates as defined in [RFC 6073] while S-PE2 and S-PE4 operate as defined in this document: these operations are the same even if one or both the PW segments switched by one S-PE are terminated at a T-PE or at another S-PE operating as defined in [RFC 6073] or at another S-PE operating as defined in this document.

The operations are also the same if the PW segment not using the CW is setup over a link or over an MPLS network.

In order to achieve the desired behavior, i.e., to avoid the issues described in [I-D ETH-CW], care must be taken by the operator to make sure that no ECMP is used within the MPLS network carrying the PW segments without the CW.

The operations described in this document work also if static configuration is used instead of T-LDP to setup some or all the PW segments.

The operations described in this document work also if dynamic MS-PW signaling procedures, as defined in [RFC7267], are used instead of static configuration of the S-PEs.

6. Security Considerations

The method described in this document adds no security issues beyond those encountered in a network running multi-segment Ethernet pseudowires with the Control Word over MPLS, as previously discussed in [RFC4385], [RFC4448] and [RFC7267]. Such networks are normally private, well managed, highly controlled environments.

7. IANA Considerations

This document makes no IANA requests.

8. References

8.1. Normative References

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

[RFC4385] Bryant, S. et al., "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[RFC4447] Martini, L. et al., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[RFC4448] Martini, L. et al., "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, April 2006.

[RFC5085] Nadeu, T., Pignataro, C., " Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.

[RFC6073] Martini, L. et al., "Segmented Pseudowire", RFC 6073, January 2011.

[RFC7267] Martini, L. et al., "Dynamic Placement of Multi-Segment Pseudowires", RFC 7267, June 2014.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[I-D ETH-CW] Bryant, S. et al., "Use of Ethernet Control Word RECOMMENDED", draft-ietf-pals-ethernet-cw, work in progress.

8.2. Informative References

[RFC6718] Muley, P. et al., "Pseudowire Redundancy", RFC 6718, August 2012.

[RFC7771] Malis, A. et al., "Switching Provider Edge (S-PE) Protection for MPLS and MPLS Transport Switching Provider Edge (S-PE) Protection for MPLS and MPLS Transport", RFC 7771, January 2016.

9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Italo Busi
Huawei

Email: italo.busi@huawei.com

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Andrew G. Malis
Huawei

Email: agmalis@gmail.com

Jie Dong
Huawei

Email: jie.dong@huawei.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 7, 2021

C. Ramachandran
V. Beeram
Juniper Networks
H. Sitaraman
Individual

January 3, 2021

Node Protection for RSVP-TE tunnels on a shared MPLS forwarding plane
draft-chandra-mpls-rsvp-shared-labels-np-05

Abstract

Segment Routed RSVP-TE tunnels provide the ability to use a shared MPLS forwarding plane at every hop of the Label Switched Path (LSP). The shared forwarding plane is realized with the use of 'Traffic Engineering (TE) link labels' that get shared by LSPs traversing these TE links. This paradigm helps significantly reduce the forwarding plane state required to support a large number of LSPs on a Label Switching Router (LSR). These tunnels require the ingress Label Edge Router (LER) to impose a stack of labels. If the ingress LER cannot impose the full label stack, it can use the assistance of one or more delegation hops along the path of the LSP to impose parts of the label stack.

The procedures for a Point of Local Repair (PLR) to provide local protection against link failures using facility backup for Segment Routed RSVP-TE tunnels are well defined and do not require specific protocol extensions. This document defines the procedures for a PLR to provide local protection against transit node failures using facility backup for these tunnels. The procedures defined in this document include protection against delegation hop failures.

Requirements Language

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

Status of This Memo

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

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on July 7, 2021.

Copyright Notice

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

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

Table of Contents

1. Introduction 3
2. Terminology 3
3. Node Protection Specific Procedures 4
3.1. Applicability of this Document 4
3.2. PLR Procedures for Protecting Next-Hop Non-Delegation LSR 4
3.3. PLR Procedures for Protecting Next-Hop Delegation LSR . . 5
3.3.1. Label Allocation and Stacking 7
3.4. Backwards Compatibility 7
3.4.1. LSR does not Support Node Protection for Shared Labels 7
3.4.2. Protected Hop does not Support Shared Labels 9
3.4.3. PLR does not Support Shared Labels 9
4. Protocol Extensions 9
4.1. DHLD Encoding in ETLD Attributes TLV 9
5. Acknowledgements 10
6. IANA Considerations 10
7. Security Considerations 10
8. References 10
8.1. Normative References 10
8.2. Informative References 11
Authors' Addresses 11

1. Introduction

With the advent of Traffic Engineering (TE) link labels and Segment Routed RSVP-TE Tunnels [RFC8577], a shared MPLS forwarding plane can be realized by allowing the TE link label to be shared by MPLS RSVP-TE Label Switched Paths (LSPs) traversing the link. The shared forwarding plane behavior helps reduce the amount of forwarding plane state required to support a large number of LSPs on a Label Switching Router (LSR).

Segment Routed RSVP-TE tunnels request the use of a shared forwarding plane at every hop of the LSP. The TE link label used at each hop is recorded in the Record Route object (RRO) of the Resv message. The ingress Label Edge Router (LER) uses this recorded information to construct a stack of labels that can be imposed on the packets steered on to the tunnel. In the scenario where the ingress LER cannot impose the full label stack, it can use the assistance of one or more delegation hops along the path of the LSP to impose parts of the label stack.

Facility backup is a local repair method [RFC4090] in which a bypass tunnel is used to provide protection against link or node failures for MPLS RSVP-TE LSPs at the Point of Local Repair (PLR). The facility backup procedures that provide protection against link failures for Segment Routed RSVP-TE LSPs are defined in [RFC8577]. This document defines the facility backup procedures that provide protection against node failures for these LSPs. These procedures include protection against delegation hop failures. The document also discusses the procedures for handling backwards compatibility scenarios where a node along the path of the LSP does not support the procedures defined in this document.

The procedures discussed in this document do not cover protection against ingress/egress node failures. They also do not apply to Point to Multipoint (P2MP) RSVP-TE Tunnels.

2. Terminology

The reader is expected to be familiar with the terminology specified in [RFC3209], [RFC4090] and [RFC8577]. Unless otherwise stated, the term LSPs in this document refer to Segment Routed RSVP-TE LSPs. The following additional terms are used in this document:

Primary forwarding action: The outbound label forwarding action performed at a PLR for a protected LSP before the occurrence of local failure.

Backup forwarding action: The outbound label forwarding action performed at a PLR for a protected LSP after the occurrence of local failure.

3. Node Protection Specific Procedures

A set of Segment Routed RSVP-TE LSPs can share a TE link label on an LSR only if all the LSPs in the set share the same outbound label forwarding action. For protected LSPs, having the same outbound label forwarding action means having the same primary forwarding action and the same backup forwarding action. In the case of LSPs that do not request local protection or LSPs that request only link protection, they can use the same outbound label forwarding action if they reach a common next-hop LSR via a common outgoing TE link. However, in the case of LSPs that request node protection, they can use the same outbound label forwarding action only if they reach a common next-next-hop LSR via a common outgoing TE link and a common next-hop LSR.

3.1. Applicability of this Document

The label allocation and signaling procedures defined in [RFC8577] can sufficiently cater to the following scenarios on an LSR:

(a) Offer no protection to LSPs that do not request local protection

(b) Offer no protection or link protection to LSPs that request link protection

(c) Offer no protection or link protection to LSPs that request node protection

The label allocation and signaling procedures defined in this document are meant to enable LSRs to offer node protection to LSPs that request node protection.

3.2. PLR Procedures for Protecting Next-Hop Non-Delegation LSR

If the protected next-hop LSR signals a TE link label for the LSP but does not set the Delegation Label flag in the RRO Label Subobject carried in Resv message, then the PLR SHOULD allocate multiple shared labels for the same TE link such that a unique label is allocated for every unique next-next-hop LSR that is reachable via the protected next-hop LSR.

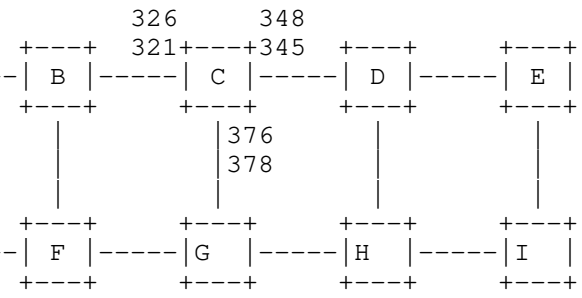


Figure 1: Per-nhop-nnhop label allocation

In the example shown in Figure 1, LSR C has allocated the following TE link labels:

321 for the TE link C-B to reach the next-next-hop LSR A
326 for the TE link C-B to reach the next-next-hop LSR F
345 for the TE link C-D to reach the next-next-hop LSR E
348 for the TE link C-D to reach the next-next-hop LSR H
376 for the TE link C-G to reach the next-next-hop LSR F
378 for the TE link C-G to reach the next-next-hop LSR H

If a LSP requesting node protection transits PLR C and if the protected next-hop LSR after C along the LSP path is not a delegation hop, then LSR C signals the respective TE link label depending on the next-next-hop LSR on the LSP path.

LSP path: A -> B -> C -> D -> E : Label = 345
LSP path: A -> B -> C -> D -> H : Label = 348
LSP path: A -> B -> C -> G -> H : Label = 378

In all LSP paths above, at PLR C, the protected next-hop LSRs D and G along the LSP paths signal TE link labels but are not delegation hops.

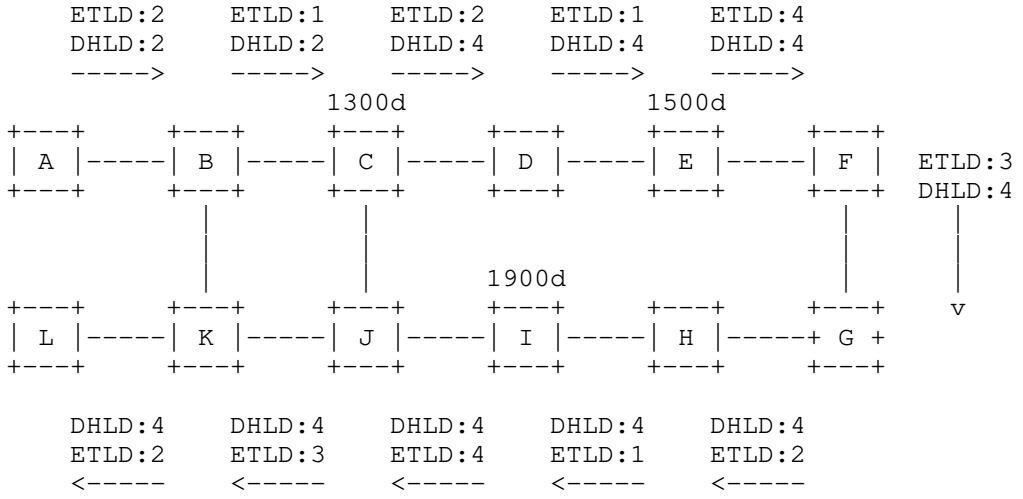
If the primary TE link is operational, LSR C will pop the TE link label and forward the packet to the corresponding next-hop LSR over that TE link. During local repair, LSR C will pop the TE link label and also the label beneath the top label, and forward the packet over the node protecting bypass tunnel to the appropriate next-next-hop LSR, which is the Merge Point (MP).

3.3. PLR Procedures for Protecting Next-Hop Delegation LSR

The outgoing backup label forwarding action corresponding to a label shared by LSPs requesting node protection MUST bypass the protected next-hop LSR. The PLR MUST push the label stack on behalf of the

next-hop delegation LSR. Hence, the number of labels that a delegation hop chooses to push also depends on the number of labels that the upstream hop (acting as PLR) along the primary LSP can push. This section extends the Effective Transport Label-Stack Depth (ETLD) signaling procedure specified in [RFC8577] for LSPs requesting node protection.

Considering Figure 2, assume LER A and LSR B can push a maximum of 3 labels on an MPLS packet while the remaining nodes can push a maximum of 5 labels. LER A originates a Path message with an ETLD of 2 after reserving space for the bypass tunnel label that should be pushed for backup forwarding action. In addition to setting the ETLD, LER A also sets the Delegation Helper Label Depth (DHLD) to 2 in the Path message. The DHLD is computed as the maximum number of labels that the node can push after reserving space for the NNHOP bypass tunnel label that should be pushed for backup forwarding action. The ETLD procedures dictate that each LSR add its own ETLD value before sending the Path message downstream. LSRs C, E and I are automatically selected as delegation hops by the time the Path message reaches the egress LER L. LSR C uses the DHLD signaled by the upstream LSR B as input when calculating the outgoing ETLD in the Path message.



Notation : <Label>d - delegation label

Figure 2: ETLD and DHLD signaling for node protection

As shown in Figure 2, delegation hop LSR C does not set outgoing ETLD to 4 that it would have normally set given that LSR C can push a maximum of 5 labels on an outgoing packet. Instead, LSR C sets the outgoing ETLD to the minimum of the ETLD that it computes and the DHLD value of its previous hop i.e. minimum(computed ETLD = 4, previous hop DHLD = 2).

The extension for signaling the DHLD in the Path message is defined in Section 4.1.

3.3.1. Label Allocation and Stacking

An LSR that decides to become a delegation hop for one or more LSPs requesting node protection MUST allocate a delegation label separate from delegation label assigned for LSPs that are offered no protection or link protection - even though the delegation segments share the same hops. In the example shown in Figure 2, the delegation hops LSRs C, E and I will set the Delegation Label flag in the Label sub-object that they add to the Resv message.

A PLR node that offers node protection to a delegation hop SHOULD be capable of helping the downstream delegation when the primary TE link to the delegation hop goes down. In the example shown in Figure 1, the LSRs B, D and H act as helpers for their respective downstream delegation hops. The PLR nodes that are delegation helpers along the path of LSPs requesting node protection SHOULD allocate a unique label for every delegation label signaled by the protected delegation node.

Before primary TE link failure, the PLR playing the role of a delegation helper pops the incoming label and forwards the packet on the primary TE link. During local repair, the delegation helper PLR pops the incoming label and also the label beneath it and pushes the label stack on behalf of the next hop delegation LSR and forwards the packet over the bypass tunnel.

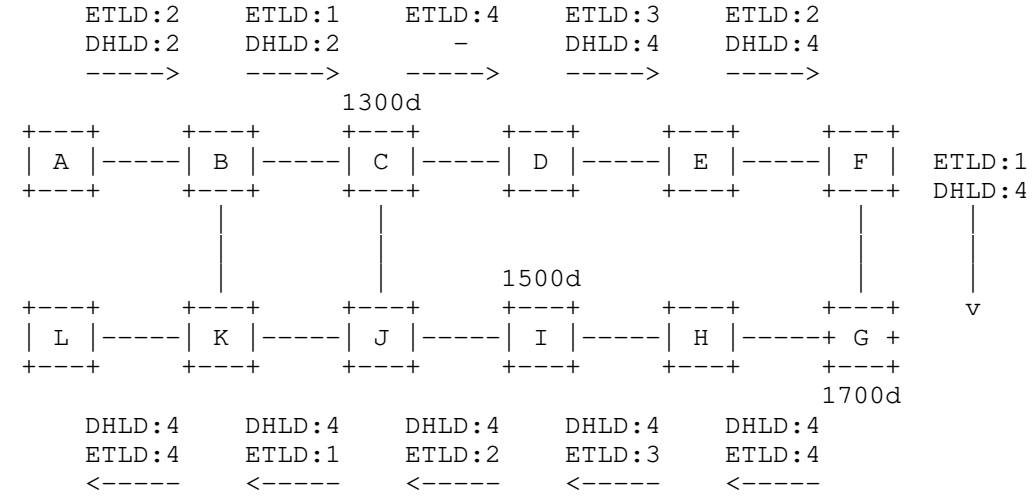
Any LSR that creates label stack upstream of the delegation helper MUST include the label signaled by the delegation helper onto the outgoing label stack just as it uses the TE link label to construct outgoing label stack.

3.4. Backwards Compatibility

3.4.1. LSR does not Support Node Protection for Shared Labels

As defined in Section 3.1, any LSR along the path of an LSP requesting node protection may choose to instead offer no protection or link protection. Hence, it must be possible to build an LSP where

some LSRs along the path support the node protection extensions defined in this document whereas the rest of the LSRs support only [RFC8577]. Any PLR along the LSP path that does not support the procedures defined in this document MUST provide either no protection or link protection for the LSPs requesting node protection.



Notation : <Label>d - delegation label

Figure 3: Delegation node does not support signaling DHLD

In Figure 3, assume LER A and LSR B can push a maximum of 3 labels to the MPLS packet while the remaining nodes can push a maximum of 5 labels. Also assume that LSR C supports the extensions defined in [RFC8577] but does not support the extensions defined in this document. Based on ETLD signaling procedure, LSR C will become a delegation hop. However, as LSR C cannot understand the DHLD signaled by the previous hop LSR B, LSR C will set outgoing ETLD to 4. If LSR C had supported the DHLD signaling, it would have set outgoing ETLD to 2 (see Section 3.3). When PLR B receives shared label from the protected next-hop LSR C in the Resv message, it must determine the number of labels it has to push in order to offer node protection from the RRO sub-object carried in Resv. As the label stack depth of the delegation hop C is greater than the number of labels LSR C can push, it must either not provide local protection or provide only link protection for the LSP.

3.4.2. Protected Hop does not Support Shared Labels

If the ingress LER has requested label stacking to reach delegation hop for the LSP requesting node protection, and if the next-hop LSR allocates a regular label for the LSP, then the LSR MUST also allocate a regular label for the LSP.

If the ingress LER has requested label stacking to reach the egress LER for the LSP requesting node protection, and if the next-hop LSR has allocated a regular label for the LSP, then the PLR MUST become a delegation hop and set the RRO Label Subobject delegation label flag in the RRO carried in Resv message. The PLR MUST set ETLD to 1 in its outgoing Path message.

3.4.3. PLR does not Support Shared Labels

If an LSR determines that its immediate upstream LSR (PLR) has not included an ETLD in the incoming Path message, then the LSR MUST become a delegation hop and set the ETLD to 1 in the outgoing Path message. The outgoing ETLD is set to 1 because the upstream LSR does not support shared labels and cannot push the label stack on behalf of this LSR.

4. Protocol Extensions

This section discusses the protocol extension required to support the procedures in Section 3.3

4.1. DHLD Encoding in ETLD Attributes TLV

Delegation Helper Label Depth (DHLD) is defined as the number of labels that an LSR has the capability to push while performing local repair protecting the next-hop delegation LSR. This document updates the ETLD Attributes TLV defined in [RFC8577]. The encoding of DHLD in the ETLD Attributes TLV is shown in Figure 4

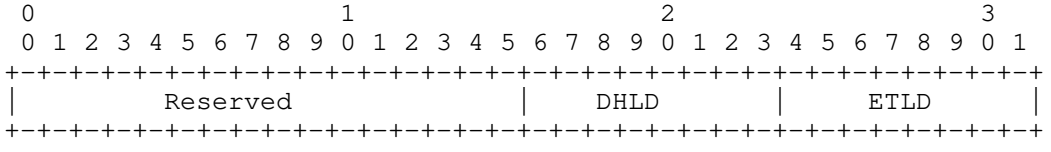


Figure 4: The ETLD Attributes TLV

The presence of ETLD Attributes TLV in the HOP_ATTRIBUTES sub-object [RFC7570] of the RRO object carried in Path message indicates that

the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID sub-object supports automatic delegation [RFC8577].

An implementation that supports this document MUST set the 8 bits from bit number 16 to bit number 23 with its DHLD value as indicated in Figure 4 when signaling Path message for an LSP for which node protection has been requested.

When processing the ETLD Attributes TLV of the previous hop LSR in the received Path message, the LSR checks whether it has to be the delegation hop based on the ETLD algorithm defined in [RFC8577].

If the LSR does not become a delegation hop along the LSP path, then no further action is required based on the DHLD value set by the previous hop.

If the LSR does become a delegation hop along the LSP path, then it MUST decode the 8 bit unsigned value from bit number 16 to bit number 23 as indicated in Figure 4. If the 8 bit value is zero, then the LSR MUST infer that the previous hop has not included DHLD in the ETLD Attributes TLV. If the 8 bit value is non-zero, then the LSR MUST consider that value as the DHLD value signaled by the previous hop LSR and use that DHLD value for computing its own outgoing ETLD.

5. Acknowledgements

The authors would like to thank Raveendra Torvi for his input from discussions.

6. IANA Considerations

This document includes no requests to IANA.

7. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

8. References

8.1. Normative References

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[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>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <https://www.rfc-editor.org/info/rfc7570>.

[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>.

[RFC8577] Sitaraman, H., Beeram, V., Parikh, T., and T. Saad, "Signaling RSVP-TE Tunnels on a Shared MPLS Forwarding Plane", RFC 8577, DOI 10.17487/RFC8577, April 2019, <https://www.rfc-editor.org/info/rfc8577>.

8.2. Informative References

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

Authors' Addresses

Chandra Ramachandran
Juniper Networks

Email: csekar@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Harish Sitaraman
Individual

Email: harish.ietf@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: February 1, 2020

Yimin Shen
Minto Jeyananth
Juniper Networks
Bruno Decraene
Orange
Hannes Gredler
RtBrick Inc
Carsten Michel
Deutsche Telekom
Huaimo Chen
Huawei Technologies Co., Ltd.
July 31, 2019

MPLS Egress Protection Framework
draft-ietf-mpls-egress-protection-framework-07

Abstract

This document specifies a fast reroute framework for protecting IP/MPLS services and MPLS transport tunnels against egress node and egress link failures. For each type of egress failure, it defines the roles of point of local repair (PLR), protector, and backup egress router, and the procedures of establishing a bypass tunnel from a PLR to a protector. It describes the behaviors of these routers in handling an egress failure, including local repair on the PLR, and context-based forwarding on the protector. The framework can be used to develop egress protection mechanisms to reduce traffic loss before global repair reacts to an egress failure and control plane protocols converge on the topology changes due to the egress failure.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 1, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction 3
2. Specification of Requirements 5
3. Terminology 5
4. Requirements 7
5. Egress node protection 8
5.1. Reference topology 8
5.2. Egress node failure and detection 8
5.3. Protector and PLR 9
5.4. Protected egress 10
5.5. Egress-protected tunnel and service 11
5.6. Egress-protection bypass tunnel 11
5.7. Context ID, context label, and context-based forwarding . 12
5.8. Advertisement and path resolution for context ID 13
5.9. Egress-protection bypass tunnel establishment 15
5.10. Local repair on PLR 16
5.11. Service label distribution from egress router to protector 16
5.12. Centralized protector mode 17
6. Egress link protection 19
7. Global repair 22
8. Operational Considerations 22
9. General context-based forwarding 23
10. Example: Layer-3 VPN egress protection 23
10.1. Egress node protection 25
10.2. Egress link protection 25
10.3. Global repair 26
10.4. Other modes of VPN label allocation 26
11. IANA Considerations 26
12. Security Considerations 26
13. Acknowledgements 27
14. References 28

14.1. Normative References 28
14.2. Informative References 28
Authors' Addresses 29

1. Introduction

In MPLS networks, label switched paths (LSPs) are widely used as transport tunnels to carry IP and MPLS services across MPLS domains. Examples of MPLS services are layer-2 VPNs, layer-3 VPNs, hierarchical LSPs, and others. In general, a tunnel may carry multiple services of one or multiple types, if the tunnel satisfies both individual and aggregate requirements (e.g., CoS, QoS) of these services. The egress router of the tunnel hosts the service instances of the services. An MPLS service instance forwards service packets via an egress link to the service destination, based on a service label. An IP service instance does the same based on a service IP address. The egress link is often called a PE-CE (provider edge - customer edge) link or attachment circuit (AC).

Today, local-repair-based fast reroute mechanisms ([RFC4090], [RFC5286], [RFC7490], and [RFC7812]) have been widely deployed to protect MPLS tunnels against transit link/node failures, with traffic restoration time in the order of tens of milliseconds. Local repair refers to the scenario where the router upstream to an anticipated failure, a.k.a. PLR (point of local repair), pre-establishes a bypass tunnel to the router downstream of the failure, a.k.a. MP (merge point), pre-installs the forwarding state of the bypass tunnel in the data plane, and uses a rapid mechanism (e.g., link layer OAM, BFD, and others) to locally detect the failure in the data plane. When the failure occurs, the PLR reroutes traffic through the bypass tunnel to the MP, allowing the traffic to continue to flow to the tunnel's egress router.

This document specifies a fast reroute framework for egress node and egress link protection. Similar to transit link/node protection, this framework also relies on a PLR to perform local failure detection and local repair. In egress node protection, the PLR is the penultimate-hop router of a tunnel. In egress link protection, the PLR is the egress router of the tunnel. The framework further uses a so-called "protector" to serve as the tailend of a bypass tunnel. The protector is a router that hosts "protection service instances" and has its own connectivity or paths to service destinations. When a PLR does local repair, the protector performs "context label switching" for rerouted MPLS service packets and "context IP forwarding" for rerouted IP service packets, to allow the service packets to continue to reach the service destinations.

This framework considers an egress node failure as a failure of a tunnel, and a failure of all the services carried by the tunnel, as service packets can no longer reach the service instances on the egress router. Therefore, the framework addresses egress node protection at both tunnel level and service level simultaneously. Likewise, the framework considers an egress link failure as a failure of all the services traversing the link, and addresses egress link protection at the service level.

This framework requires that the destination (a CE or site) of a service MUST be dual-homed or have dual paths to an MPLS network, via two MPLS edge routers. One of the routers is the egress router of the service's transport tunnel, and the other is a backup egress router which hosts a "backup service instance". In the "co-located" protector mode in this document, the backup egress router serves as the protector, and hence the backup service instance acts as the protection service instance. In the "centralized" protector mode (Section 5.12), the protector and the backup egress router are decoupled, and the protection service instance and the backup service instance are hosted separately by the two routers.

The framework is described by mainly referring to P2P (point-to-point) tunnels. However, it is equally applicable to P2MP (point-to-multipoint), MP2P (multipoint-to-point), and MP2MP (multipoint-to-multipoint) tunnels, as the sub-LSPs of these tunnels can be viewed as P2P tunnels.

The framework is a multi-service and multi-transport framework. It assumes a generic model where each service is comprised of a common set of components, including a service instance, a service label, a service label distribution protocol, and an MPLS transport tunnel. The framework also assumes the service label to be downstream assigned, i.e., assigned by an egress router. Therefore, the framework is generally applicable to most existing and future services. However, there are services with certain modes, where a protector is unable to pre-establish forwarding state for egress protection, or a PLR is not allowed to reroute traffic to other routers in order to avoid traffic duplication, e.g., the broadcast, multicast, and unknown unicast traffic in VPLS and EVPN. These cases are left for future study. Services which use upstream-assigned service labels are also out of scope of this document and left for future study.

The framework does not require extensions for the existing signaling and label distribution protocols (e.g., RSVP, LDP, BGP, etc.) of MPLS tunnels. It assumes transport tunnels and bypass tunnels to be established by using the generic procedures provided by the protocols. On the other hand, it does not preclude extensions to the

protocols which may facilitate the procedures. One example of such extension is [RFC8400]. The framework does see the need for extensions of IGPs and service label distribution protocols in some procedures, particularly for supporting protection establishment and context label switching. This document provides guidelines for these extensions, but leaves the specific details to separate documents.

The framework is intended to complement control-plane convergence and global repair. Control-plane convergence relies on control protocols to react on the topology changes due to a failure. Global repair relies an ingress router to remotely detect a failure and switch traffic to an alternative path. An example of global repair is the BGP Prefix Independent Convergence mechanism [BGP-PIC] for BGP established services. Compared with these mechanisms, this framework is considered as faster in traffic restoration, due to the nature of local failure detection and local repair. It is RECOMMENDED that the framework be used in conjunction with control-plane convergence or global repair, in order to take the advantages of both approaches. That is, the framework provides fast and temporary repair, while control-plane convergence or global repair provides ultimate and permanent repair.

2. Specification of Requirements

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] and [RFC8174].

3. Terminology

Egress router - A router at the egress endpoint of a tunnel. It hosts service instances for all the services carried by the tunnel, and has connectivity with the destinations of the services.

Egress node failure - A failure of an egress router.

Egress link failure - A failure of the egress link (e.g., PE-CE link, attachment circuit) of a service.

Egress failure - An egress node failure or an egress link failure.

Egress-protected tunnel - A tunnel whose egress router is protected by a mechanism according to this framework. The egress router is hence called a protected egress router.

Egress-protected service - An IP or MPLS service which is carried by an egress-protected tunnel, and hence protected by a mechanism according to this framework.

Backup egress router - Given an egress-protected tunnel and its egress router, this is another router which has connectivity with all or a subset of the destinations of the egress-protected services carried by the egress-protected tunnel.

Backup service instance - A service instance which is hosted by a backup egress router, and corresponding to an egress-protected service on a protected egress router.

Protector - A role acted by a router as an alternate of a protected egress router, to handle service packets in the event of an egress failure. A protector may be physically co-located with or decoupled from a backup egress router, depending on the co-located or centralized protector mode.

Protection service instance - A service instance hosted by a protector, corresponding to the service instance of an egress-protected service on a protected egress router. A protection service instance is a backup service instance, if the protector is co-located with a backup egress router.

PLR - A router at the point of local repair. In egress node protection, it is the penultimate-hop router on an egress-protected tunnel. In egress link protection, it is the egress router of the egress-protected tunnel.

Protected egress {E, P} - A virtual node consisting of an ordered pair of egress router E and protector P. It serves as the virtual destination of an egress-protected tunnel, and as the virtual location of the egress-protected services carried by the tunnel.

Context identifier (ID) - A globally unique IP address assigned to a protected egress {E, P}.

Context label - A non-reserved label assigned to a context ID by a protector.

Egress-protection bypass tunnel - A tunnel used to reroute service packets around an egress failure.

Co-located protector mode - The scenario where a protector and a backup egress router are co-located as one router, and hence each backup service instance serves as a protection service instance.

Centralized protector mode - The scenario where a protector is a dedicated router, and is decoupled from backup egress routers.

Context label switching - Label switching performed by a protector, in the label space of an egress router indicated by a context label.

Context IP forwarding - IP forwarding performed by a protector, in the IP address space of an egress router indicated by a context label.

4. Requirements

This document considers the following as the design requirements of this egress protection framework.

- The framework must support P2P tunnels. It should equally support P2MP, MP2P and MP2MP tunnels, by treating each sub-LSP as a P2P tunnel.

- The framework must support multi-service and multi-transport networks. It must accommodate existing and future signaling and label-distribution protocols of tunnels and bypass tunnels, including RSVP, LDP, BGP, IGP, segment routing, and others. It must also accommodate existing and future IP/MPLS services, including layer-2 VPNs, layer-3 VPNs, hierarchical LSP, and others. It MUST provide a general solution for networks where different types of services and tunnels co-exist.

- The framework must consider minimizing disruption during deployment. It should only involve routers close to egress, and be transparent to ingress routers and other transit routers.

- In egress node protection, for scalability and performance reasons, a PLR must be agnostic to services and service labels. It must maintain bypass tunnels and bypass forwarding state on a per-transport-tunnel basis, rather than on a per-service-destination or per-service-label basis. It should also support bypass tunnel sharing between transport tunnels.

- A PLR must be able to use its local visibility or information of routing or TE topology to compute or resolve a path for a bypass tunnel.

- A protector must be able to perform context label switching for rerouted MPLS service packets, based on service label(s) assigned by an egress router. It must be able to perform context IP forwarding for rerouted IP service packets, in the public or private IP address space used by an egress router.

- The framework must be able to work seamlessly with transit link/ node protection mechanisms to achieve end-to-end coverage.

- The framework must be able to work in conjunction with global repair and control plane convergence.

5. Egress node protection

5.1. Reference topology

This document refers to the following topology when describing the procedures of egress node protection.

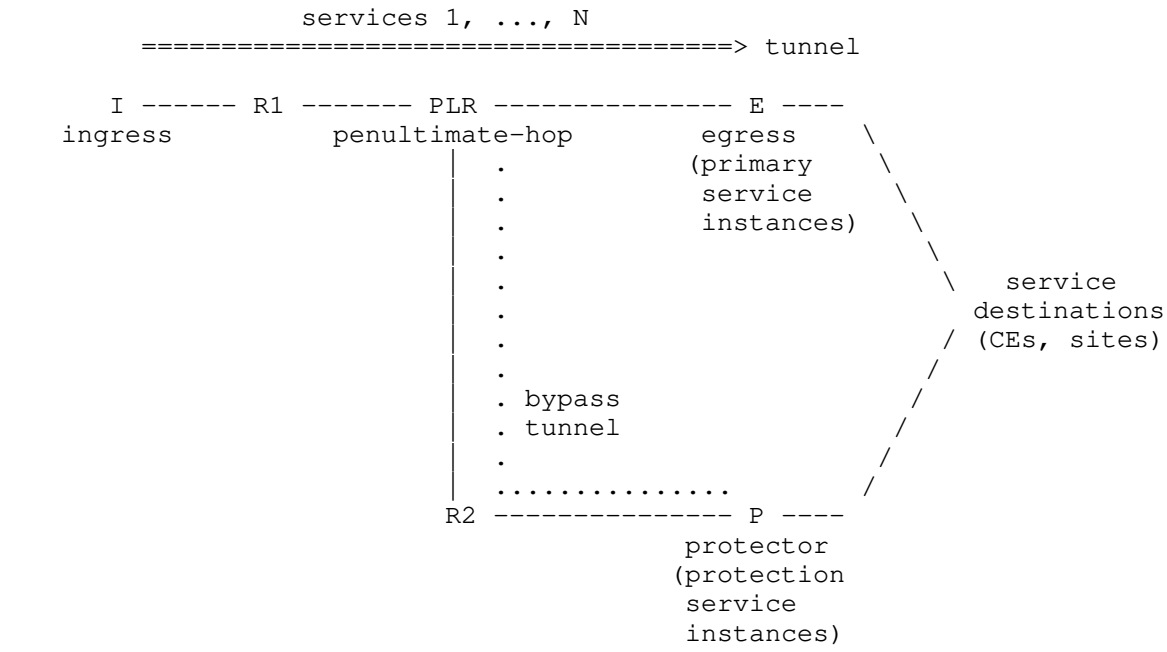


Figure 1

5.2. Egress node failure and detection

An egress node failure refers to the failure of an MPLS tunnel's egress router. At the service level, it is also a service instance failure for each IP/MPLS service carried by the tunnel.

An egress node failure can be detected by an adjacent router (i.e., PLR in this framework) through a node liveness detection mechanism,

or a mechanism based on a collective failure of all the links to that node. The mechanisms MUST be reasonably fast, i.e., faster than control plane failure detection and remote failure detection. Otherwise, local repair will not be able to provide much benefit compared to control plane convergence or global repair. In general, the speed, accuracy, and reliability of a failure detection mechanism are the key factors to decide its applicability in egress node protection. This document provides the following guidelines for network operators to choose a proper type of protection on a PLR.

- If the PLR has a mechanism to detect and differentiate a link failure (of the link between the PLR and the egress node) and an egress node failure, it SHOULD set up both link protection and egress node protection, and trigger one and only one protection upon a corresponding failure.

- If the PLR has a fast mechanism to detect a link failure and an egress node failure, but cannot distinguish them; or, if the PLR has a fast mechanism to detect a link failure only, but not an egress node failure, the PLR has two options:

  1. It MAY set up link protection only, and leave the egress node failure to be handled by global repair and control plane convergence.

  2. It MAY set up egress node protection only, and treat a link failure as a trigger for the egress node protection. The assumption is that treating a link failure as an egress node failure MUST NOT have a negative impact on services. Otherwise, it SHOULD adopt the previous option.

5.3. Protector and PLR

A router is assigned to the "protector" role to protect a tunnel and the services carried by the tunnel against an egress node failure. The protector is responsible for hosting a protection service instance for each protected service, serving as the tailend of a bypass tunnel, and performing context label switching and/or context IP forwarding for rerouted service packets.

A tunnel is protected by only one protector. Multiple tunnels to a given egress router may be protected by a common protector or different protectors. A protector may protect multiple tunnels with a common egress router or different egress routers.

For each tunnel, its penultimate-hop router acts as a PLR. The PLR pre-establishes a bypass tunnel to the protector, and pre-installs bypass forwarding state in the data plane. Upon detection of an

egress node failure, the PLR reroutes all the service packets received on the tunnel though the bypass tunnel to the protector. For MPLS service packets, the PLR keeps service labels intact in the packets. The protector in turn forwards the service packets towards the ultimate service destinations. Specifically, it performs context label switching for MPLS service packets, based on the service labels assigned by the protected egress router; it performs context IP forwarding for IP service packets, based on their destination addresses.

The protector MUST have its own connectivity with each service destination, via a direct link or a multi-hop path, which MUST NOT traverse the protected egress router or be affected by the egress node failure. This also means that each service destination MUST be dual-homed or have dual paths to the egress router and a backup egress router which may serve as the protector. Each protection service instance on the protector relies on such connectivity to set up forwarding state for context label switching and context IP forwarding.

5.4. Protected egress

This document introduces the notion of "protected egress" as a virtual node consisting of the egress router E of a tunnel and a protector P. It is denoted by an ordered pair of {E, P}, indicating the primary-and-protector relationship between the two routers. It serves as the virtual destination of the tunnel, and the virtual location of service instances for the services carried by the tunnel. The tunnel and services are considered as being "associated" with the protected egress {E, P}.

A given egress router E may be the tailend of multiple tunnels. In general, the tunnels may be protected by multiple protectors, e.g., P1, P2, and so on, with each Pi protecting a subset of the tunnels. Thus, these routers form multiple protected egresses, i.e., {E, P1}, {E, P2}, and so on. Each tunnel is associated with one and only one protected egress {E, Pi}. All the services carried by the tunnel are then automatically associated with the protected egress {E, Pi}. Conversely, a service associated with a protected egress {E, Pi} MUST be carried by a tunnel associated with the protected egress {E, Pi}. This mapping MUST be ensured by the ingress router of the tunnel and the service (Section 5.5).

Two routers X and Y may be protectors for each other. In this case, they form two distinct protected egresses {X, Y} and {Y, X}.

5.5. Egress-protected tunnel and service

A tunnel, which is associated with a protected egress {E, P}, is called an egress-protected tunnel. It is associated with one and only one protected egress {E, P}. Multiple egress-protected tunnels may be associated with a given protected egress {E, P}. In this case, they share the common egress router and protector, but may or may not share a common ingress router, or a common PLR (i.e., penultimate-hop router).

An egress-protected tunnel is considered as logically "destined" for its protected egress {E, P}. Its path MUST be resolved and established with E as the physical tailend.

A service, which is associated with a protected egress {E, P}, is called an egress-protected service. The egress router E hosts the primary instance of the service, and the protector P hosts the protection instance of the service.

An egress-protected service is associated with one and only one protected egress {E, P}. Multiple egress-protected services may be associated with a given protected egress {E, P}. In this case, these services share the common egress router and protector, but may or may not be carried by a common egress-protected tunnel or a common ingress router.

An egress-protected service MUST be mapped to an egress-protected tunnel by its ingress router, based on the common protected egress {E, P} of the service and the tunnel. This is achieved by introducing the notion of "context ID" for protected egress {E, P}, as described in (Section 5.7).

5.6. Egress-protection bypass tunnel

An egress-protected tunnel destined for a protected egress {E, P} MUST have a bypass tunnel from its PLR to the protector P. This bypass tunnel is called an egress-protection bypass tunnel. The bypass tunnel is considered as logically "destined" for the protected egress {E, P}. Due to its bypass nature, it MUST be established with P as the physical tailend and E as the node to avoid. The bypass tunnel MUST have the property that it MUST NOT be affected by the topology change caused by an egress node failure.

An egress-protection bypass tunnel is associated with one and only one protected egress {E, P}. A PLR may share an egress-protection bypass tunnel for multiple egress-protected tunnels associated with a common protected egress {E, P}.

5.7. Context ID, context label, and context-based forwarding

In this framework, a globally unique IPv4 or IPv6 address is assigned to a protected egress {E, P} as the identifier of the protected egress {E, P}. It is called a "context ID" due to its specific usage in context label switching and context IP forwarding on the protector. It is an IP address that is logically owned by both the egress router and the protector. For the egress router, it indicates the protector. For the protector, it indicates the egress router, particularly the egress router's forwarding context. For other routers in the network, it is an address reachable via both the egress router and the protector (Section 5.8), similar to an anycast address.

The main purpose of a context ID is to coordinate ingress router, egress router, PLR and protector to establish egress protection. The procedures are described below, given an egress-protected service associated with a protected egress {E, P} with context ID.

- If the service is an MPLS service, when E distributes a service label binding message to the ingress router, E attaches the context ID to the message. If the service is an IP service, when E advertises the service destination address to the ingress router, E attaches the context ID to the advertisement message. How the context ID is encoded in the messages is a choice of the service protocol. A protocol extension of a "context ID" object may be needed, if there is no existing mechanism for this purpose.

- The ingress router uses the service's context ID as the destination to establish or resolve an egress-protected tunnel. The ingress router then maps the service to the tunnel for transportation. The semantics of the context ID is transparent to the ingress router. The ingress router only treats the context ID as an IP address of E, in the same manner as establishing or resolving a regular transport tunnel.

- The context ID is conveyed to the PLR by the signaling protocol of the egress-protected tunnel, or learned by the PLR via an IGP (i.e., OSPF or ISIS) or a topology-driven label distribution protocol (e.g., LDP). The PLR uses the context ID as destination to establish or resolve an egress-protection bypass tunnel to P while avoiding E.

- P maintains a dedicated label space and a dedicated IP address space for E. They are referred to as "E's label space" and "E's IP address space", respectively. P uses the context ID to identify the label space and IP address space.

- If the service is an MPLS service, E also distributes the service label binding message to P. This is the same label binding message that E advertises to the ingress router, which includes the context ID. Based on the context ID, P installs the service label in an MPLS forwarding table corresponding to E's label space. If the service is an IP service, P installs an IP route in an IP forwarding table corresponding to E's IP address space. In either case, the protection service instance on P constructs forwarding state for the label route or IP route based on P's own connectivity with the service's destination.

- P assigns a non-reserved label to the context ID. In the data plane, this label represents the context ID and indicates E's label space and IP address space. Therefore, it is called a "context label".

- The PLR may establish the egress-protection bypass tunnel to P in several manners. If the bypass tunnel is established by RSVP, the PLR signals the bypass tunnel with the context ID as destination, and P binds the context label to the bypass tunnel. If the bypass tunnel is established by LDP, P advertises the context label for the context ID as an IP prefix FEC. If the bypass tunnel is established by the PLR in a hierarchical manner, the PLR treats the context label as a one-hop LSP over a regular bypass tunnel to P (e.g., a bypass tunnel to P's loopback IP address). If the bypass tunnel is constructed by using segment routing, the bypass tunnel is represented by a stack of SID labels with the context label as the inner-most SID label (Section 5.9). In any case, the bypass tunnel is a ultimate-hop-popping (UHP) tunnel whose incoming label on P is the context label.

- During local repair, all the service packets received by P on the bypass tunnel have the context label as the top label. P first pops the context label. For an MPLS service packet, P further looks up the service label in E's label space indicated by the context label. Such kind of forwarding is called context label switching. For an IP service packet, P looks up the IP destination address in E's IP address space indicated by the context label. Such kind of forwarding is called context IP forwarding.

5.8. Advertisement and path resolution for context ID

Path resolution and computation for a context ID are done on ingress routers for egress-protected tunnels, and on PLRs for egress-protection bypass tunnels. Given a protected egress {E, P} and its context ID, E and P MUST coordinate on the reachability of the context ID in the routing domain and the TE domain. The context ID

MUST be advertised in such a manner that all egress-protected tunnels MUST have E as tailend, and all egress-protection bypass tunnels MUST have P as tailend while avoiding E.

This document suggests three approaches:

1. The first approach is called "proxy mode". It requires E and P, but not the PLR, to have the knowledge of the egress protection schema. E and P advertise the context ID as a virtual proxy node (i.e., a logical node) connected to the two routers, with the link between the proxy node and E having more preferable IGP and TE metrics than the link between the proxy node and P. Therefore, all egress-protected tunnels destined for the context ID will automatically follow the IGP or TE paths to E. Each PLR will no longer view itself as a penultimate-hop, but rather two hops away from the proxy node, via E. The PLR will be able to find a bypass path via P to the proxy node, while the bypass tunnel is actually be terminated by P.

2. The second approach is called "alias mode". It requires P and the PLR, but not E, to have the knowledge of the egress protection schema. E simply advertises the context ID as an IP address. P advertises the context ID and the context label by using a "context ID label binding" advertisement. In both routing domain and TE domain, the context ID is only reachable via E. Therefore, all egress-protected tunnels destined for the context ID will have E as tailend. Based on the "context ID label binding" advertisement, the PLR can establish an egress-protection bypass tunnel in several manners (Section 5.9). The "context ID label binding" advertisement is defined as IGP mirroring context segment in [RFC8402] and [SR-ISIS]. These IGP extensions are generic in nature, and hence can be used for egress protection purposes. It is RECOMMENDED that a similar advertisement be defined for OSPF as well.

3. The third approach is called "stub link mode". In this mode, both E and P advertise the context ID as a link to a stub network, essentially modelling the context ID as an anycast IP address owned by the two routers. E, P and the PLR do not need to have the knowledge of the egress protection schema. The correctness of the egress-protected tunnels and the bypass tunnels relies on the path computations for the anycast IP address performed by the ingress routers and PLR. Therefore, care MUST be taken for the applicability of this approach to a network.

This framework considers the above approaches as technically equal, and the feasibility of each approach in a given network as dependent

on the topology, manageability, and available protocols of the network. For a given context ID, all relevant routers, including primary PE, protector, and PLR, MUST support and agree on the chosen approach. The coordination between these routers can be achieved by configuration.

In a scenario where an egress-protected tunnel is an inter-area or inter-AS tunnel, its associated context ID MUST be propagated by IGP or BGP from the original area or AS to the area or AS of the ingress router. The propagation process of the context ID SHOULD be the same as that of an IP address in an inter-area or inter-AS environment.

5.9. Egress-protection bypass tunnel establishment

A PLR MUST know the context ID of a protected egress {E, P} in order to establish an egress-protection bypass tunnel. The information is obtained from the signaling or label distribution protocol of the egress-protected tunnel. The PLR may or may not need to have the knowledge of the egress protection schema. All it does is to set up a bypass tunnel to a context ID while avoiding the next-hop router (i.e., egress router). This is achievable by using a constraint-based computation algorithm similar to those commonly used for traffic engineering paths and loop-free alternate (LFA) paths. Since the context ID is advertised in the routing domain and the TE domain by IGP according to Section 5.8, the PLR is able to resolve or establish such a bypass path with the protector as tailend. In the case of proxy mode, the PLR may do so in the same manner as transit node protection.

An egress-protection bypass tunnel may be established via several methods:

(1) It may be established by a signaling protocol (e.g., RSVP), with the context ID as destination. The protector binds the context label to the bypass tunnel.

(2) It may be formed by a topology driven protocol (e.g., LDP with various LFA mechanisms). The protector advertises the context ID as an IP prefix FEC, with the context label bound to it.

(3) It may be constructed as a hierarchical tunnel. When the protector uses the alias mode (Section 5.8), the PLR will have the knowledge of the context ID, context label, and protector (i.e., the advertiser). The PLR can then establish the bypass tunnel in a hierarchical manner, with the context label as a one-hop LSP over a regular bypass tunnel to the protector's IP address (e.g., loopback address). This regular bypass tunnel may be established by RSVP, LDP, segment routing, or another protocol.

5.10. Local repair on PLR

In this framework, a PLR is agnostic to services and service labels. This obviates the need to maintain bypass forwarding state on a per-service basis, and allows bypass tunnel sharing between egress-protected tunnels. The PLR may share an egress-protection bypass tunnel for multiple egress-protected tunnels associated with a common protected egress {E, P}. During local repair, the PLR reroutes all service packets received on the egress-protected tunnels to the egress-protection bypass tunnel. Service labels remain intact in MPLS service packets.

Label operation performed by the PLR depends on the bypass tunnel's characteristics. If the bypass tunnel is a single level tunnel, the rerouting will involve swapping the incoming label of an egress-protected tunnel to the outgoing label of the bypass tunnel. If the bypass tunnel is a hierarchical tunnel, the rerouting will involve swapping the incoming label of an egress-protected tunnel to a context label, and pushing the outgoing label of a regular bypass tunnel. If the bypass tunnel is constructed by segment routing, the rerouting will involve swapping the incoming label of an egress-protected tunnel to a context label, and pushing the stack of SID labels of the bypass tunnel.

5.11. Service label distribution from egress router to protector

When a protector receives a rerouted MPLS service packet, it performs context label switching based on the packet's service label which is assigned by the corresponding egress router. In order to achieve this, the protector MUST maintain the labels of egress-protected services in dedicated label spaces on a per protected egress {E, P} basis, i.e., one label space for each egress router that it protects.

Also, there MUST be a service label distribution protocol session between each egress router and the protector. Through this protocol, the protector learns the label binding of each egress-protected service. This is the same label binding that the egress router advertises to the service's ingress router, which includes a context ID. The corresponding protection service instance on the protector recognizes the service, and resolves forwarding state based on its own connectivity with the service's destination. It then installs the service label with the forwarding state in the label space of the egress router, which is indicated by the context ID (i.e., context label).

Different service protocols may use different mechanisms for such kind of label distribution. Specific extensions may be needed on a per-protocol basis or per-service-type basis. The details of the

extensions should be specified in separate documents. As an example, [RFC8104] specifies the LDP extensions for pseudowire services.

5.12. Centralized protector mode

In this framework, it is assumed that the service destination of an egress-protected service MUST be dual-homed to two edge routers of an MPLS network. One of them is the protected egress router, and the other is a backup egress router. So far in this document, the discussion has been focusing on the scenario where a protector and a backup egress router are co-located as one router. Therefore, the number of protectors in a network is equal to the number of backup egress routers. As another scenario, a network may assign a small number of routers to serve as dedicated protectors, each protecting a subset of egress routers. These protectors are called centralized protectors.

Topologically, a centralized protector may be decoupled from all backup egress routers, or it may be co-located with one backup egress router while decoupled from the other backup egress routers. The procedures in this section assume that a protector and a backup egress router are decoupled.

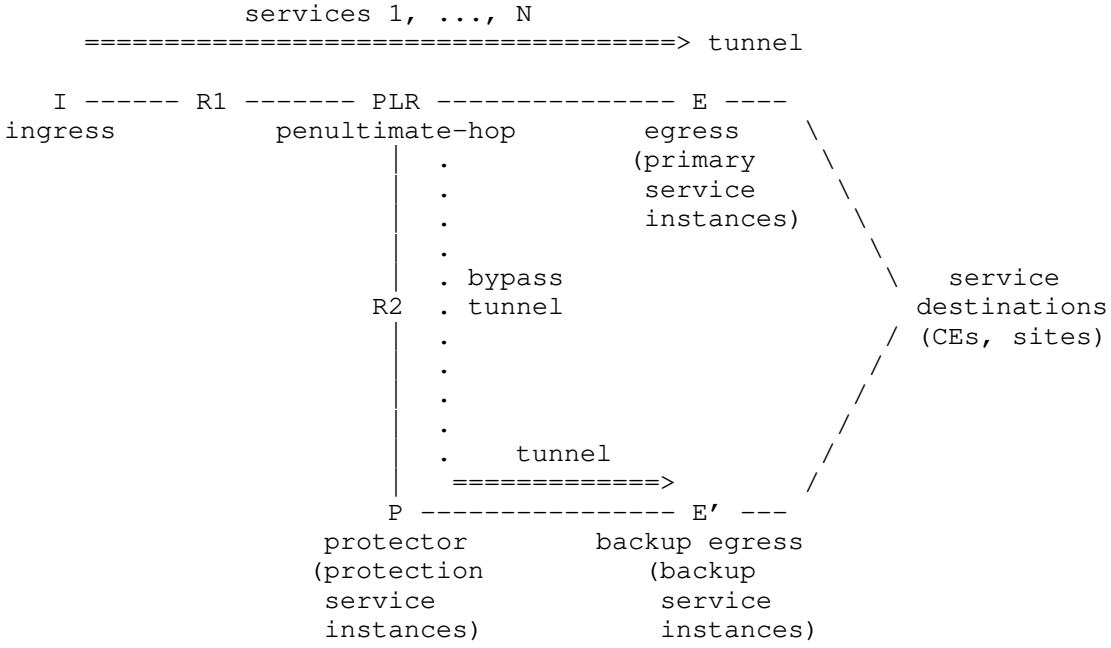


Figure 2

Like a co-located protector, a centralized protector hosts protection service instances, receives rerouted service packets from PLRs, and performs context label switching and/or context IP forwarding. For each service, instead of sending service packets directly to the service destination, the protector MUST send them via another transport tunnel to the corresponding backup service instance on a backup egress router. The backup service instance in turn forwards the service packets to the service destination. Specifically, if the service is an MPLS service, the protector MUST swap the service label in each received service packet to the label of the backup service advertised by the backup egress router, and then push the label (or label stack) of the transport tunnel.

In order for a centralized protector to map an egress-protected MPLS service to a service hosted on a backup egress router, there MUST be a service label distribution protocol session between the backup egress router and the protector. Through this session, the backup egress router advertises the service label of the backup service, attached with the FEC of the egress-protected service and the context ID of the protected egress {E, P}. Based on this information, the protector associates the egress-protected service with the backup service, resolves or establishes a transport tunnel to the backup

egress router, and sets up forwarding state for the label of the egress-protected service in the label space of the egress router.

The service label which the backup egress router advertises to the protector can be the same as the label which the backup egress router advertises to the ingress router(s), if and only if the forwarding state of the label does not direct service packets towards the protected egress router. Otherwise, the label MUST NOT be used for egress protection, because it would create a loop for the service packets. In this case, the backup egress router MUST advertise a unique service label for egress protection, and set up the forwarding state of the label to use the backup egress router's own connectivity with the service destination.

6. Egress link protection

Egress link protection is achievable through procedures similar to that of egress node protection. In normal situations, an egress router forwards service packets to a service destination based on a service label, whose forwarding state points to an egress link. In egress link protection, the egress router acts as the PLR, and performs local failure detection and local repair. Specifically, the egress router pre-establishes an egress-protection bypass tunnel to a protector, and sets up the bypass forwarding state for the service label to point to the bypass tunnel. During local repair, the egress router reroutes service packets via the bypass tunnel to the protector. The protector in turn forwards the packets to the service destination (in the co-located protector mode, as shown in Figure 3), or forwards the packets to a backup egress router (in the centralized protector mode, as shown in Figure 4).

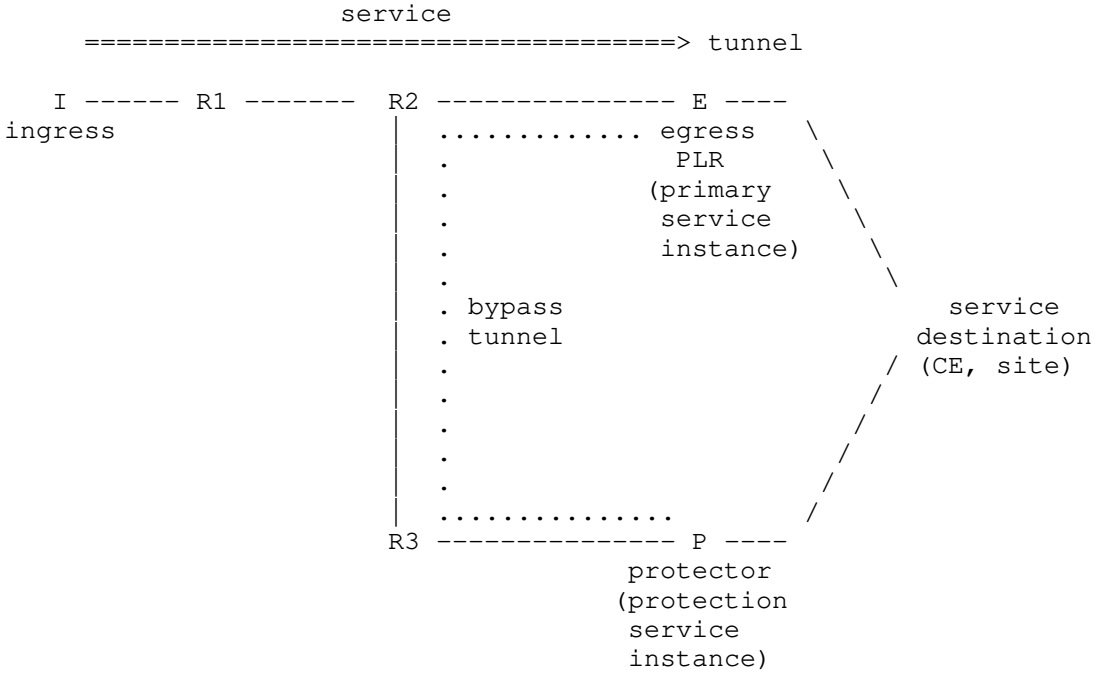

service
tunnel
I
R1
R2
E
ingress
egress
PLR
(primary
service
instance)
bypass
tunnel
service
destination
(CE, site)
R3
P
protector
(protection
service
instance)

Figure 3

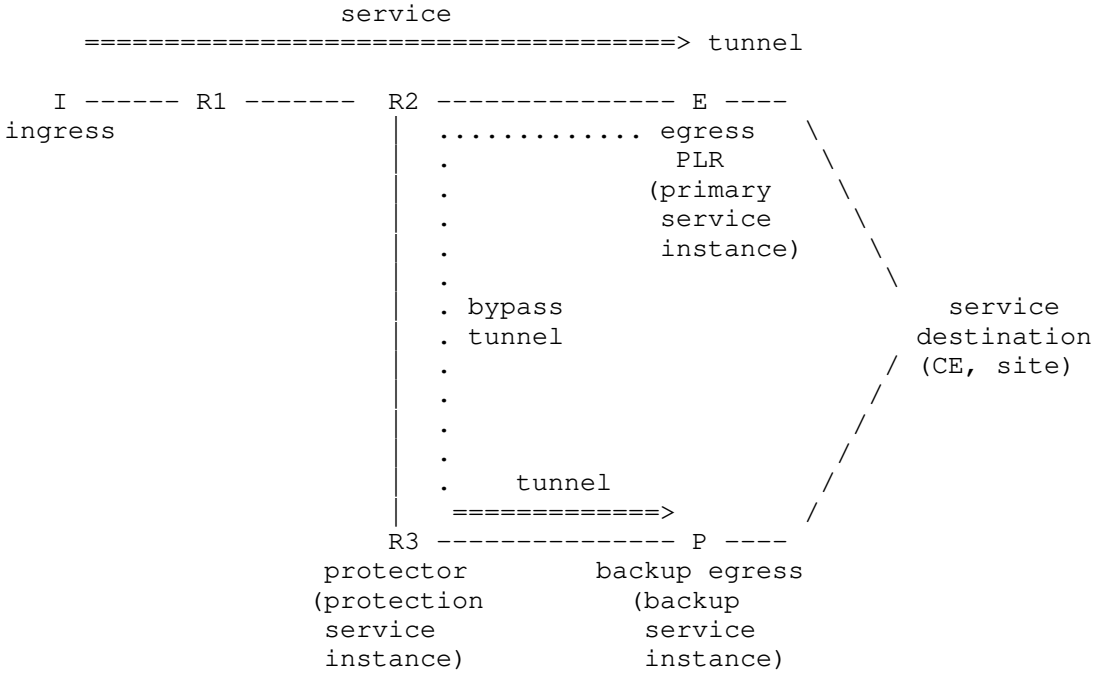


Figure 4

There are two approaches to set up the bypass forwarding state on the egress router, depending on whether the egress router knows the service label allocated by the backup egress router. The difference is that one approach requires the protector to perform context label switching, and the other one does not. Both approaches are equally supported by this framework.

(1) The first approach applies when the egress router does not know the service label allocated by the backup egress router. In this case, the egress router sets up the bypass forwarding state as a label push with the outgoing label of the egress-protection bypass tunnel. Rerouted packets will have the egress router's service label intact. Therefore, the protector MUST perform context label switching, and the bypass tunnel MUST be destined for the context ID of the protected egress {E, P} and established as described in Section 5.9. This approach is consistent with egress node protection. Hence, a protector can serve in egress node protection and egress link protection in a consistent manner, and both the co-located protector mode and the centralized protector mode are supported (Figure 3 and Figure 4).

(2) The second approach applies when the egress router knows the service label allocated by the backup egress router, via a label distribution protocol session. In this case, the backup egress router serves as the protector for egress link protection, regardless of the protector of egress node protection, which will be the same router in the co-located protector mode but a different router in the centralized protector mode. The egress router sets up the bypass forwarding state as a label swap from the incoming service label to the service label of the backup egress router (i.e., protector), followed by a push with the outgoing label (or label stack) of the egress link protection bypass tunnel. The bypass tunnel is a regular tunnel destined for an IP address of the protector, instead of the context ID of the protected egress {E, P}. The protector simply forwards rerouted service packets based on its own service label, rather than performing context label switching. In this approach, only the co-located protector mode is applicable.

Note that for a bidirectional service, the physical link of an egress link may carry service traffic bi-directionally. Therefore, an egress link failure may simultaneously be an ingress link failure for the traffic in the opposite direction. Protection for ingress link failure SHOULD be provided by a separate mechanism, and hence is out of the scope of this document.

7. Global repair

This framework provides a fast but temporary repair for egress node and egress link failures. For permanent repair, the services affected by a failure SHOULD be moved to an alternative tunnel, or replaced by alternative services, which are fully functional. This is referred to as global repair. Possible triggers of global repair include control plane notifications of tunnel status and service status, end-to-end OAM and fault detection at tunnel and service level, and others. The alternative tunnel and services may be pre-established in standby state, or dynamically established as a result of the triggers or network protocol convergence.

8. Operational Considerations

When a PLR performs local repair, the router SHOULD generate an alert for the event. The alert may be logged locally for tracking purposes, or it may be sent to the operator at a management station. The communication channel and protocol between the PLR and the management station may vary depending on networks, and are out of the scope of this document.

9. General context-based forwarding

So far, this document has been focusing on the cases where service packets are MPLS or IP packets and protectors perform context label switching or context IP forwarding. Although this should cover most common services, it is worth mentioning that the framework is also applicable to services or sub-modes of services where service packets are layer-2 packets or encapsulated in non-IP/MPLS formats. The only specific in these cases is that a protector MUST perform context-based forwarding based on the layer-2 table or corresponding lookup table which is indicated by a context ID (i.e., context label).

10. Example: Layer-3 VPN egress protection

This section shows an example of egress protection for layer-3 IPv4 and IPv6 VPNs.

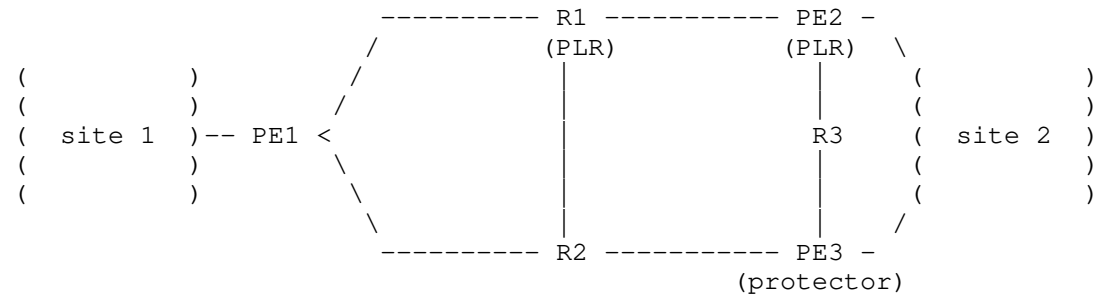


Figure 5

In this example, the core network is IPv4 and MPLS. Both of the IPv4 VPN and the IPv6 VPN consist of site 1 and site 2. Site 1 is connected to PE1, and site 2 is dual-homed to PE2 and PE3. Site 1 includes an IPv4 subnet 203.0.113.64/26 and an IPv6 subnet 2001:db8:1:1::/64. Site 2 includes an IPv4 subnet 203.0.113.128/26 and an IPv6 subnet 2001:db8:1:2::/64. PE2 is the primary PE for site 2, and PE3 is the backup PE. Each of PE1, PE2 and PE3 hosts an IPv4 VPN instance and an IPv6 VPN instance. The PEs use BGP to exchange VPN prefixes and VPN labels between each other. In the core network, R1 and R2 are transit routers, OSPF is used as the routing protocol, and RSVP-TE as the tunnel signaling protocol.

Using the framework in this document, the network assigns PE3 to be the protector of PE2 to protect the VPN traffic in the direction from site 1 to site 2. This is the co-located protector mode. PE2 and

PE3 form a protected egress {PE2, PE3}. A context ID 198.51.100.1 is assigned to the protected egress {PE2, PE3}. (If the core network is IPv6, the context ID would be an IPv6 address.) The IPv4 and IPv6 VPN instances on PE3 serve as protection instances for the corresponding VPN instances on PE2. On PE3, a context label 100 is assigned to the context ID, and a label table pe2.mpls is created to represent PE2's label space. PE3 installs label 100 in its MPLS forwarding table, with nexthop pointing to the label table pe2.mpls. PE2 and PE3 are coordinated to use the proxy mode to advertise the context ID in the routing domain and the TE domain.

PE2 uses per-VRF label allocation mode for both of its IPv4 and IPv6 VPN instances. It assigns label 9000 to the IPv4 VRF, and label 9001 to the IPv6 VRF. For the IPv4 prefix 203.0.113.128/26 in site 2, PE2 advertises it with label 9000 and NEXT_HOP 198.51.100.1 to PE1 and PE3 via BGP. Likewise, for the IPv6 prefix 2001:db8:1:2::/64 in site 2, PE2 advertises it with label 9001 and NEXT_HOP 198.51.100.1 to PE1 and PE3 via BGP.

PE3 also uses per-VRF VPN label allocation mode for both of its IPv4 and IPv6 VPN instances. It assigns label 10000 to the IPv4 VRF, and label 10001 to the IPv6 VRF. For the prefix 203.0.113.128/26 in site 2, PE3 advertises it with label 10000 and NEXT_HOP as itself to PE1 and PE2 via BGP. For the IPv6 prefix 2001:db8:1:2::/64 in site 2, PE3 advertises it with label 10001 and NEXT_HOP as itself to PE1 and PE2 via BGP.

Upon receipt of the above BGP advertisements from PE2, PE1 uses the context ID 198.51.100.1 as destination to compute a path for an egress-protected tunnel. The resultant path is PE1->R1->PE2. PE1 then uses RSVP to signal the tunnel, with the context ID 198.51.100.1 as destination, and with the "node protection desired" flag set in the SESSION_ATTRIBUTE of RSVP Path message. Once the tunnel comes up, PE1 maps the VPN prefixes 203.0.113.128/26 and 2001:db8:1:2::/64 to the tunnel, and installs a route for each prefix in the corresponding IPv4 or IPv6 VRF. The nexthop of the route 203.0.113.128/26 is a push of the VPN label 9000, followed by a push of the outgoing label of the egress-protected tunnel. The nexthop of the route 2001:db8:1:2::/64 is a push of the VPN label 9001, followed by a push of the outgoing label of the egress-protected tunnel.

Upon receipt of the above BGP advertisements from PE2, PE3 recognizes the context ID 198.51.100.1 in the NEXT_HOP attribute, and installs a route for label 9000 and a route for label 9001 in the label table pe2.mpls. PE3 sets the nexthop of the route 9000 to the IPv4 protection VRF, and the nexthop of the route 9001 to the IPv6 protection VRF. The IPv4 protection VRF contains the routes to the IPv4 prefixes in site 2. The IPv6 protection VRF contains the routes

to the IPv6 prefixes in site 2. The nexthops of these routes must be based on PE3's connectivity with site 2, even if the connectivity may not have the best metrics (e.g., MED, local preference, etc.) to be used in PE3's own VRF. The nexthops must not use any path traversing PE2. Note that the protection VRFs are a logical concept, and they may simply be PE3's own VRFs if they satisfies the requirement.

10.1. Egress node protection

R1, i.e., the penultimate-hop router of the egress-protected tunnel, serves as the PLR for egress node protection. Based on the "node protection desired" flag and the destination address (i.e., context ID 198.51.100.1) of the tunnel, R1 computes a bypass path to 198.51.100.1 while avoiding PE2. The resultant bypass path is R1->R2->PE3. R1 then signals the path (i.e., egress-protection bypass tunnel), with 198.51.100.1 as destination.

Upon receipt of an RSVP Path message of the egress-protection bypass tunnel, PE3 recognizes the context ID 198.51.100.1 as the destination, and responds with the context label 100 in an RSVP Resv message.

After the egress-protection bypass tunnel comes up, R1 installs a bypass nexthop for the egress-protected tunnel. The bypass nexthop is a label swap from the incoming label of the egress-protected tunnel to the outgoing label of the egress-protection bypass tunnel.

When R1 detects a failure of PE2, it will invoke the above bypass nexthop to reroute VPN packets. Each IPv4 VPN packet will have the label of the bypass tunnel as outer label, and the IPv4 VPN label 9000 as inner label. Each IPv6 VPN packets will have the label of the bypass tunnel as outer label, and the IPv6 VPN label 9001 as inner label. When the packets arrive at PE3, they will have the context label 100 as outer label, and the VPN label 9000 or 9001 as inner label. The context label will first be popped, and then the VPN label will be looked up in the label table pe2.mpls. The lookup will cause the VPN label to be popped, and the IPv4 and IPv6 packets to be forwarded to site 2 based on the IPv4 and IPv6 protection VRFs, respectively.

10.2. Egress link protection

PE2 serves as the PLR for egress link protection. It has already learned PE3's IPv4 VPN label 10000 and IPv6 VPN label 10001. Hence it uses the approach (2) described in Section 6 to set up bypass forwarding state. It signals an egress-protection bypass tunnel to PE3, by using the path PE2->R3->PE3, and PE3's IP address as destination. After the bypass tunnel comes up, PE2 installs a bypass

nexthop for the IPv4 VPN label 9000, and a bypass nexthop for the IPv6 VPN label 9001. For label 9000, the bypass nexthop is a label swap to label 10000, followed by a label push with the outgoing label of the bypass tunnel. For label 9001, the bypass nexthop is a label swap to label 10001, followed by a label push with the outgoing label of the bypass tunnel.

When PE2 detects a failure of the egress link, it will invoke the above bypass nexthop to reroute VPN packets. Each IPv4 VPN packet will have the label of the bypass tunnel as outer label, and label 10000 as inner label. Each IPv6 VPN packet will have the label of the bypass tunnel as outer label, and label 10001 as inner label. When the packets arrive at PE3, the VPN label 10000 or 10001 will be popped, and the exposed IPv4 and IPv6 packets will be forwarded based on PE3's IPv4 and IPv6 VRFs, respectively.

10.3. Global repair

Eventually, global repair will take effect, as control plane protocols converge on the new topology. PE1 will choose PE3 as a new entrance to site 2. Before that happens, the VPN traffic has been protected by the above local repair.

10.4. Other modes of VPN label allocation

It is also possible that PE2 may use per-route or per-interface VPN label allocation mode. In either case, PE3 will have multiple VPN label routes in the pe2.mpls table, corresponding to the VPN labels advertised by PE2. PE3 forwards rerouted packets by popping a VPN label and performing an IP lookup in the corresponding protection VRF. PE3's forwarding behavior is consistent with the above case where PE2 uses per-VRF VPN label allocation mode. PE3 does not need to know PE2's VPN label allocation mode, or construct a specific nexthop for each VPN label route in the pe2.mpls table.

11. IANA Considerations

This document has no request for new IANA allocation.

12. Security Considerations

The framework in this document involves rerouting traffic around an egress node or link failure, via a bypass path from a PLR to a protector, and ultimately to a backup egress router. The forwarding performed by the routers in the data plane is anticipated, as part of the planning of egress protection.

Control plane protocols MAY be used to facilitate the provisioning of the egress protection on the routers. In particular, the framework requires a service label distribution protocol between an egress router and a protector over a secure session. The security properties of this provisioning and label distribution depend entirely on the underlying protocol chosen to implement these activities . Their associated security considerations apply. This framework introduces no new security requirements or guarantees relative to these activities.

Also, the PLR, protector, and backup egress router are located close to the protected egress router, and normally in the same administrative domain. If they are not in the same administrative domain, a certain level of trust MUST be established between them in order for the protocols to run securely across the domain boundary. The basis of this trust is the security model of the protocols (as described above), and further security considerations for inter-domain scenarios should be addressed by the protocols as a common requirement.

Security attacks may sometimes come from a customer domain. Such kind of attacks are not introduced by the framework in this document, and may occur regardless of the existence of egress protection. In one possible case, the egress link between an egress router and a CE could become a point of attack. An attacker that gains control of the CE might use it to simulate link failures and trigger constant and cascading activities in the network. If egress link protection is in place, egress link protection activities may also be triggered. As a general solution to defeat the attack, a damping mechanism SHOULD be used by the egress router to promptly suppress the services associated with the link or CE. The egress router would stop advertising the services, essentially detaching them from the network and eliminating the effect of the simulated link failures.

From the above perspectives, this framework does not introduce any new security threat to networks.

13. Acknowledgements

This document leverages work done by Yakov Rekhter, Kevin Wang and Zhaohui Zhang on MPLS egress protection. Thanks to Alexander Vainshtein, Rolf Winter, Lizhong Jin, Krzysztof Szarkowicz, Roman Danyliw, and Yuanlong Jiang for their valuable comments that helped to shape this document and improve its clarity.

14. References

14.1. Normative References

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

[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>.

[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>.

[SR-ISIS] Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions (work in progress), 2017.

14.2. Informative References

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

[RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <https://www.rfc-editor.org/info/rfc7490>.

[RFC7812] Atlas, A., Bowers, C., and G. Enyedi, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)", RFC 7812, DOI 10.17487/RFC7812, June 2016, <https://www.rfc-editor.org/info/rfc7812>.

[RFC8104] Shen, Y., Aggarwal, R., Henderickx, W., and Y. Jiang, "Pseudowire (PW) Endpoint Fast Failure Protection", RFC 8104, DOI 10.17487/RFC8104, March 2017, <https://www.rfc-editor.org/info/rfc8104>.

[RFC8400] Chen, H., Liu, A., Saad, T., Xu, F., and L. Huang, "Extensions to RSVP-TE for Label Switched Path (LSP) Egress Protection", RFC 8400, DOI 10.17487/RFC8400, June 2018, <https://www.rfc-editor.org/info/rfc8400>.

[BGP-PIC] Bashandy, P., Filsfils, C., and P. Mohapatra, "BGP Prefix Independent Convergence", draft-ietf-rtgwg-bgp-pic-09.txt (work in progress), 2017.

Authors' Addresses

Yimin Shen
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

Phone: +1 9785890722
Email: yshen@juniper.net

Minto Jeyananth
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

Phone: +1 4089367563
Email: minto@juniper.net

Bruno Decraene
Orange

Email: bruno.decraene@orange.com

Hannes Gredler
RtBrick Inc

Email: hannes@rtbrick.com

Carsten Michel
Deutsche Telekom

Email: c.michel@telekom.de

Huaimo Chen
Huawei Technologies Co., Ltd.

Email: huaimo.chen@huawei.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 4, 2019

H. Sitaraman
V. Beeram
Juniper Networks
T. Parikh
Verizon
T. Saad
Cisco Systems

January 31, 2019

Signaling RSVP-TE tunnels on a shared MPLS forwarding plane
draft-ietf-mpls-rsvp-shared-labels-09.txt

Abstract

As the scale of MPLS RSVP-TE networks has grown, so the number of Label Switched Paths (LSPs) supported by individual network elements has increased. Various implementation recommendations have been proposed to manage the resulting increase in control plane state.

However, those changes have had no effect on the number of labels that a transit Label Switching Router (LSR) has to support in the forwarding plane. That number is governed by the number of LSPs transiting or terminated at the LSR and is directly related to the total LSP state in the control plane.

This document defines a mechanism to prevent the maximum size of the label space limit on an LSR from being a constraint to control plane scaling on that node. It introduces the notion of pre-installed 'per Traffic Engineering (TE) link labels' that can be shared by MPLS RSVP-TE LSPs that traverse these TE links. This approach significantly reduces the forwarding plane state required to support a large number of LSPs. This couples the feature benefits of the RSVP-TE control plane with the simplicity of the Segment Routing MPLS forwarding plane.

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 http://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 4, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction 3
2. Terminology 5
3. Allocation of TE Link Labels 5
4. Segment Routed RSVP-TE Tunnel Setup 5
5. Delegating Label Stack Imposition 7
5.1. Stacking at the Ingress 8
5.1.1. Stack to Reach Delegation Hop 8
5.1.2. Stack to Reach Egress 9
5.2. Explicit Delegation 10
5.3. Automatic Delegation 10
5.3.1. Effective Transport Label-Stack Depth (ETLD) 10
6. Mixing TE Link Labels and Regular Labels in an RSVP-TE Tunnel 12
7. Construction of Label Stacks 12
8. Facility Backup Protection 13

8.1. Link Protection 13
9. Protocol Extensions 14
9.1. Requirements 14
9.2. Attribute Flags TLV: TE Link Label 15
9.3. RRO Label Subobject Flag: TE Link Label 15
9.4. Attribute Flags TLV: LSI-D 15
9.5. RRO Label Subobject Flag: Delegation Label 16
9.6. Attributes Flags TLV: LSI-D-S2E 16
9.7. Attributes TLV: ETLD 16
10. OAM Considerations 17
11. Acknowledgements 17
12. Contributors 17
13. IANA Considerations 18
13.1. Attribute Flags: TE Link Label, LSI-D, LSI-D-S2E 18
13.2. Attribute TLV: ETLD 18
13.3. Record Route Label Sub-object Flags: TE Link Label, Delegation Label 19
13.4. Error Codes and Error Values 19
14. Security Considerations 19
15. References 20
15.1. Normative References 20
15.2. Informative References 21
Authors' Addresses 21

1. Introduction

The scaling of RSVP-TE [RFC3209] control plane implementations can be improved by adopting the guidelines and mechanisms described in [RFC2961] and [RFC8370]. These documents do not make any difference to the forwarding plane state required to handle the control plane state. The forwarding plane state remains unchanged and is directly proportional to the total number of Label Switching Paths (LSPs) supported by the control plane.

This document describes a mechanism that prevents the size of the platform specific label space on a Label Switching Router (LSR) from being a constraint to pushing the limits of control plane scaling on that node.

This work introduces the notion of pre-installed 'per Traffic Engineering (TE) link labels' that are allocated by an LSR. Each such label is installed in the MPLS forwarding plane with a 'pop' operation and the instruction to forward the received packet over the TE link. An LSR advertises this label in the Label object of a Resv message as LSPs are set up and they are recorded hop-by-hop in the Record Route object (RRO) of the Resv message as it traverses the network. To make use of this feature, the ingress Label Edge Router (LER) pushes a stack of labels [RFC3031] as received in the RRO.

These 'TE link labels' can be shared by MPLS RSVP-TE LSPs that traverse the same TE link.

This forwarding plane behavior fits in the MPLS architecture [RFC3031] and is same as that exhibited by Segment Routing (SR) [RFC8402] when using an MPLS forwarding plane and a series of adjacency segments [I-D.ietf-spring-segment-routing-mpls]. This work couples the feature benefits of the RSVP-TE control plane with the simplicity of the Segment Routing MPLS forwarding plane.

RSVP-TE using a shared MPLS forwarding plane offers the following benefits:

1. Shared Labels: The transit label on a TE link is shared among RSVP-TE tunnels traversing the link and is used independent of the ingress and egress of the LSPs.

2. Faster LSP setup time: No forwarding plane state needs to be programmed during LSP setup and teardown resulting in faster time for provisioning and deprovisioning LSPs.

3. Hitless re-routing: New transit labels are not required during make-before-break (MBB) in scenarios where the new LSP instance traverses the exact same path as the old LSP instance. This saves the ingress LER and the services that use the tunnel from needing to update the forwarding plane with new tunnel labels and so makes MBB events faster. Periodic MBB events are relatively common in networks that deploy the 'auto-bandwidth' feature on RSVP-TE LSPs to monitor bandwidth utilization and periodically adjust LSP bandwidth.

4. Mix and match labels: Both 'TE link labels' and regular labels can be used on transit hops for a single RSVP-TE tunnel (see Section 6). This allows backward compatibility with transit LSRs that provide regular labels in Resv messages.

No additional extensions to routing protocols are required in order to support key functionalities such as bandwidth admission control, LSP priorities, preemption and auto-bandwidth on this shared MPLS forwarding plane. This document also discusses how Fast Reroute [RFC4090] via facility backup link protection using regular bypass tunnels can be supported on this forwarding plane.

The signaling procedures and extensions discussed in this document do not apply to Point to Multipoint (P2MP) RSVP-TE Tunnels.

2. Terminology

The following terms are used in this document:

TE link label: An incoming label at an LSR that will be popped by the LSR with the packet being forwarded over a specific outgoing TE link to a neighbor.

Shared MPLS forwarding plane: An MPLS forwarding plane where every participating LSR uses TE link labels on every LSP.

Segment Routed RSVP-TE tunnel: An MPLS RSVP-TE tunnel that requests the use of a shared MPLS forwarding plane at every hop of the LSP. The corresponding LSPs are referred to as Segment Routed RSVP-TE LSPs.

Delegation hop: A transit hop of a Segment Routed RSVP-TE LSP that is selected to assist in the imposition of the label stack in scenarios where the ingress LER cannot impose the full label stack. There could be multiple delegation hops along the path of a Segment Routed RSVP-TE LSP.

Delegation label: A label assigned at the delegation hop to represent a set of labels that will be pushed at this hop.

3. Allocation of TE Link Labels

An LSR that participates in a shared MPLS forwarding plane MUST allocate a unique TE link label for each TE link. When an LSR encounters a TE link label at the top of the label stack it MUST pop the label and forward the packet over the TE link to the downstream neighbor on the RSVP-TE tunnel.

Multiple TE link labels MAY be allocated for the TE link to accommodate tunnels requesting protection.

Implementations that maintain per label bandwidth accounting at each hop must aggregate the reservations made for all the LSPs using the shared TE link label.

4. Segment Routed RSVP-TE Tunnel Setup

This section provides an example of how the RSVP-TE signaling procedure works to set up a tunnel utilizing a shared MPLS forwarding plane. The sample topology below is used to explain the example. Labels shown at each node are TE link labels that, when present at the top of the label stack, indicate that they should be popped and that the packet should be forwarded on the TE link to the neighbor.

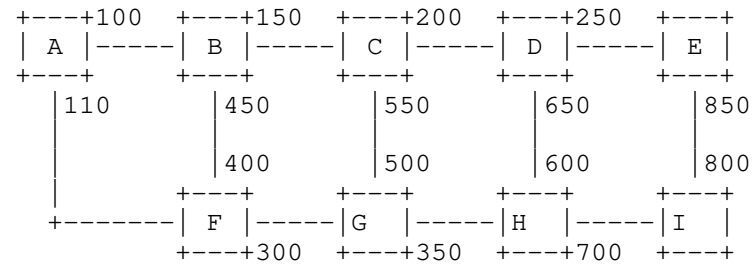


Figure 1: Sample Topology - TE Link Labels

Consider two tunnels:

RSVP-TE tunnel T1: From A to E on path A-B-C-D-E

RSVP-TE tunnel T2: From F to E on path F-B-C-D-E

Both tunnels share the TE links B-C, C-D, and D-E.

RSVP-TE is used to signal the setup of tunnel T1 (using the TE link label attributes flag defined in Section 9.2). When LSR D receives the Resv message from the egress LER E, it checks the next-hop TE link (D-E) and provides the TE link label (250) in the Resv message for the tunnel placing the label value in the Label object and also in the Label subobject carried in the RRO and setting the TE link label flag as defined in Section 9.3.

Similarly, LSR C provides the TE link label (200) for the TE link C-D, and LSR B provides the TE link label (150) for the TE link B-C.

For tunnel T2, the transit LSRs provide the same TE link labels as described for tunnel T1 as the links B-C, C-D, and D-E are common between the two LSPs.

The ingress LERs (A and F) will push the same stack of labels (from top of stack to bottom of stack) {150, 200, 250} for tunnels T1 and T2 respectively.

It should be noted that a transit LSR does not swap the top TE link label on an incoming packet (the label that it advertised in the Resv message it sent). All it has to do is pop the top label and forward the packet.

The values in the Label subobjects in the RRO are of interest to the ingress LERs in order to construct the stack of labels to impose on the packets.

If, in this example, there was another RSVP-TE tunnel T3 from F to I on path F-B-C-D-E-I, then this would also share the TE links B-C, C-D, and D-E and additionally traverse link E-I. The label stack used by F would be {150, 200, 250, 850}. Hence, regardless of the ingress and egress LERs from where the LSPs start and end, they will share LSR labels at shared hops in the shared MPLS forwarding plane.

There MAY be local operator policy at the ingress LER that influences the maximum depth of the label stack that can be pushed for a Segment Routed RSVP-TE tunnel. Prior to signaling the LSP, the ingress LER may determine that it would be unable to push a label stack containing one label for each hop along the path. In some scenarios, the ingress LER may not have sufficient information to make that determination. In these cases the LER SHOULD adopt the techniques described in Section 5.

5. Delegating Label Stack Imposition

One or more transit LSRs can assist the ingress LER by imposing part of the label stack required for the path. Consider the example in Figure 2 with an RSVP-TE tunnel from A to L on path A-B-C-D-E-F-G-H-I-J-K-L. In this case, the LSP is too long for LER A to impose the full label stack, so it uses the assistance of delegation hops LSR D and LSR I to impose parts of the label stack.

Each delegation hop allocates a delegation label to represent a set of labels that will be pushed at this hop. When a packet arrives at a delegation hop LSR with a delegation label, the LSR pops the label and pushes a set of labels before forwarding the packet.

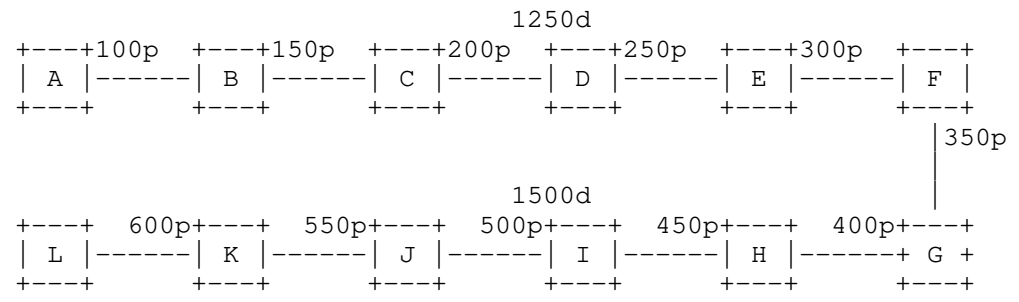


Figure 2: Delegating Label Stack Imposition

5.1. Stacking at the Ingress

When delegation labels come into play, there are two stacking approaches that the ingress can choose from. Section 7 explains how the label stack can be constructed.

5.1.1. Stack to Reach Delegation Hop

In this approach, the stack pushed by the ingress carries a set of labels that will take the packet to the first delegation hop. When this approach is employed, the set of labels represented by a delegation label at a given delegation hop will include the corresponding delegation label from the next delegation hop. As a result, this delegation label can only be shared among LSPs that are destined to the same egress and traverse the same downstream path.

This approach is shown in Figure 3. The delegation label 1250 represents the stack {300, 350, 400, 450, 1500} and the delegation label 1500 represents the label stack {550, 600}.

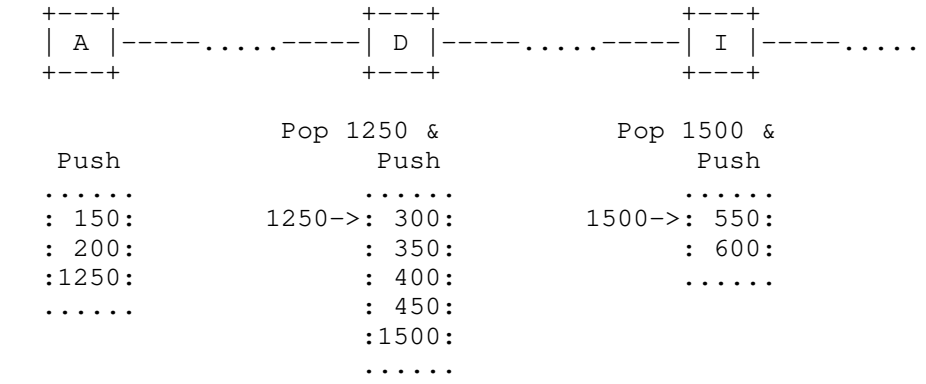


Figure 3: Stack to Reach Delegation Hop

With this approach, the ingress LER A will push {150, 200, 1250} for the tunnel in Figure 2. At LSR D, the delegation label 1250 will get popped and {300, 350, 400, 450, 1500} will get pushed. And at LSR I, the delegation label 1500 will get popped and the remaining set of labels {550, 600} will get pushed.

5.1.2. Stack to Reach Egress

In this approach, the stack pushed by the ingress carries a set of labels that will take the packet all the way to the egress so that all the delegation labels are part of the stack. When this approach is employed, the set of labels represented by a delegation label at a given delegation hop will not include the corresponding delegation label from the next delegation hop. As a result, this delegation label can be shared among all LSPs traversing the segment between the two delegation hops.

The downside of this approach is that the number of hops that the LSP can traverse is dictated by the label stack push limit of the ingress.

This approach is shown in Figure 4. The delegation label 1250 represents the stack {300, 350, 400, 450} and the delegation label 1500 represents the label stack {550, 600}.

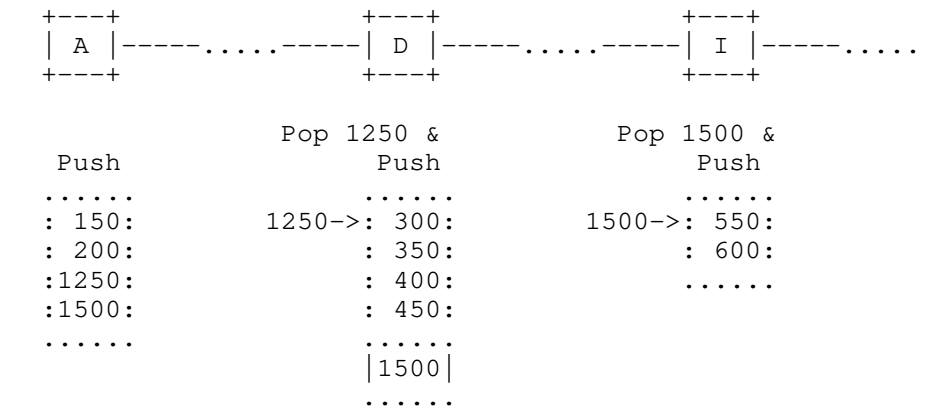


Figure 4: Stack to reach egress

With this approach, the ingress LER A will push {150, 200, 1250, 1500} for the tunnel in Figure 2. At LSR D, the delegation label 1250 will get popped and {300, 350, 400, 450} will get pushed. And at LSR I, the delegation label 1500 will get popped and the remaining set of labels {550, 600} will get pushed. The signaling extension required for the ingress to indicate the chosen stacking approach is defined in Section 9.6.

5.2. Explicit Delegation

In this delegation option, the ingress LER can explicitly delegate one or more specific transit LSRs to handle pushing labels for a certain number of their downstream hops. In order to accurately pick the delegation hops, the ingress needs to be aware of the label stack depth push limit (total number of MPLS labels that can be imposed, including all service/transport/special labels) of each of the transit LSRs prior to initiating the signaling sequence. The mechanism by which the ingress or controller (hosting the path computation element) learns this information is outside the scope of this document. An example of such a mechanism is specified in [RFC8491] (BMI-MSD advertisement).

The signaling extension required for the ingress LER to explicitly delegate one or more specific transit hops is defined in Section 9.4. The extension required for the delegation hop to indicate that the recorded label is a delegation label is defined in Section 9.5.

5.3. Automatic Delegation

In this approach, the ingress LER lets the downstream LSRs automatically pick suitable delegation hops during the initial signaling sequence. The ingress does not need to be aware up front of the label stack depth push limit of each of the transit LSRs. This approach SHOULD be used if there are loose hops [RFC3209] in the explicit route. The delegation hops are picked based on a per-hop signaled attribute called the Effective Transport Label-Stack Depth (ETLD) as described in the next section.

5.3.1. Effective Transport Label-Stack Depth (ETLD)

The ETLD is signaled as a per-hop recorded attribute in the Path message [RFC7570]. When automatic delegation is requested, the ingress MUST populate the ETLD with the maximum number of transport labels that it can potentially send to its downstream hop. This value is then decremented at each successive hop. If a node is reached and it is determined that this hop cannot support automatic delegation, then it MUST NOT use TE link labels and use regular labels instead. If a node is reached where the ETLD set from the previous hop is 1, then that node MUST select itself as the delegation hop. If a node is reached and it is determined that this hop cannot receive more than one transport label, then that node MUST select itself as the delegation hop. If there is a node or a sequence of nodes along the path of the LSP that do not support ETLD, then the immediate hop that supports ETLD MUST select itself as the delegation hop. The ETLD MUST be decremented at each non-delegation transit hop by either 1 or some appropriate number based on local

policy. For example, consider a transit node with a local policy that mandates it to take the label stack read limit into account when decrementing the ETLD. With this policy, the ETLD is decremented in such a way that the transit hop does not receive any more labels in the stack than it can read. At each delegation hop, the ETLD MUST be reset to the maximum number of transport labels that the hop can send and the ETLD decrements start again at each successive hop until either a new delegation hop is selected or the egress is reached. As a result, by the time the Path message reaches the egress, all delegation hops are selected. During the Resv processing, at each delegation hop, a suitable delegation label is selected (either an existing label is reused or a new label is allocated) and recorded in the Resv message.

Consider the example shown in Figure 5. Let's assume ingress LER A can push up to 3 transport labels while the remaining nodes can push up to 5 transport labels. The ingress LER A signals the initial Path message with ETLD set to 3. The ETLD value is adjusted at each successive hop and signaled downstream as shown. By the time the Path message reaches the egress LER L, LSRs D and I are automatically selected as delegation hops.

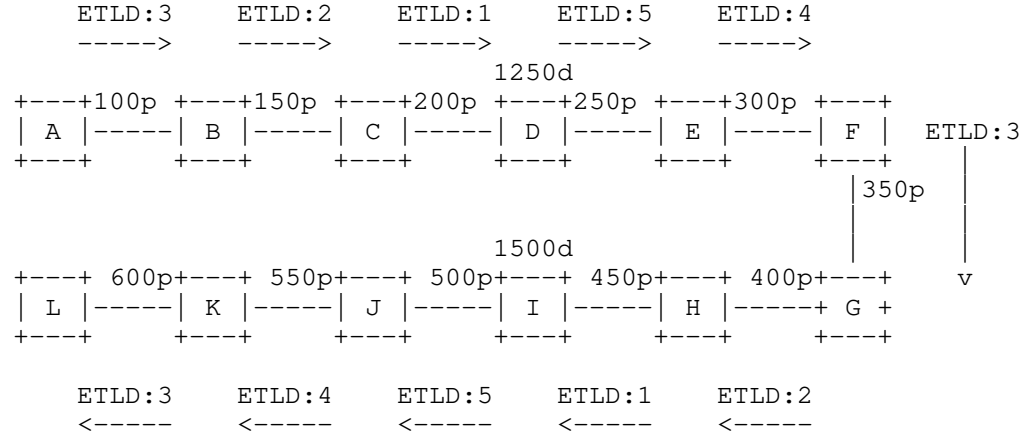


Figure 5: ETLD

When an LSP that requests automatic delegation also requests facility backup protection [RFC4090], the ingress or the delegation hop MUST account for the bypass tunnel's label(s) when populating the ETLD. Hence, when a regular bypass tunnel is used to protect the facility, the ETLD that gets populated on these nodes is one less than what gets populated for a corresponding unprotected LSP.

Signaling extension for the ingress LER to request automatic delegation is defined in Section 9.4. The extension for signaling the ETLD is defined in Section 9.7. The extension required for the delegation hop to indicate that the recorded label is a delegation label is defined in Section 9.5.

6. Mixing TE Link Labels and Regular Labels in an RSVP-TE Tunnel

Labels can be mixed across transit hops in a single MPLS RSVP-TE LSP. Certain LSRs can use TE link labels and others can use regular labels. The ingress can construct a label stack appropriately based on what type of label is recorded from every transit LSR.

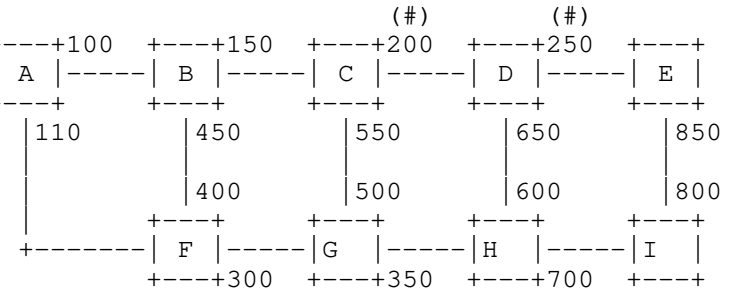


Figure 6: Sample Topology - TE Link Labels and Regular Labels

If the transit LSR allocates a regular label to be sent upstream in the Resv, then the label operation at the LSR is a swap to the label received from the downstream LSR. If the transit LSR is using a TE link label to be sent upstream in the Resv, then the label operation at the LSR is a pop and forward regardless of any label received from the downstream LSR. There is no change in the behavior of a penultimate hop popping (PHP) LSR [RFC3031].

Section 7 explains how the label stack can be constructed. For example, the LSP from A to I using path A-B-C-D-E-I will use a label stack of {150, 200}.

7. Construction of Label Stacks

The ingress LER or delegation hop MUST check the type of label received from each transit hop as recorded in the RRO in the Resv

message and generate the appropriate label stack to reach the next delegation hop or the egress.

The following logic is used by the node constructing the label stack:

> Each RRO label sub-object MUST be processed starting with the label sub-object from the first downstream hop. Any label provided by the first downstream hop MUST always be pushed on the label stack regardless of the label type. If the label type is a TE link label, then any label from the next downstream hop MUST also be pushed on the constructed label stack. If the label type is a regular label, then any label from the next downstream hop MUST NOT be pushed on the constructed label stack. If the label type is a delegation label, then the type of stacking approach chosen by the ingress for this LSP (Section 5.1) MUST be used to determine how the delegation labels are pushed in the label stack.

8. Facility Backup Protection

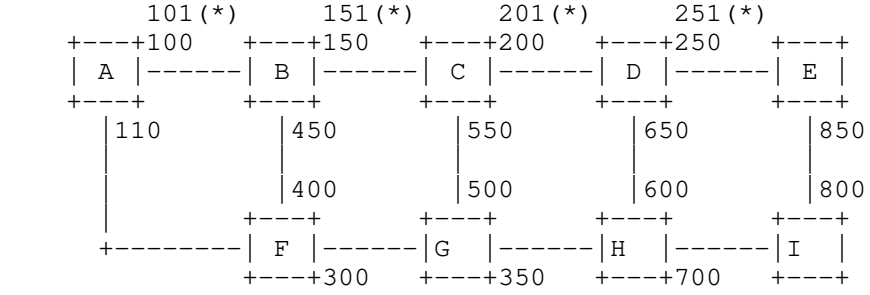
The following section describes how link protection works with facility backup protection [RFC4090] using regular bypass tunnels for the Segment Routed RSVP-TE tunnels. The procedures for supporting node protection are not discussed in this document. The use of Segment Routed bypass tunnels for providing facility protection is left for further study.

8.1. Link Protection

To provide link protection at a Point of Local Repair (PLR) with a shared MPLS forwarding plane, the LSR MUST allocate a separate TE link label for the TE link that will be used for RSVP-TE tunnels that request link protection from the ingress. No signaling extensions are required to support link protection for RSVP-TE tunnels over the shared MPLS forwarding plane.

At each LSR, link protected TE link labels can be allocated for each TE link and a link protecting facility backup LSP can be created to protect the TE link. The link protected TE link label can be sent by the LSR for LSPs requesting link protection over the specific TE link. Since the facility backup terminates at the next-hop (merge point), the incoming label on the packet will be what the merge point expects.

Consider the network shown in Figure 7. LSR B can install a facility backup LSP for the link protected TE link label 151. When the TE link B-C is up, LSR B will pop 151 and send the packet to C. If the TE link B-C is down, the LSR can pop 151 and send the packet via the facility backup to C.



Notation : (*) denotes link protection TE link labels

Figure 7: Link Protection Topology

9. Protocol Extensions

9.1. Requirements

The functionality discussed in this document imposes the following requirements on the signaling protocol.

- The Ingress of the LSP needs to have the ability to mandate/ request the use and recording of TE link labels at all hops along the path of the LSP.
- When the use of TE link labels is mandated/requested for the path:
  - the node recording the TE link label needs to have the ability to indicate if the recorded label is a TE link label.
  - the ingress needs to have the ability to delegate label stack imposition by:
    - explicitly mandating specific hops to be delegation hops (or)
    - requesting automatic delegation.
  - When explicit delegation is mandated or automatic delegation is requested:
    - the ingress needs to have the ability to indicate the chosen stacking approach (and)
    - the delegation hop needs to have the ability to indicate that the recorded label is a delegation label.

9.2. Attribute Flags TLV: TE Link Label

Bit Number 16 (Early allocation by IANA): TE Link Label

The presence of this in the LSP_ATTRIBUTES/LSP_REQUIRED_ATTRIBUTES object [RFC5420] of a Path message indicates that the ingress has requested/mandated the use and recording of TE link labels at all hops along the path of this LSP. When a node that recognizes this flag but does not cater to the mandate because of local policy receives a Path message carrying the LSP_REQUIRED_ATTRIBUTES object with this flag set, it MUST send a PathErr message with an error code of 'Routing Problem (24)' and an error value of 'TE link label usage failure (TBD3)'. A transit hop that caters to this request/mandate MUST also check for the presence of other Attribute Flags introduced in this document (Section 9.4 and Section 9.6) and process them as specified. An ingress LER that sets this bit MUST also set the "label recording desired" flag [RFC3209] in the SESSION_ATTRIBUTE object.

9.3. RRO Label Subobject Flag: TE Link Label

Bit Number (TBD1): TE Link Label

The presence of this flag indicates that the recorded label is a TE link label. This flag MUST be used by a node only if the use and recording of TE link labels is requested/mandated for the LSP.

9.4. Attribute Flags TLV: LSI-D

Bit Number 17 (Early allocation by IANA): Label Stack Imposition - Delegation (LSI-D)

Automatic Delegation: The presence of this flag in the LSP_ATTRIBUTES object of a Path message indicates that the ingress has requested automatic delegation of label stack imposition. This flag MUST be set in the LSP_ATTRIBUTES object of a Path message only if the use and recording of TE link labels is requested/mandated for this LSP. If the transit hop does not support this flag, it MUST NOT use TE link labels and use regular labels instead. If the use of TE link labels was mandated in the LSP_REQUIRED_ATTRIBUTES object, it MUST send a PathErr message with an error code of 'Routing Problem (24)' and an error value of 'TE link label usage failure (TBD3)'.

Explicit Delegation: The presence of this flag in the HOP_ATTRIBUTES subobject [RFC7570] of an Explicit Route Object (ERO) in the Path message indicates that the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject has been picked as an

explicit delegation hop. The HOP_ATTRIBUTES subobject carrying this flag MUST have the R (Required) bit set. This flag MUST be set in the HOP_ATTRIBUTES subobject of an ERO object in the Path message only if the use and recording of TE link labels is requested/mandated for this LSP. If the hop recognizes this flag but is not able to comply with this mandate because of local policy, it MUST send a PathErr message with an error code of 'Routing Problem (24)' and an error value of 'Label stack imposition failure (TBD4)'.

9.5. RRO Label Subobject Flag: Delegation Label

Bit Number (TBD2): Delegation Label

The presence of this flag indicates that the recorded label is a delegation label. This flag MUST be used by a node only if the use and recording of TE link labels and delegation are requested/mandated for the LSP.

9.6. Attributes Flags TLV: LSI-D-S2E

Bit Number 18 (Early allocation by IANA): Label Stack Imposition - Delegation - Stack to reach egress (LSI-D-S2E)

The presence of this flag in the LSP_ATTRIBUTES object of a Path message indicates that the ingress has chosen to use the "Stack to reach egress" approach for stacking. The absence of this flag in the LSP_ATTRIBUTES object of a Path message indicates that the ingress has chosen to use the "Stack to reach delegation hop" approach for stacking. This flag MUST be set in the LSP_ATTRIBUTES object of a Path message only if the use and recording of TE link labels and delegation are requested/mandated for this LSP. If the transit hop is not able to support the "Stack to reach egress" approach, it MUST send a PathErr message with an error code of 'Routing Problem (24)' and an error value of 'Label stack imposition failure (TBD4)'.

9.7. Attributes TLV: ETLD

The format of the ETLD Attributes TLV is shown in Figure 8. The Attribute TLV Type is 6 (Early allocation by IANA).

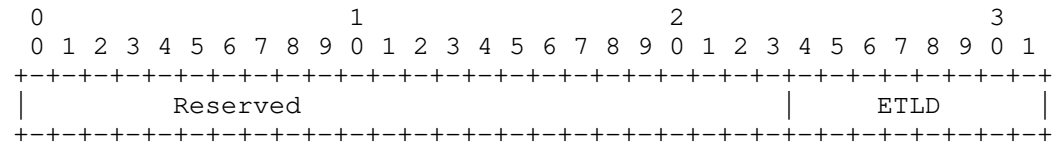


Figure 8: The ETLD Attributes TLV

The presence of this TLV in the HOP_ATTRIBUTES subobject of an RRO object in the Path message indicates that the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject supports automatic delegation. This attribute MUST be used only if the use and recording of TE link labels is requested/mandated and automatic delegation is requested for the LSP.

The ETLD field specifies the effective number of transport labels that this hop (in relation to its position in the path) can potentially send to its downstream hop. It MUST be set to a non-zero value.

The Reserved field is for future specification. It SHOULD be set to zero on transmission and MUST be ignored on receipt to ensure future compatibility.

10. OAM Considerations

MPLS LSP ping and traceroute [RFC8029] are applicable for Segment Routed RSVP-TE tunnels. The existing procedures allow for the label stack imposed at a delegation hop to be reported back in the Label Stack Sub-TLV in the MPLS echo reply for traceroute.

11. Acknowledgements

The authors would like to thank Adrian Farrel, Kireeti Kompella, Markus Jork and Ross Callon for their input from discussions.

Adrian Farrel provided a review and text suggestion for clarity and readability.

12. Contributors

The following individuals contributed to this document:

Raveendra Torvi
Juniper Networks
Email: rtorvi@juniper.net

Chandra Ramachandran
Juniper Networks
Email: csekar@juniper.net

George Swallow
Email: swallow.ietf@gmail.com

13. IANA Considerations

13.1. Attribute Flags: TE Link Label, LSI-D, LSI-D-S2E

IANA manages the 'Attribute Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at http://www.iana.org/assignments/rsvp-te-parameters. This document introduces three new Attribute Flags.

Bit No.	Name	Attribute FlagsPath	Attribute FlagsResv	RRO	ERO	Reference
16	TE Link Label	Yes	No	No	No	[This.ID] (Section 9.2)
17	LSI-D	Yes	No	No	Yes	[This.ID] (Section 9.4)
18	LSI-D-S2E	Yes	No	No	No	[This.ID] (Section 9.6)

Note: The code points specified for TE Link Label, LSI-D and LSI-D-S2E are early allocations by IANA.

13.2. Attribute TLV: ETLD

IANA manages the "Attribute TLV Space" registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at http://www.iana.org/assignments/rsvp-te-parameters. This document introduces a new Attribute TLV.

Type	Name	Allowed on LSP ATTRIBUTES	Allowed on LSP REQUIRED ATTRIBUTES	Allowed on LSP Hop Attributes	Reference
6	ETLD	No	No	Yes	[This.ID] (Section 9.7)

Note: The code point specified for ETLD is an early allocation by IANA.

13.3. Record Route Label Sub-object Flags: TE Link Label, Delegation Label

IANA manages the 'Record Route Object Sub-object Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at http://www.iana.org/assignments/rsvp-te-parameters. This registry currently does not include Label Sub-object Flags. This document requests the addition of a new sub-registry for Label Sub-object Flags as shown below.

Flag	Name	Reference
0x1	Global Label	RFC 3209
TBD1	TE Link Label	[This.ID] (Section 9.3)
TBD2	Delegation Label	[This.ID] (Section 9.5)

All assignments in this sub-registry are to be performed via Standards Action.

13.4. Error Codes and Error Values

IANA maintains a registry called "Resource Reservation Protocol (RSVP) Parameters" with a subregistry called "Error Codes and Globally-Defined Error Value Sub-Codes". Within this subregistry there is a definition of the "Routing Problem" error code with error code value 24. The definition lists a number of error values that may be used with this error code. IANA is requested to allocate further error values for use with this error code as described in this document. The resulting entry in the registry should look as follows.

24 Routing Problem [RFC3209]

This Error Code has the following globally-defined Error Value sub-codes:

TBD3 = TE link label usage failure [This.ID]
TBD4 = Label stack imposition failure [This.ID]

14. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

15. References

15.1. Normative References

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[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>.

[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>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangarps, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, DOI 10.17487/RFC5420, February 2009, <https://www.rfc-editor.org/info/rfc5420>.

[RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <https://www.rfc-editor.org/info/rfc7570>.

[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>.

15.2. Informative References

[I-D.ietf-spring-segment-routing-mpls] Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-18 (work in progress), December 2018.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <https://www.rfc-editor.org/info/rfc2961>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

[RFC8370] Beeram, V., Ed., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP-TE Deployments", RFC 8370, DOI 10.17487/RFC8370, May 2018, <https://www.rfc-editor.org/info/rfc8370>.

[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>.

[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

Harish Sitaraman
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: hsitaraman@juniper.net

Vishnu Pavan Beeram
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
US

Email: vbeeram@juniper.net

Tejal Parikh
Verizon
400 International Parkway
Richardson, TX 75081
US

Email: tejal.parikh@verizon.com

Tarek Saad
Cisco Systems
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: tsaad@cisco.com

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

A. Farrel
Old Dog Consulting
S. Bryant
Huawei
J. Drake
Juniper Networks

March 7, 2019

An MPLS-Based Forwarding Plane for Service Function Chaining
draft-ietf-mpls-sfc-07

Abstract

This document describes how Service Function Chaining (SFC) can be achieved in an MPLS network by means of a logical representation of the Network Service Header (NSH) in an MPLS label stack. That is, the NSH is not used, but the fields of the NSH are mapped to fields in the MPLS label stack. It does not deprecate or replace the NSH, but acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks.

Status of This Memo

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

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

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

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

Copyright Notice

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

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

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

Table of Contents

1. Introduction 2
2. Requirements Language 4
3. Choice of Data Plane SPI/SI Representation 4
4. Use Case Scenarios 4
4.1. Label Swapping for Logical NSH 5
4.2. Hierarchical Encapsulation 5
4.3. Fine Control of Service Function Instances 5
4.4. Micro Chains and Label Stacking 5
4.5. SFC and Segment Routing 6
5. Basic Unit of Representation 6
6. MPLS Label Swapping 7
7. MPLS Label Stacking 10
8. Mixed Mode Forwarding 11
9. A Note on Service Function Capabilities and SFC Proxies . . . 13
10. Control Plane Considerations 13
11. Use of the Entropy Label 14
12. Metadata 14
12.1. Indicating Metadata in User Data Packets 15
12.2. Inband Programming of Metadata 17
12.2.1. Loss of Inband Metadata 20
13. Worked Examples 21
14. Implementation Notes 24
15. Security Considerations 25
16. IANA Considerations 27
17. Acknowledgements 27
18. Contributors 28
19. References 28
19.1. Normative References 28
19.2. Informative References 29
Authors' Addresses 30

1. Introduction

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract service functions before being delivered to the intended destination. An architecture for SFC is defined in [RFC7665].

When applying a particular Service Function Chain to the traffic selected by a service classifier, the traffic needs to be steered

through an ordered set of Service Functions (SFs) in the network. This ordered set of SFs is termed a Service Function Path (SFP), and the traffic is passed between Service Function Forwarders (SFFs) that are responsible for delivering the packets to the SFs and for forwarding them onward to the next SFF.

In order to steer the selected traffic between SFFs and to the correct SFs the service classifier needs to attach information to each packet. This information indicates the SFP on which the packet is being forwarded and hence the SFs to which it must be delivered. The information also indicates the progress the packet has already made along the SFP.

The Network Service Header (NSH) [RFC8300] has been defined to carry the necessary information for Service Function Chaining in packets. The NSH can be inserted into packets and contains various information including a Service Path Indicator (SPI), a Service Index (SI), and a Time To Live (TTL) counter.

Multiprotocol Label Switching (MPLS) [RFC3031] is a widely deployed forwarding technology that uses labels placed in a packet in a label stack to identify the forwarding actions to be taken at each hop through a network. Actions may include swapping or popping the labels as well, as using the labels to determine the next hop for forwarding the packet. Labels may also be used to establish the context under which the packet is forwarded. In many cases, MPLS will be used as a tunneling technology to carry packets through networks between SFFs.

This document describes how Service Function Chaining can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack. This approach is applicable to all forms of MPLS forwarding (where labels are looked up at each hop, and swapped or popped [RFC3031]). It does not deprecate or replace the NSH, but acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks. The mechanisms described in this document are a compromise between the full function that can be achieved using the NSH, and the benefits of reusing the existing MPLS forwarding paradigms (the approach defined here does not include the O-bit defined in [RFC8300] and has some limitations to the use of metadata as described in Section 12.

Section 4 provides a short overview of several use case scenarios that help to explain the relationship between the MPLS label operations (swapping, popping, stacking) and the MPLS encoding of the logical NSH described in this document.

It is assumed that the reader is fully familiar with the terms and concepts introduced in [RFC7665] and [RFC8300].

Note that one of the features of the SFC architecture described in [RFC7665] is the "SFC proxy" that exists to include legacy SFs that are not able to process NSH-encapsulated packets. This issue is equally applicable to the use of MPLS-encapsulated packets that encode a logical representation of an NSH. It is discussed further in Section 9.

2. Requirements Language

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

3. Choice of Data Plane SPI/SI Representation

While [RFC8300] defines the NSH that can be used in a number of environments, this document provides a mechanism to handle situations in which the NSH is not ubiquitously deployed. In this case it is possible to use an alternative data plane representation of the SPI/SI by carrying the identical semantics in MPLS labels.

In order to correctly select the mechanism by which SFC information is encoded and carried between SFFs, it may be necessary to configure the capabilities and choices either within the whole Service Function Overlay Network, or on a hop by hop basis. It is a requirement that both ends of a tunnel over the underlay network (i.e., a pair of SFFs adjacent in the SFC) know that the tunnel is used for SFC and know what form of NSH representation is used. A control plane signalling approach to achieve these objectives is provided using BGP in [I-D.ietf-bess-nsh-bgp-control-plane].

Note that the encoding of the SFC information is independent of the choice of tunneling technology used between SFFs. Thus, an MPLS representation of the logical NSH (as defined in this document) may be used even if the tunnel between a pair of SFFs is not an MPLS tunnel. Conversely, MPLS tunnels may be used to carry other encodings of the logical NSH (specifically, the NSH itself).

4. Use Case Scenarios

There are five scenarios that can be considered for the use of an MPLS encoding in support of SFC. These are set out in the following sub-sections.

4.1. Label Swapping for Logical NSH

The primary use case for SFC is described in [RFC7665] and delivered using the NSH which, as described in [RFC8300], uses an encapsulation with a position indicator that is modified at each SFC hop along the chain to indicate the next hop.

The label swapping use case scenario effectively replaces the NSH with an MPLS encapsulation as described in Section 6. The MPLS labels encode the same information as the NSH to form a logical NSH. The labels are modified (swapped per [RFC3031]) at each SFC hop along the chain to indicate the next hop. The processing and forwarding state for a chain (i.e., the actions to take on a received label) are programmed in to the network using a control plane or management plane.

4.2. Hierarchical Encapsulation

[RFC8459] describes an architecture for hierarchical encapsulation using the NSH. It facilitates partitioning of SFC domains for administrative reasons, and allows concatenation of service function chains under the control of a service classifier.

The same function can be achieved in an MPLS network using an MPLS encoding of the logical NSH, and label stacking as defined in [RFC3031] and described in Section 7. In this model, swapping is used per Section 4.1 to navigate one chain, and when the end of the chain is reached, the final label is popped revealing the label for another chain. Thus, the primary mode is swapping, but stacking is used to enable the ingress classifier to control concatenation of service function chains.

4.3. Fine Control of Service Function Instances

It may be that a service function chain (as described in Section 4.1) allows some leeway in the choice of service function instances along the chain. However, it may be that a service classifier wishes to constrain the choice and this can be achieved using chain concatenation so that the first chain ends at the point of choice, the next label in the stack indicates the specific service function instance to be executed, and the next label in the stack starts a new chain. Thus, a mixture of label swapping and stacking is used.

4.4. Micro Chains and Label Stacking

The scenario in Section 4.2 may be extended to its logical extreme by making each concatenated chain as short as it can be: one service function. Each label in the stack indicates the next service

function to be executed, and the network is programmed through the control plane or management plane to know how to route to the next (i.e., first) hop in each chain just as it would be to support the scenarios in Section 4.1 and Section 4.2.

This scenario is functionally identical to the use of MPLS-SR for SFC as described Section 4.5, and the discussion in that section applies to this section as well.

4.5. SFC and Segment Routing

Segment Routing (SR) in an MPLS network (known as MPLS-SR) uses a stack of MPLS labels to encode information about the path and network functions that a packet should traverse. MPLS-SR is achieved by applying control plane and management plane techniques to program the MPLS forwarding plane, and by imposing labels on packets at the entrance to the MPLS-SR network. An implementation proposal for achieving SFC using MPLS-SR can be found in [I-D.xuclad-spring-sr-service-programming] and is not discussed further in this document.

5. Basic Unit of Representation

When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used. This unit comprises two MPLS labels as shown below. The unit may be present one or more times in the label stack as explained in subsequent sections.

In order to convey the same information as is present in the NSH, two MPLS label stack entries are used. One carries a label to provide context within the SFC scope (the SFC Context Label), and the other carries a label to show which service function is to be actioned (the SF Label). This two-label unit is shown in Figure 1.

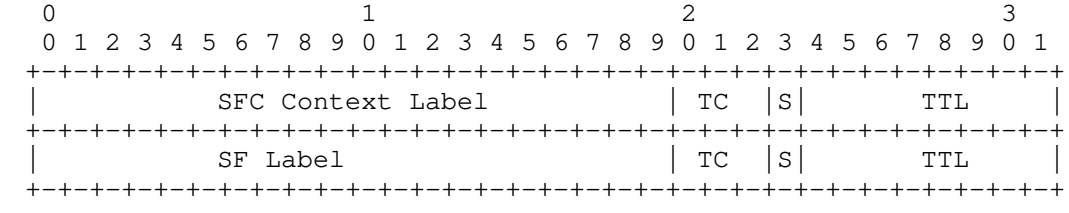


Figure 1: The Basic Unit of MPLS Label Stack for SFC

The fields of these two label stack entries are encoded as follows:

Label: The Label fields contain the values of the SFC Context Label and the SF Label encoded as 20 bit integers. The precise semantics of these label fields are dependent on whether the label stack entries are used for MPLS label swapping (see Section 6) or MPLS label stacking (see Section 7).

TC: The TC bits have no meaning in this case. They SHOULD be set to zero in both label stack entries when a packet is sent and MUST be ignored on receipt.

S: The bottom of stack bit has its usual meaning in MPLS. It MUST be clear in the SFC Context label stack entry. In the SF label stack entry it MUST be clear in all cases except when the label is the bottom of stack, when it MUST be set.

TTL: The TTL field in the SFC Context label stack entry SHOULD be set to 1. The TTL in SF label stack entry (called the SF TTL) is set according to its use for MPLS label swapping (see Section 6) or MPLS label stacking (see Section 7 and is used to mitigate packet loops.

The sections that follow show how this basic unit of MPLS label stack may be used for SFC in the MPLS label swapping case and in the MPLS label stacking. For simplicity, these sections do not describe the use of metadata: that is covered separately in Section 12.

6. MPLS Label Swapping

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 5 is used when MPLS label swapping is in use. The use case scenario for this approach is introduced in Section 4.1.

As can be seen from Figure 2, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used (i.e., MPLS, MPLS in UDP, MPLS in GRE, and MPLS in VXLAN or GPE), thus the tunnel technology does not need to be MPLS, but that is shown here for simplicity.

An entropy label ([RFC6790]) may also be present as described in Section 11.

Under these labels (or other encapsulation) comes a single instance of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 5 the following meanings are applied:

SPI Label: The Label field of the SFC Context label stack entry contains the value of the SPI encoded as a 20 bit integer. The semantics of the SPI is exactly as defined in [RFC8300]. Note that an SPI as defined by [RFC8300] can be encoded in 3 octets (i.e., 24 bits), but that the Label field allows for only 20 bits and reserves the values 0 though 15 as 'special purpose' labels [RFC7274]. Thus, a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16.

SI Label: The Label field of the SF label stack entry contains the value of the SI exactly as defined in [RFC8300]. Since the SI requires only 8 bits, and to avoid overlap with the 'special purpose' label range of 0 through 15 [RFC7274], the SI is carried in the top (most significant) 8 bits of the Label field with the low order 12 bits set to zero.

TC: The TC fields are as described in Section 5.

S: The S bits are as described in Section 5.

TTL: The TTL field in the SPI label stack entry SHOULD be set to 1 as stated in Section 5. The TTL in SF label stack entry is decremented once for each forwarding hop in the SFP, i.e., for each SFF transited, and so mirrors the TTL field in the NSH.

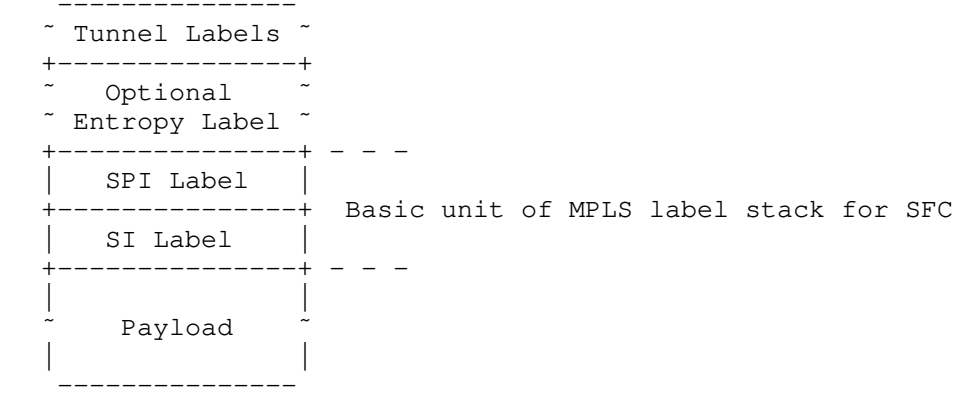


Figure 2: The MPLS SFC Label Stack

The following processing rules apply to the Label fields:

- When a classifier inserts a packet onto an SFP it sets the SPI Label to indicate the identity of the SFP, and sets the SI Label to indicate the first SF in the path.

- When a component of the SFC system processes a packet it uses the SPI Label to identify the SFP and the SI Label to determine which SFF or instance of an SF (an SFI) to deliver the packet to. Under normal circumstances (with the exception of branching and re-classification - see [I-D.ietf-bess-nsh-bgp-control-plane]) the SPI Label value is preserved on all packets. The SI Label value is modified by SFFs and through re-classification to indicate the next hop along the SFP.

The following processing rules apply to the TTL field of the SF label stack entry, and are derived from section 2.2 of [RFC8300]:

- When a classifier places a packet onto an SFP it MUST set the TTL to a value between 1 and 255. It SHOULD set this according to the expected length of the SFP (i.e., the number of SFs on the SFP), but it MAY set it to a larger value according to local configuration. The maximum TTL value supported in an NSH is 63, and so the practical limit here may also be 63.

- When an SFF receives a packet from any component of the SFC system (classifier, SFI, or another SFF) it MUST discard any packets with TTL set to zero. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- An SFF MUST decrement the TTL by one each time it performs a lookup to forward a packet to the next SFF.

- If an SFF decrements the TTL to zero it MUST NOT send the packet, and MUST discard the packet. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- SFIs MUST ignore the TTL, but MUST mirror it back to the SFF unmodified along with the SI (which may have been changed by local re-classification).

- If a classifier along the SFP makes any change to the intended path of the packet including for looping, jumping, or branching (see [I-D.ietf-bess-nsh-bgp-control-plane]) it MUST NOT change the SI TTL of the packet. In particular, each component of the SFC system MUST NOT increase the SI TTL value otherwise loops may go undetected.

7. MPLS Label Stacking

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 5 is used when MPLS label stacking is used to carry information about the SFP and SFs to be executed. The use case scenarios for this approach is introduced in Section 4.

As can be seen in Figure 3, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used.

An entropy label ([RFC6790]) may also be present as described in Section 11.

Under these labels comes one of more instances of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 5 the following meanings are applied:

SFC Context Label: The Label field of the SFC Context label stack entry contains a label that delivers SFC context. This label may be used to indicate the SPI encoded as a 20 bit integer using the semantics of the SPI is exactly as defined in [RFC8300] and noting that in this case a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16. This label may also be used to convey other SFC context-specific semantics such as indicating how to interpret the SF Label or how to forward the packet to the node that offers the SF if so configured and coordinated with the controller that programs the labels for the SFP.

SF Label: The Label field of the SF label stack entry contains a value that identifies the next SFI to be actioned for the packet. This label may be scoped globally or within the context of the preceding SFC Context Label and comes from the range 16 ... 2^20 - 1.

TC: The TC fields are as described in Section 5.

S: The S bits are as described in Section 5.

TTL: The TTL fields in the SFC Context label stack entry and in the SF label stack entry SHOULD be set to 1 as stated in Section 5, but MAY be set to larger values if the label indicated a forwarding operation towards the node that hosts the SF.

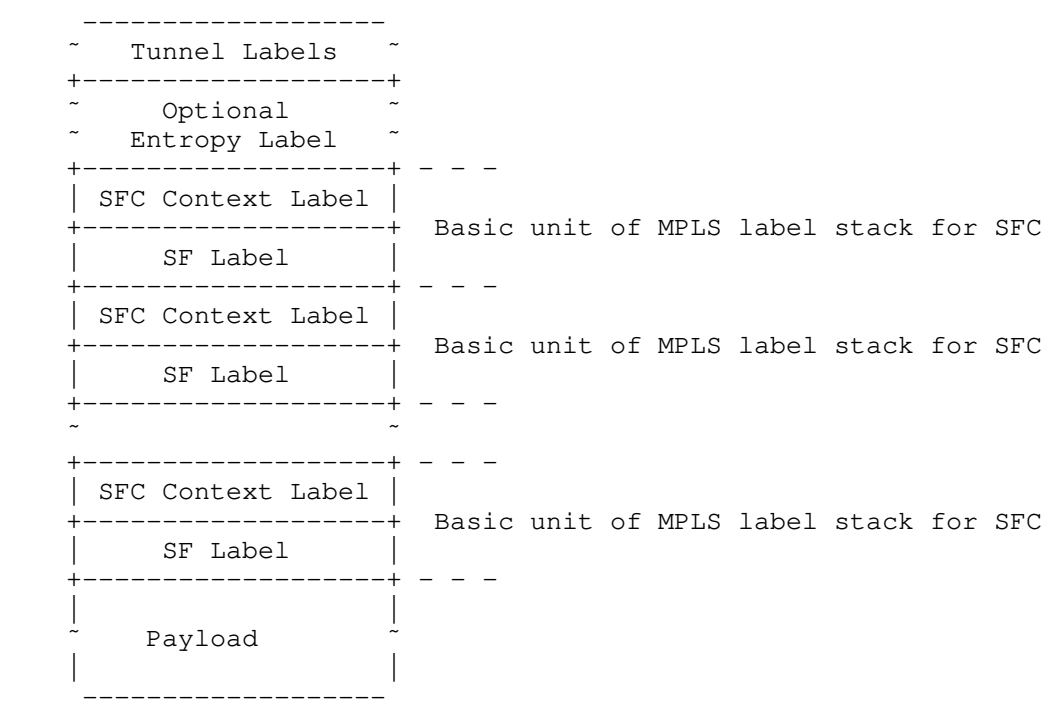


Figure 3: The MPLS SFC Label Stack for Label Stacking

The following processing rules apply to the Label fields:

- When a classifier inserts a packet onto an SFP it adds a stack comprising one or more instances of the basic unit of MPLS label stack for SFC. Taken together, this stack defines the SFs to be actioned and so defines the SFP that the packet will traverse.

- When a component of the SFC system processes a packet it uses the top basic unit of label stack for SFC to determine to which SFI to next deliver the packet. When an SFF receives a packet it examines the top basic unit of MPLS label stack for SFC to determine where to send the packet next. If the next recipient is a local SFI, the SFC strips the basic unit of MPLS label stack for SFC before forwarding the packet.

8. Mixed Mode Forwarding

The previous sections describe homogeneous networks where SFC forwarding is either all label swapping or all label popping

(stacking). This simplification helps to clarify the explanation of the mechanisms.

However, as described in Section 4.2, some uses cases may use label swapping and stacking at the same time. Furthermore, it is also possible that different parts of the network utilize swapping or popping such that an end-to-end service chain has to utilize a combination of both techniques. It is also worth noting that a classifier may be content to use an SFP as installed in the network by a control plane or management plane and so would use label swapping, but that there may be a point in the SFP where a choice of SFIs can be made (perhaps for load balancing) and where, in this instance, the classifier wishes to exert control over that choice by use of a specific entry on the label stack as described in Section 4.3.

When an SFF receives a packet containing an MPLS label stack, it checks from the context of the incoming interface, and from the SFP indicated by the top label whether it is processing an {SPI, SI} label pair for label swapping or a {context label, SFI index} label pair for label stacking. It then selects the appropriate SFI to which to send the packet. When it receives the packet back from the SFI, it has four cases to consider.

- If the current hop requires an {SPI, SI} and the next hop requires an {SPI, SI}, it sets the SPI label according to the SFP to be traversed, selects an instance of the SF to be executed at the next hop, sets the SI label to the SI value of the next hop, and tunnels the packet to the SFF for that SFI.

- If the current hop requires an {SPI, SI} and the next hop requires a {context label, SFI label}, it pops the {SPI, SI} from the top of the MPLS label stack and tunnels the packet to the SFF indicated by the context label.

- If the current hop requires a {context label, SFI label}, it pops the {context label, SFI label} from the top of the MPLS label stack.

  * If the new top of the MPLS label stack contains an {SPI, SI} label pair, it selects an SFI to use at the next hop, and tunnels the packet to SFF for that SFI.

  * If the new top of the MPLS label stack contains a {context label, SFI label}, it tunnels the packet to the SFF indicated by the context label.

9. A Note on Service Function Capabilities and SFC Proxies

The concept of an "SFC proxy" is introduced in [RFC7665]. An SFC proxy is logically located between an SFF and an SFI that is not "SFC-aware". Such SFIs are not capable of handling the SFC encapsulation (whether that be NSH or MPLS) and need the encapsulation stripped from the packets they are to process. In many cases, legacy SFIs that were once deployed as "bumps in the wire" fit into this category until they have been upgraded to be SFC-aware.

The job of an SFC proxy is to remove and then reimpose SFC encapsulation so that the SFF is able to process as though it was communication with an SFC-aware SFI, and so that the SFI is unaware of the SFC encapsulation. In this regard, the job of an SFC proxy is no different when NSH encapsulation is used and when MPLS encapsulation is used as described in this document, although (of course) it is different encapsulation bytes that must be removed and reimposed.

It should be noted that the SFC proxy is a logical function. It could be implemented as a separate physical component on the path from the SFF to SFI, but it could be co-resident with the SFF or it could be a component of the SFI. This is purely an implementation choice.

Note also that the delivery of metadata (see Section 12) requires specific processing if an SFC proxy is in use. This is also no different when NSH or the MPLS encoding defined in this document is in use, and how it is handled will depend on how (or if) each non-SFC-aware SFI can receive metadata.

10. Control Plane Considerations

In order that a packet may be forwarded along an SFP several functional elements must be executed.

- Discovery/advertisement of SFIs.
- Computation of SFP.
- Programming of classifiers.
- Advertisement of forwarding instructions.

Various approaches may be taken. These include a fully centralized model where SFFs report to a central controller the SFIs that they support, the central controller computes the SFP and programs the classifiers, and (if the label swapping approach is taken) the

central controller installs forwarding state in the SFFs that lie on the SFP.

Alternatively, a dynamic control plane may be used such as that described in [I-D.ietf-bess-nsh-bgp-control-plane]. In this case the SFFs use the control plane to advertise the SFIs that they support, a central controller computes the SFP and programs the classifiers, and (if the label swapping approach is taken) the central controller uses the control plane to advertise the SFPs so that SFFs that lie on the SFP can install the necessary forwarding state.

11. Use of the Entropy Label

Entropy is used in ECMP situations to ensure that packets from the same flow travel down the same path, thus avoiding jitter or re-ordering issues within a flow.

Entropy is often determined by hashing on specific fields in a packet header such as the "five-tuple" in the IP and transport headers. However, when an MPLS label stack is present, the depth of the stack could be too large for some processors to correctly determine the entropy hash. This problem is addressed by the inclusion of an Entropy Label as described in [RFC6790].

When entropy is desired for packets as they are carried in MPLS tunnels over the underlay network, it is RECOMMENDED that an Entropy Label is included in the label stack immediately after the tunnel labels and before the SFC labels as shown in Figure 2 and Figure 3.

If an Entropy Label is present in an MPLS payload, it is RECOMMENDED that the initial classifier use that value in an Entropy Label inserted in the label stack when the packet is forwarded (on the first tunnel) to the first SFF. In this case it is not necessary to remove the Entropy Label from the payload.

12. Metadata

Metadata is defined in [RFC7665] as providing "the ability to exchange context information between classifiers and SFs, and among SFs." [RFC8300] defines how this context information can be directly encoded in fields that form part of the NSH encapsulation.

The next two sections describe how metadata is associated with user data packets, and how metadata may be exchanged between SFC nodes in the network, when using an MPLS encoding of the logical representation of the NSH.

It should be noted that the MPLS encoding is less functional than the direct use of the NSH. Both methods support metadata that is "per-SFP" or "per-packet-flow" (see [RFC8393] for definitions of these terms), but "per-packet" metadata (where the metadata must be carried on each packet because it differs from one packet to the next even on the same flow or SFP) is only supported using the NSH and not using the mechanisms defined in this document.

12.1. Indicating Metadata in User Data Packets

Metadata is achieved in the MPLS realization of the logical NSH by the use of an SFC Metadata Label which uses the Extended Special Purpose Label construct [RFC7274]. Thus, three label stack entries are present as shown in Figure 4:

- The Extension Label (value 15)
- An extended special purpose label called the Metadata Label Indicator (MLI) (value TBD1 by IANA)
- The Metadata Label (ML).

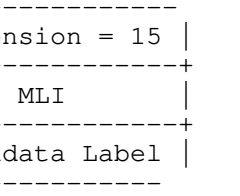


Figure 4: The MPLS SFC Metadata Label

The Metadata Label value is an index into a table of metadata that is programmed into the network using in-band or out-of-band mechanisms. Out-of-band mechanisms potentially include management plane and control plane solutions (such as [I-D.ietf-bess-nsh-bgp-control-plane]), but are out of scope for this document. The in-band mechanism is described in Section 12.2

The SFC Metadata Label (as a set of three labels as indicated in Figure 4) may be present zero, one, or more times in an MPLS SFC packet. For MPLS label swapping, the SFC Metadata Labels are placed immediately after the basic unit of MPLS label stack for SFC as shown in Figure 5. For MPLS label stacking, the SFC Metadata Labels are placed at the bottom of the label stack as shown in Figure 6.

Figure 5: The MPLS SFC Label Stack for Label Swapping with Metadata Label

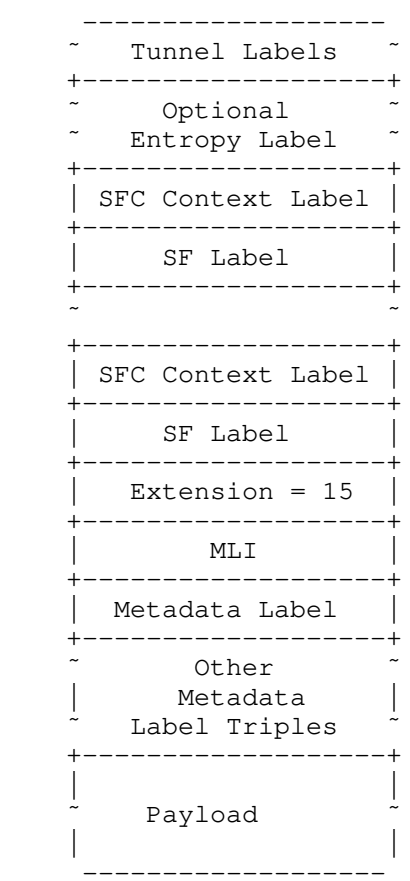


Figure 6: The MPLS SFC Label Stack for Label Stacking with Metadata Label

12.2. Inband Programming of Metadata

A mechanism for sending metadata associated with an SFP without a payload packet is described in [RFC8393]. The same approach can be used in an MPLS network where the NSH is logically represented by an MPLS label stack.

The packet header is formed exactly as previously described in this document so that the packet will follow the SFP through the SFC network. However, instead of payload data, metadata is included after the bottom of the MPLS label stack. An Extended Special Purpose Label is used to indicate that the metadata is present. Thus, three label stack entries are present:

- The Extension Label (value 15)

- An extended special purpose label called the Metadata Present Indicator (MPI) (value TBD2 by IANA)

- The Metadata Label (ML) that is associated with this metadata on this SFP and can be used to indicate the use of the metadata as described in Section 12.

The SFC Metadata Present Label, if present, is placed immediately after the last basic unit of MPLS label stack for SFC. The resultant label stacks are shown in Figure 7 for the MPLS label swapping case and Figure 8 for the MPLS label stacking case.

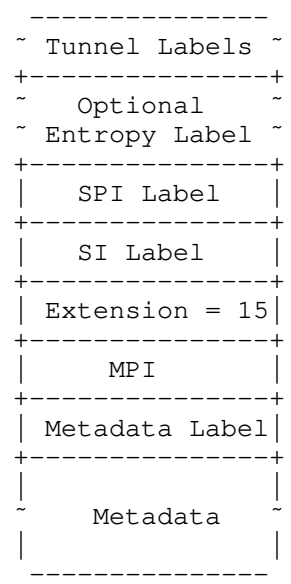


Figure 7: The MPLS SFC Label Stack for Label Swapping Carrying Metadata

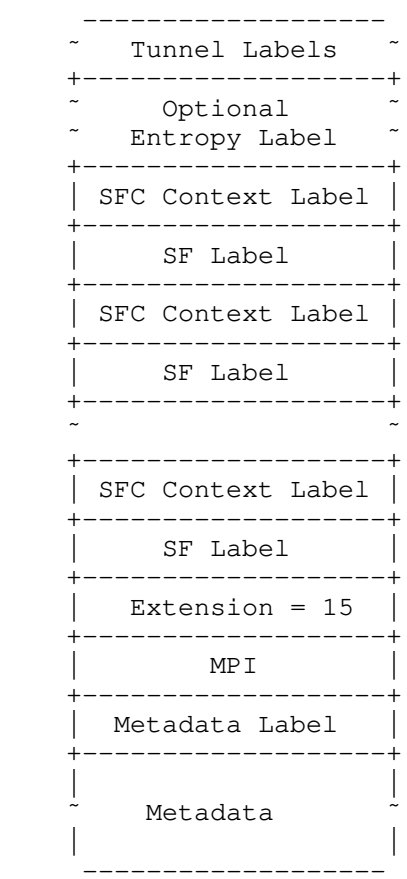


Figure 8: The MPLS SFC Label Stack for Label Stacking Carrying Metadata

In both cases the metadata is formatted as a TLV as shown in Figure 9.

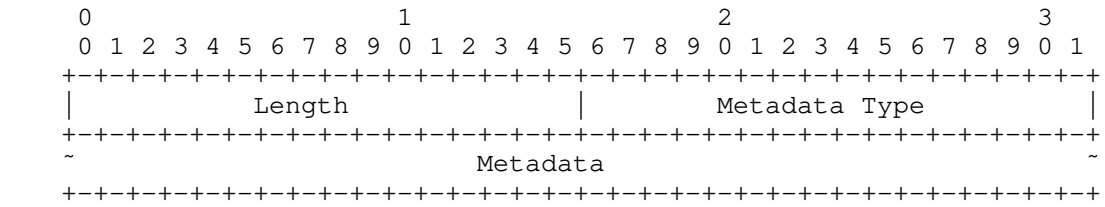


Figure 9: The Metadata TLV

The fields of this TLV are interpreted as follows:

Length: The length of the metadata carried in the Metadata field in octets not including any padding.

Metadata Type: The type of the metadata present. Values for this field are taken from the "MD Types" registry maintained by IANA and defined in [RFC8300] and encoded with the most significant bit first.

Metadata: The actual metadata formatted as described in whatever document defines the metadata. This field is end-padded with zero to three octets of zeroes to take it up to a four octet boundary.

12.2.1. Loss of Inband Metadata

Note that inband exchange of metadata is vulnerable to packet loss. This is both a risk arising from network faults and an attack vulnerability.

If packets that arrive at an SFF use an MLI that does not have an entry in the metadata table, an alarm can be raised and the packet can be discarded or processed without the metadata according to local configuration. This provides some long-term mitigation, but is not an ideal solution.

Further mitigation to loss of metadata packets can be achieved by retransmitting them at a configurable interval. This is a relatively cheap, but only partial solution because there may still be a window during which the metadata has not been received.

The concern of lost metadata may be particularly important when the metadata applicable to a specific MPI is being changed. This could result in out-of-date metadata being applied to a packet. If this is a concern, it is RECOMMENDED that a new MPI is used to install a new entry in the metadata table, and the packets in the flow should be marked with the equivalent new MLI.

Finally, if an application that requires metadata is sensitive to this potential loss or attack, it SHOULD NOT use inband metadata distribution, but SHOULD rely on control plane or management plane mechanisms because these approaches can use a more sophisticated protocol that includes confirmation of delivery, and can perform verification or inspection of entries in the metadata table.

13. Worked Examples

This section reverts to the simplified descriptions of networks that rely wholly on label swapping or label stacking. As described in Section 4, actual deployment scenarios may depend on the use of both mechanisms and utilize a mixed mode as described in Section 8.

Consider the simplistic MPLS SFC overlay network shown in Figure 10. A packet is classified for an SFP that will see it pass through two Service Functions, SFa and SFb, that are accessed through Service Function Forwarders SFFa and SFFb respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. The forwarding details of the SFP are distributed (perhaps using the mechanisms of [I-D.ietf-bess-nsh-bgp-control-plane]) so that the SFFs are programmed with the necessary forwarding instructions.

The packet progresses as follows:

a. The classifier assigns the packet to the SFP and imposes two label stack entries comprising a single basic unit of MPLS SFC representation:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 255.

   Further labels may be imposed to tunnel the packet from the classifier to SFFa.

b. When the packet arrives at SFFa it strips any labels associated with the tunnel that runs from the classifier to SFFa. SFFa examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFa. The packet is forwarded to SFa unmodified.

c. SFa performs its designated function and returns the packet to SFFa.

d. SFFa modifies the SI in the lower label stack entry (to 254) and uses the SPI/SI to look up the forwarding instructions. It sends the packet with two label stack entries:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 254.

   Further labels may be imposed to tunnel the packet from the SFFa to SFFb.

e. When the packet arrives at SFFb it strips any labels associated with the tunnel from SFFa. SFFb examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFb. The packet is forwarded to SFb unmodified.

f. SFb performs its designated function and returns the packet to SFFb.

g. SFFb modifies the SI in the lower label stack entry (to 253) and uses the SPI/SI to lookup up the forwarding instructions. It determines that it is the last SFF in the SFP so it strips the two SFC label stack entries and forwards the payload toward D using the payload protocol.

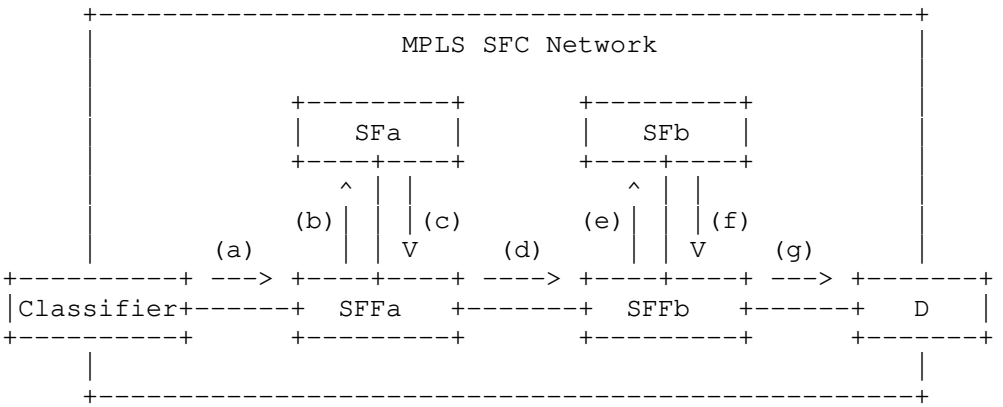


Figure 10: Service Function Chaining in an MPLS Network

Alternatively, consider the MPLS SFC overlay network shown in Figure 11. A packet is classified for an SFP that will see it pass

through two Service Functions, SFx and SFy, that are accessed through Service Function Forwarders SFFx and SFFy respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. However, the forwarding state for the SFP is not distributed and installed in the network. Instead it will be attached to the individual packets using the MPLS label stack.

The packet progresses as follows:

1. The classifier assigns the packet to the SFP and imposes two basic units of MPLS SFC representation to describe the full SFP:

   * The top basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SFx.

   * The lower basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SFy.

   Further labels may be imposed to tunnel the packet from the classifier to SFFx.

2. When the packet arrives at SFFx it strips any labels associated with the tunnel from the classifier. SFFx examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFx. The packet is forwarded to SFx unmodified.

3. SFx performs its designated function and returns the packet to SFFx.

4. SFFx strips the top basic unit of MPLS SFC representation revealing the next basic unit. It then uses the revealed context/SF values to determine how to route the packet to the

next SFF, SFFy. It sends the packet with just one basic unit of MPLS SFC representation comprising two label stack entries:

* The higher label stack entry contains a label carrying the SFC context.

* The lower label stack entry contains a label carrying the SF indicator for SFy.

Further labels may be imposed to tunnel the packet from the SFFx to SFFy.

5. When the packet arrives at SFFy it strips any labels associated with the tunnel from SFFx. SFFy examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFy. The packet is forwarded to SFy unmodified.

6. SFy performs its designated function and returns the packet to SFFy.

7. SFFy strips the top basic unit of MPLS SFC representation revealing the payload packet. It forwards the payload toward D using the payload protocol.

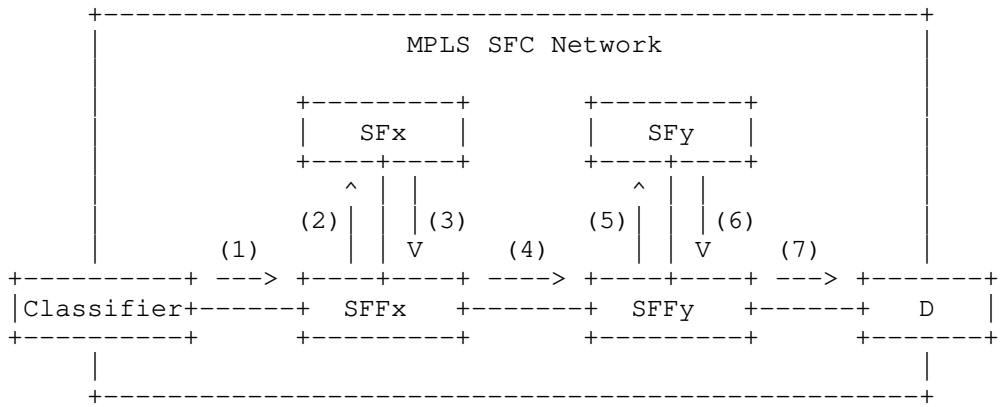


Figure 11: Service Function Chaining Using MPLS Label Stacking

14. Implementation Notes

It is not the job of an IETF specification to describe the internals of an implementation except where that directly impacts upon the bits on the wire that change the likelihood of interoperability, or where

the availability of configuration or security options directly affect the utility of an implementation.

However, in view of the objective of this document to acknowledge that there may be a need for an interim deployment of SFC functionality in brownfield MPLS networks, this section provides some observations about how an SFF might utilize MPLS features that are available in existing routers. This section is not intended to be definitive or technically complete, but is indicative.

Consider the mechanism used to indicate to which Virtual Routing and Forwarding (VRF) an incoming MPLS packet should be routed in a Layer 3 Virtual Private Network (L3VPN) [RFC4364]. In this case, the top MPLS label is an indicator of the VRF that is to be used to route the payload.

A similar approach can be taken with the label swapping SFC technique described in Section 6 such that the SFC Context Label identifies a routing table specific to the SFP. The SF Label can be looked up in the context of this routing table to determine to which SF to direct the packet, and how to forward it to the next SFF.

Advanced features (such as metadata) are not inspected by SFFs. The packets are passed to SFIs that are MPLS-SFC-aware or to SFC proxies, and those components are responsible for handling all metadata issues.

Of course, an actual implementation might make considerable optimizations on this approach, but this section should provide hints about how MPLS-based SFC might be achieved with relatively small modifications to deployed MPLS devices.

15. Security Considerations

Discussion of the security properties of SFC networks can be found in [RFC7665]. Further security discussion for the NSH and its use is present in [RFC8300]. Those documents provide analysis and present a set of requirements and recommendations for security and the normative security requirements from those documents apply to this specification. However, it should be noted that those documents do not describe any mechanisms for securing NSH systems.

It is fundamental to the SFC design that the classifier is a fully trusted element. That is, the classification decision process is not visible to the other elements and its output is treated as accurate. As such, the classifier has responsibility for determining the processing that the packet will be subject to, including, for example, firewall functions. It is also fundamental to the MPLS

design that packets are routed through the network using the path specified by the node imposing the labels, and that labels are swapped or popped correctly. Where an SF is not encapsulation-aware the encapsulation may be stripped by an SFC proxy such that packet may exist as a native packet (perhaps IP) on the path between SFC proxy and SF, however this is an intrinsic part of the SFC design which needs to define how a packet is protected in that environment.

SFC components are configured and enabled through a management system or a control plane. This document does not make any assumptions about what mechanisms are used. Deployments should, however, be aware that vulnerabilities in the management plane or control plane of an SFC system imply vulnerabilities in the whole SFC system. Thus, control plane solutions (such as [I-D.ietf-bess-nsh-bgp-control-plane]) and management plane mechanisms must include security measures that can be enable by operators to protect their SFC systems.

An analysis of the security of MPLS systems is provided in [RFC5920]. That document notes the MPLS forwarding plane has no built-in security mechanisms. Some proposals to add encryption to the MPLS forwarding plane have been suggested ([I-D.ietf-mpls-opportunistic-encrypt]), but no mechanisms have been agreed at the time of publication of this document. Additionally, MPLS does not provide any cryptographic integrity protection on the MPLS headers. That means that procedures described in this document rely on three basic principles:

- The MPLS network is often considered to be a closed network such that insertion, modification, or inspection of packets by an outside party is not possible. MPLS networks are operated with closed boundaries so that MPLS encapsulated packets are not admitted to the network, and MPLS headers are stripped before packets are forwarded from the network. This is particularly pertinent in the SFC context because [RFC7665] notes that "The architecture described herein is assumed to be applicable to a single network administrative domain." Furthermore, [RFC8300] states that packets originating outside the SFC-enabled domain MUST be dropped if they contain an NSH and packets exiting the SFC-enabled domain MUST be dropped if they contain an NSH. These constraints apply equally to the use of MPLS to encode a logical representation of the NSH.

- The underlying transport mechanisms (such as Ethernet) between adjacent MPLS nodes may offer security mechanisms that can be used to defend packets "on the wire".

- The SFC-capable devices participating in an SFC system are responsible for verifying and protecting payload packets and their contents as well as providing other security capabilities that might be required in the particular system.

Additionally, where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how the tunnel is secured.

Thus, this design relies on the component underlying technologies to address the potential security vulnerabilities, and documents the necessary protections (or risk of their absence) above. It does not include any native security mechanisms in-band with the MPLS encoding of the NSH functionality.

Note that configuration elements of this system (such as the programming of the table of metadata, see Section 12) must also be adequately secured although such mechanisms are not in scope for this protocol specification.

No known new security vulnerabilities over the SFC architecture [RFC7665] and the NSH specification [RFC8300] are introduced by this design, but if issues are discovered in the future it is expected that they will be addressed through modifications to control/ management components of any solution, or through changes to the underlying technology.

16. IANA Considerations

This document requests IANA to make allocations from the "Extended Special-Purpose MPLS Label Values" subregistry of the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry as follows:

Value	Description	
TBD1	Metadata Label Indicator (MLI)	[This.I-D]
TBD2	Metadata Present Indicator (MPI)	[This.I-D]

17. Acknowledgements

This document derives ideas and text from [I-D.ietf-bess-nsh-bgp-control-plane].

The authors are grateful to all those who contributed to the discussions that led to this work: Loa Andersson, Andrew G. Malis, Alexander Vainshtein, Joel M. Halpern, Tony Przygienda, Stuart

Mackie, Keyur Patel, and Jim Guichard. Loa Andersson provided helpful review comments.

Thanks to Loa Andersson, Lizhong Jin, Matthew Bocci, Joel Halpern, and Mach Chen for reviews of this text. Thanks to Russ Mundy for his Security Directorate review and to S Moonesamy for useful discussions. Thanks also to Benjamin Kaduk, Alissa Cooper, Eric Rescorla, Mirja Kuehlewind, Alvaro Retana, and Martin Vigoureux for comprehensive reviews during IESG evaluation.

The authors would like to be able to thank the authors of [I-D.xuclad-spring-sr-service-programming] and [RFC8402] whose original work on service chaining and the identification of services using SIDs, and conversation with whom helped clarify the application of MPLS-SR to SFC.

Particular thanks to Loa Andersson for conversations and advice about working group process.

18. Contributors

The following people contributed text to this document:

Andrew Malis
Email: agmalis@gmail.com

19. References

19.1. 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>.

[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>.

[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>.

[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>.

[RFC8393] Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", RFC 8393, DOI 10.17487/RFC8393, May 2018, <https://www.rfc-editor.org/info/rfc8393>.

19.2. Informative References

[I-D.ietf-bess-nsh-bgp-control-plane] Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-09 (work in progress), March 2019.

[I-D.ietf-mpls-opportunistic-encrypt] Farrel, A. and S. Farrell, "Opportunistic Security in MPLS Networks", draft-ietf-mpls-opportunistic-encrypt-03 (work in progress), March 2017.

[I-D.xuclad-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-xuclad-spring-sr-service-programming-01 (work in progress), October 2018.

[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>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

[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>.

[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>.

[RFC8459] Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", RFC 8459, DOI 10.17487/RFC8459, September 2018, <https://www.rfc-editor.org/info/rfc8459>.

Authors' Addresses

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

John Drake
Juniper Networks

Email: jdrake@juniper.net

Network Working Group
Internet-Draft
Updates: 8287 (if approved)
Intended status: Standards Track
Expires: December 30, 2018

F. Iqbal, Ed.
N. Kumar
Z. Ali
C. Pignataro
Cisco

June 28, 2018

Supporting Flexible Algorithm Prefix SIDs in LSP Ping/Traceroute draft-iqbal-spring-mpls-ping-algo-00

Abstract

RFC8287 defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifier (SIDs) with an MPLS data plane. [I-D.ietf-lsr-flex-algo] proposes a mechanism to allow IGPs to compute constraint based path over network and use Segment Routing Prefix-SIDs to steer packets along the constraint-based paths. All Prefix-SIDs associated with the Flexible Algorithm are assigned to the same IPv4/IPv6 Prefix. Any Segment Routing network that uses Flexible Algorithm based path computation needs additional details to be carried in the FEC Stack sub-TLV for FEC validaiton.

This document updates [RFC8287] by modifying IPv4 and IPv6 IGP-Prefix Segment ID FEC sub-TLVs to also include algorithm identification.

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 December 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 3
1.1. Conventions 3
2. Motivation 3
3. Algorithm Identification for IGP-Prefix SID Sub-TLVs 4
3.1. IPv4 IGP-Prefix Segment ID Sub-TLV 4
3.2. IPv6 IGP-Prefix Segment ID Sub-TLV 5
4. Procedures 5
4.1. Initiator Node Procedures 5
4.2. Responder Node Procedures 6
5. IANA Considerations 6
6. Security Considerations 6
7. Acknowledgements 6
8. Contributors 6
9. References 6
9.1. Normative References 6
9.2. Informative References 7
Authors' Addresses 7

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. Multiple Flexible Algorithms are assigned to the same IPv4/IPv6 Prefix while each utilizing a different MPLS Prefix SID label.

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 Flexible Algorithm Prefix-SID, the validating device also requires algorithm 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 Flexible Algorithm, while maintaining backwards compatibility with existing implementations of [RFC8287].

1.1. 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.

2. Motivation

In presence of Flexible 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, Flexible Algorithm based IGP-Prefix SID may take a different path than default IGP Prefix SID algorithm.

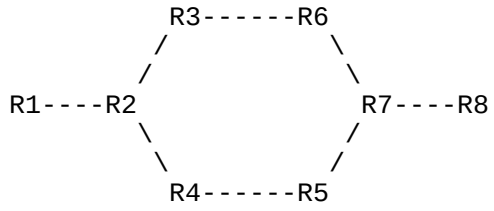


Figure above, which is a simplification of the diagram used in [RFC8287] illustrates this point through an example. Node Segment IDs for R1, R2, R3, R4, R5, R6, R7, and R8 for the default algorithm are 5001, 5002, 5003, 5004, 5005, 5006, 5007, and 5008, respectively. Nodes R1, R2, R4, R5, R7, and R8 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.

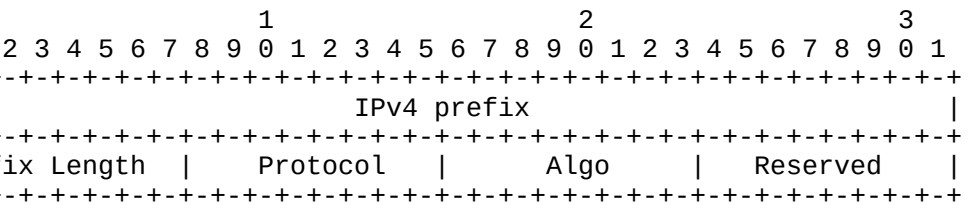
Carrying algorithm identification in the Target FEC Stack sub-TLV of MPLS echo request will help avoid such false negatives. It will also help detect forwarding deviations such as when the packet for a particular destination is incorrectly forwarded to a device that is participating in the default algo but does not participate in a given Flexible Algorithm.

3. 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.

3.1. IPv4 IGP-Prefix Segment ID Sub-TLV

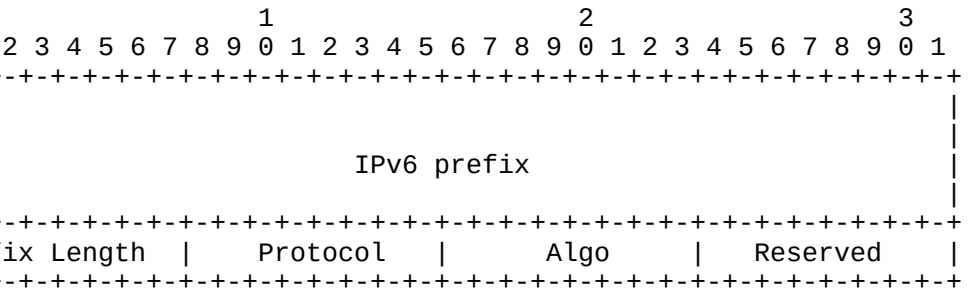
The Sub-TLV format for IPv4 IGP-Prefix Segment ID MUST be set as shown in the below TLV format:



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).

3.2. 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).

4. Procedures

4.1. 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.

4.2. 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.

5. IANA Considerations

This document does not introduce any IANA considerations.

6. Security Considerations

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

7. Acknowledgements

TBA.

8. Contributors

TBA

9. References

9.1. Normative References

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

[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>.

9.2. Informative References

[I-D.ietf-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., Gulko, A., "IGP Flexible Algorithm", <http://tools.ietf.org/html/ietf-lsr-flex-algo>.

Authors' Addresses

Faisal Iqbal (editor)
Cisco Systems, Inc.

Email: faiqbal@cisco.com

Nagendra Kumar
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

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

Z. Li
Z. Zhuang
Huawei Technologies

March 5, 2018

Service-Oriented MPLS Path Programming (MPP)
draft-li-mpls-path-programming-00

Abstract

Segment Routing has been proposed to cope with the use cases in traffic engineering, fast re-reroute, service chain, etc. It can leverage existing MPLS dataplane without any modification. In fact, the label stack capability in MPLS would have been utilized well to implement flexible path programming to satisfy all kinds of requirements of service bearing. But in the distributed environment, the flexible programming capability is difficult to implement and always confined to reachability. As the introducing of central control in the network, the flexible MPLS programming capability becomes possible owing to two factors: 1. It becomes easier to allocate label for more purposes than reachability; 2. It is easy to calculate the MPLS path in a global network view. Moreover, the MPLS path programming capability can be utilized to satisfy more requirements of service bearing in the service layer which is defined as service-oriented MPLS path programming. This document defines the concept of MPLS path programming, then proposes use cases, architecture and protocol extension requirements in the service layer.

Requirements Language

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

Status of This Memo

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

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

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

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

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction 3
2. Terminology 3
3. Programming Capability of MPLS Path 4
3.1. History Review 4
3.2. Gap Analysis of Segment Routing 5
4. Use Cases of Service-Oriented MPLS Path Programming 6
4.1. Use Cases for Unicast Service 6
4.1.1. Basic Reachability 6
4.1.2. VPN Identification 6
4.1.3. ECMP(Equal Cost Multi-Path) 6
4.1.4. Service OAM 6
4.1.5. Traffic Steering 7
4.2. Use Cases of Multicast Service 7
4.2.1. Basic Reachability 7
4.2.2. MVPN Identification 8
4.2.3. Source Identification 8
4.3. Use Cases of MPLS Virtual Network 8
4.4. Use cases of More Label Combinations 8
4.5. Use cases for Centralized Mapping of Service to Tunnels . 9
5. Framework of Service-Oriented MPLS Path Programming 9
5.1. Central Control for MPLS Path Programming 9
5.2. BGP-based MPLS Segment Distribution 11
5.3. MPLS Service Path Programming 11
5.3.1. Label Combination and Download of MPLS Path 11
5.3.2. Mapping of Service Path to Service Path 11
5.4. Compatibility 12

5.5. Protocol Extensions Requirements 12
5.5.1. BGP 12
5.5.2. I2RS 13
6. IANA Considerations 13
7. Security Considerations 13
8. References 13
8.1. Normative References 13
8.2. Informative References 13
Authors' Addresses 15

1. Introduction

Segment Routing has been proposed to cope with the use cases in traffic engineering, fast re-reroute, service chain, etc. It can leverage existing MPLS dataplane without any modification. In fact, the label stack capability in MPLS would have been utilized well to implement flexible path programming to satisfy all kinds of requirements of service bearing. But in the distributed environment, the flexible programming capability is difficult to implement and always confined to reachability. As the introducing of central control in the network, the flexible MPLS programming capability becomes possible owing to two factors: 1. It becomes easier to allocate label for more purposes than reachability; 2. It is easy to calculate the MPLS path in a global network view. Moreover, the MPLS path programming capability can be utilized to satisfy more requirements of service bearing in the service layer which is defined as service-oriented MPLS path programming. This document defines the concept of MPLS path programming, then proposes use cases, architecture and protocol extension requirements in the service layer.

2. Terminology

BGP: Border Gateway Protocol

BUM: Broadcast, Unknown unicast and Multicast

EVPN: Ethernet VPN

FRR: Fast Re-Route

L2VPN: Layer 2 VPN

L3VPN: Layer 3 VPN

MPP: MPLS Path Programming

MVPN: Multicast VPN

RR: Route Reflector

SDN: Software-Defined Network

SR-path: Segment Routing Path

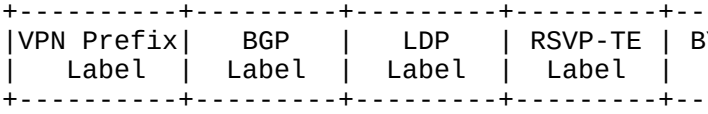
3. Programming Capability of MPLS Path

MPLS path is composed by label stacks. Since in the label stack the labels in different layers can represent different meaning and the depth of the label stack can be unlimited in theory, it is possible can make up all kinks of MPLS paths based on the combination of labels. If we look on the combination of MPLS labels as programming, it is can be seen that the MPLS path has high programming capability.

3.1. History Review

The solutions based on MPLS label stack have been widely deployed. For example, in the scenario of Options C inter-AS VPN ([RFC4364]), we assume that LDP over TE is used as the transport tunnel and the TE tunnel starts at the ingress PE, following label stack can be composed by the ingress PE for MPLS path to bear VPN service:

If facility FRR ([RFC4090]) is deployed for the MPLS TE tunnel, once the failure happens, additional label will be pushed for the label stack which is shown as follows:



The combination of labels in the above label stack is not simpler than the existing segment routing solution which composes the segment routing path through combination of segments. In fact, this is also a use case of source packet routing. But the combination is not as flexible as the segment routing since the combination of labels is always to cope with the reachability issue with limited capability in the distributed environment as follows:

1. Each label in the label stack is always binded with the reachability to a specific prefix. That is, the purpose of the label binding is limited.

2. It is difficult to implement flexible path calculation based on policy or constraints. For example, MPLS TE proposes rich set of traffic engineering attributes for transport. But it needs complex configurations in each ingress node in an unscalable way. That is, the path calculation and composition capability is limited.

As more concepts on MPLS label are proposed such as entropy label, source label, segment routing, etc., the purpose of label binding expands and the combination of labels can become more flexible. MPLS path programming capability becomes more realistic to satisfy more application scenarios.

3.2. Gap Analysis of Segment Routing

Segment Routing ([I-D.ietf-spring-segment-routing]) is a typical example of MPLS path programming. The segment based on MPLS label is to represent nodes or agencies in the network. Through the collected information of network segments and path calculation based on the service requirement in the central controller, there will be flexible segment routing paths for the usage of traffic engineering. The SR-path can be advertised to the ingress node through PCE extensions. ([I-D.ietf-pce-segment-routing]).

Segment routing can implement source packet routing with high flexibility. On the other hand, there are multiple layers for MPLS path to bear services which is shown in the following figure:

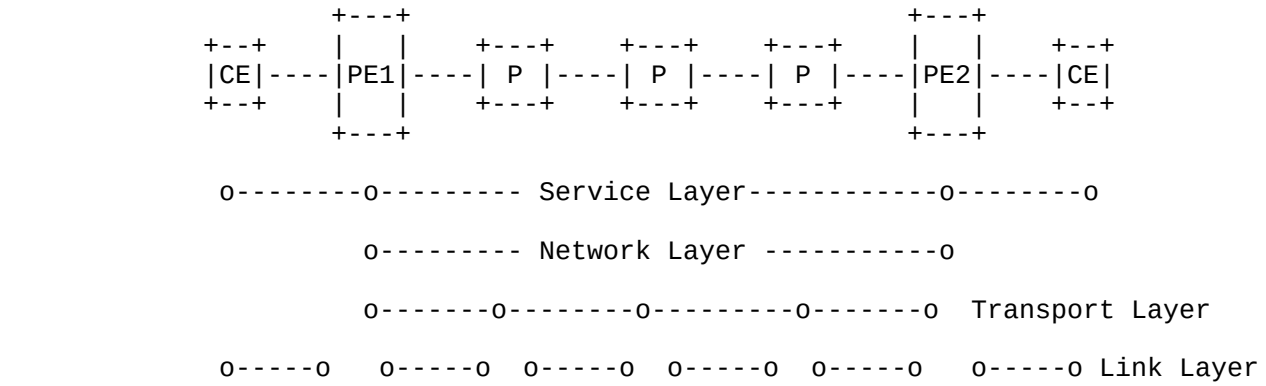


Figure 1: Multiple Layers of Service Bearing

Now the segment routing is to provide the source packet routing in the transport layer. We can call this type of source packet routing as Transport-Oriented MPLS path programming. There will be more

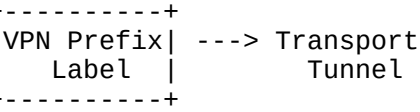
application scenarios which needs the source packet routing in the service layer and network layer. We call these types of source packet routing as Service-Oriented MPLS path programming.

4. Use Cases of Service-Oriented MPLS Path Programming

4.1. Use Cases for Unicast Service

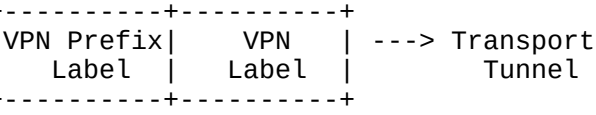
4.1.1. Basic Reachability

The basic reachability for VPN service is to allocate label to specific prefix including IP address or MAC address. MPLS path is as follows (using L3VPN as the example):



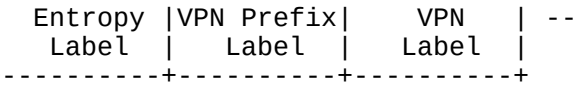
4.1.2. VPN Identification

There are several use cases which need to indentify the VPN the packet belongs to in the forwarding plane such as the egress PE node protection for VPN ([I-D.zhang-l3vpn-label-sharing]). MPLS Path can be as follows:



4.1.3. ECMP(Equal Cost Multi-Path)

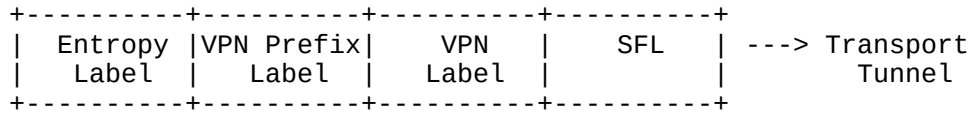
In order to satisfy ECMP to take full advantage of link bandwidth in the network, the entropy label ([RFC6790]) can be encapsulated. MPLS path can be as follows:



4.1.4. Service OAM

OAM is an important requirement for the service. The performance metrics should be measured against the Service Level Agreement(SLA) for the user. Now there are relatively complete and mature OAM mechanism for the point-to-point service. But for LDP LSP, owing to

the MP2P model it is difficult to identify the flow from a specific PE based on the label. Synonymous flow label (SFL) has been proposed as a possible solution([I-D.ietf-mpls-sfl-framework]). When the source label is applied, MPLS path can be as follows:



4.1.5. Traffic Steering

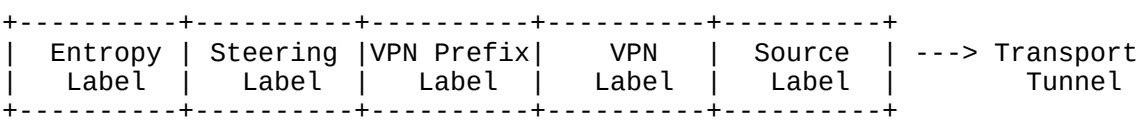
Service traffic may span multiple ASes. It is an important use case to steer traffic at ASBR in an AS to specific ASBR in neighboring AS. There are possible solutions for this type of traffic steering:

1. Traffic Steering based on Transport Tunnel

This method looks on the segment between two ASBRs as the extension of the transport tunnel in an AS. It can steer the traffic through the specific path to the neighboring AS.

2. Traffic Steering in Service/Network Layer

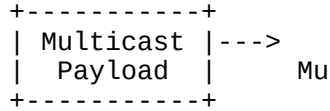
This method is to directly encapsulate the service flow with the steering label in the ingress PE before it enters into the transport tunnel. [I-D.ietf-spring-segment-routing-central-epe] illustrates the application of Segment Routing to solve the Egress Peer Engineering (EPE) requirement. When this method is applied, the MPLS path can be as follows:



4.2. Use Cases of Multicast Service

4.2.1. Basic Reachability

When MPLS multicast tunnel is applied for the multicast service in BGP-based MVPN, VPLS or EVPN, the basic MPLS path can be as follows:

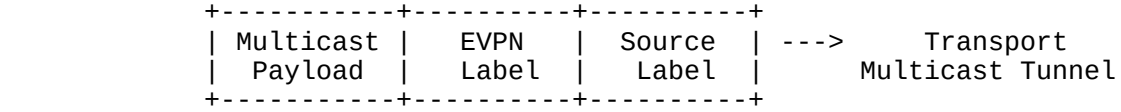


4.2.2. MVPN Identification

When multiple MVPNs shares the MPLS multicast tunnel, it is necessary to encapsulate the label to identify specific MVPN([RFC6514]). The MPLS path can be as follows:

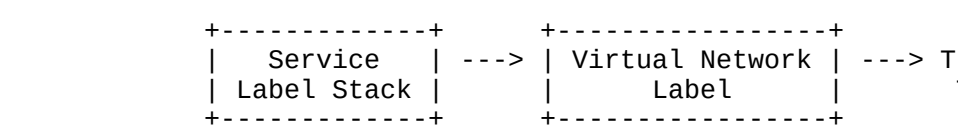
4.2.3. Source Identification

In order to implement the split horizon or C-MAC learning in the fowarding plane when MPLS multicast is to bear BUM traffic in L2VPN, it is necessary to introduce the label to identify the source of the BUM traffic([I-D.li-l2vpn-segment-evpn]). The MPLS path is as follows:



4.3. Use Cases of MPLS Virtual Network

The framework of MPLS virtual network has been proposed in [I-D.li-mpls-network-virtualization-framework]. When the unicast service or the multicast service enters into the transport tunnel, it may take different MPLS virtual network identified by the MPLS label for the purpose of QoS routing, security or virtual operations. The MPLS path is as follows:



4.4. Use cases of More Label Combinations

Service-oriented MPLS path programming can make full use of flexible combination of MPLS labels to satisfy different requirements for the service flow. Based on the above proposed use cases, MPLS path can be composed adopting part or whole labels for these use cases based on the service requirement. Besides this, more flexible MPLS label combination may be provided:

1. Hierarchical process or multiple repeated process: The label for the same usage can exist in different layers. Or the process identified by the label can exist in multiple nodes along the path. Then the labels for the same usage can be encapsulated several times in the label stack. The encapsulation can be as follows (using SERVICE LABEL to identify the label for the same service process in different layers):

2. Special-purpose label indicator: Since the label in the service-oriented MPLS programming is for special-purpose process, it may need a special purpose label to indicate the usage of the label followed the special-purpose labels. For example, the ELI(Entropy Label Indicator) is introduced for the entropy label. This may introduce more labels for the combination.

This document is not to define all possible use cases for the service-oriented path programming. The new use cases can be defined in the future independent document.

4.5. Use cases for Centralized Mapping of Service to Tunnels

In the transport layers, there can be multiple tunnels to one specific destination which satisfy different constraints. In the traditional way, the tunnel is set up by the distributed forwarding nodes. As the PCE-initiated LSP setup [I-D.ietf-pce-pce-initiated-lsp]is introduced, the tunnel can be setup in the central controlled way. In order to satisfy the different service requirements, it is necessary to provide the capability to flexibly map the service to different tunnels. Since the central control point has enough information based on the whole network view, it can be an effective way to map the service to the tunnel by the central point and advertise the mapping information to the end-points of the service to guide the mapping in the forwarding node.

5. Framework of Service-Oriented MPLS Path Programming

5.1. Central Control for MPLS Path Programming

Central control plays an important role in MPLS path programming. It can extend the MPLS path programming capability easily. There are two important functionalities for the central control:

1. Central controlled MPLS label allocation: Label can be allocated centrally for special usage other than reachability. These labels can be used to compose MPLS path. We call it as MPLS Segment.

2. Central controlled MPLS path programming: Central controller can calculate path in a global network view and implement the MPLS path programming based on the collected information of MPLS segments to satisfy different requirements of services.

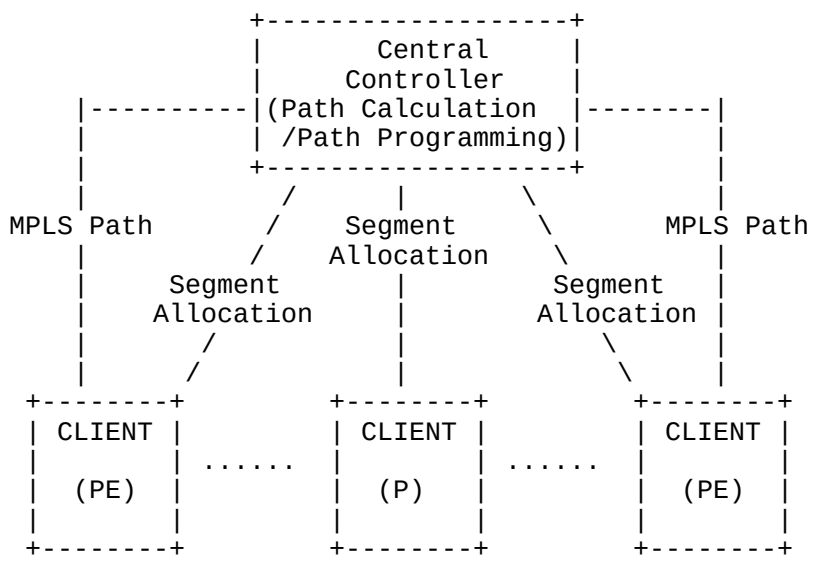


Figure 2 Central Control for MPLS Path Programming

There are two types of MPLS path: Transport-Oriented MPLS Path and Service-Oriented MPLS Path. For the transport-oriented MPLS path, segment routing is the typical solution: MPLS segment distribution is done by IGP extensions ([I-D.ietf-isis-segment-routing-extensions]) and [I-D.ietf-ospf-segment-routing-extensions]); the programmed MPLS path can be downloaded through PCEP extensions from PCE to PCC([I-D.ietf-pce-segment-routing]). For the service-oriented MPLS path programming, it not only includes composing the MPLS path in the service and network layer, but also includes determining the mapping of the service path to the transport path. Since the process corresponding to the label in the service label stack is always located at the PE nodes, BGP extensions can be introduced for service-oriented path programming.

5.2. BGP-based MPLS Segment Distribution

1. Label Allocation

There are two types of label used for MPLS segments:

1) Local Label: The service process is done locally. The label can be allocated by the local PE which provides the process.

2) Global Label: The service process is common in multiple PEs. This means the label has global meaning. The label allocation can be done by the central controller. The global label work can refer to [I-D.li-mpls-global-label-framework].

2. Label Mapping Distribution

BGP extensions can be used to distribution label mapping. Regarding to the above two types of label allocation, the process is as follows:

1) Local Label Mapping: BGP can directly distribute the label mapping from the local PE to peer PEs. The local PE can also only distribute the label mapping to central controller. Then the central controller re-distribute the label mapping to other PEs. In this method, the central controller plays the role of traditional RR.

2) Global Label Mapping: The label mapping for the service can be directly distributed by the central controller to multiple PEs. It can be done by BGP extensions.

5.3. MPLS Service Path Programming

5.3.1. Label Combination and Download of MPLS Path

According to the service requirements, the central controller can combine MPLS segments flexibly. Then it can download the service label combination for specific prefix related with the service. The BGP extensions can be reused to download the programmed MPLS path.

5.3.2. Mapping of Service Path to Service Path

Since the transport path is also to satisfy the service bearing the requirement, it can reuse the existing MPLS tunnel technology or it can also be programmed according to traffic engineering requirements of service. Then there needs to be implements the mapping of the service path to the transport path. There are two ways to implement the mapping:

1. BGP Extensions: Through the community attribute of BGP, the identifier of the transport path can be carrier when distribute label stack for a specific prefix.

2. I2RS Extensions: I2RS can be used to download route policy to the client node. Based on the policy, the client node can implement the required mapping.

5.4. Compatibility

When the MPLS path programming is done the central controller and downloaded through BGP extensions to the Client node, the path SHOULD has higher priority than the path calculated on the Client node’s own.

5.5. Protocol Extensions Requirements

5.5.1. BGP

REQ 01: BGP extensions SHOULD be introduced to distribute local label mapping for specific process to the central controller and other client nodes.

REQ 02: BGP extensions SHOULD be introduced to distribute global label mapping for specific process from the central controller to the client nodes .

REQ 03: BGP extensions SHOULD be introduced to download label stack for service-oriented MPLS path.

REQ 04: BGP extensions SHOULD be introduced to carry the identifier of the transport MPLS path with service MPLS path to implement the mapping.

REQ 05: BGP extensions SHOULD be introduced to specify the end-points to accept the prefix advertised by the central controller.

REQ 06: BGP extensions SHOULD be introduced to specify the priority for the prefix with attributes of MPLS path programming advertised by the central controller.

REQ 07: When route selection is done in the client node, the path advertised by the central controller SHOULD has higher priority than the path calculated on the client’s own.

5.5.2. I2RS

REQ 11: I2RS clients SHOULD provide interface to I2RS agent to download policy to implement the mapping of the service path to the transport path.

6. IANA Considerations

This document makes no request of IANA.

7. Security Considerations

TBD.

8. References

8.1. Normative References

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

8.2. Informative References

[I-D.ietf-isis-segment-routing-extensions] Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-15 (work in progress), December 2017.

[I-D.ietf-mpls-sfl-framework] Bryant, S., Chen, M., Li, Z., Swallow, G., Sivabalan, S., and G. Mirsky, "Synonymous Flow Label Framework", draft-ietf-mpls-sfl-framework-01 (work in progress), January 2018.

[I-D.ietf-ospf-segment-routing-extensions] Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-24 (work in progress), December 2017.

[I-D.ietf-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-11 (work in progress), October 2017.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-11 (work in progress), November 2017.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[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.

[I-D.li-l2vpn-segment-evpn]
Li, Z., Yong, L., and J. Zhang, "Segment-Based EVPN(S-EVPN)", draft-li-l2vpn-segment-evpn-01 (work in progress), February 2014.

[I-D.li-mpls-global-label-framework]
Li, Z., Zhao, Q., Chen, X., Yang, T., and R. Raszuk, "A Framework of MPLS Global Label", draft-li-mpls-global-label-framework-02 (work in progress), July 2014.

[I-D.li-mpls-global-label-usecases]
Li, Z., Zhao, Q., Yang, T., Raszuk, R., and L. Fang, "Usecases of MPLS Global Label", draft-li-mpls-global-label-usecases-03 (work in progress), October 2015.

[I-D.li-mpls-network-virtualization-framework]
Li, Z. and M. Li, "Framework of Network Virtualization Based on MPLS Global Label", draft-li-mpls-network-virtualization-framework-00 (work in progress), October 2013.

[I-D.zhang-l3vpn-label-sharing]
Zhang, M., Zhou, P., and R. White, "Label Sharing for Fast PE Protection", draft-zhang-l3vpn-label-sharing-02 (work in progress), June 2014.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[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>.

[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>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: zhuangshunwan@huawei.com

MPLS Working Group
Internet-Draft
Intended status: Informational
Expires: April 14, 2019

A. Malis
S. Bryant
Huawei Technologies
J. Halpern
Ericsson
W. Henderickx
Nokia

October 11, 2018

MPLS Encapsulation for SFC NSH
draft-malis-mpls-sfc-encapsulation-03

Abstract

This document describes how to use a Service Function Forwarder (SFF) Label (similar to a pseudowire label or VPN label) to indicate the presence of a Service Function Chaining (SFC) Network Service Header (NSH) between an MPLS label stack and the packet payload. This allows SFC packets using the NSH to be forwarded between SFFs over an MPLS network, and the selection between multiple SFFs in the destination MPLS node.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 14, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction 2
2. MPLS Encapsulation Using an SFF Label 3
2.1. MPLS Label Stack Construction at the Sending Node 3
2.2. SFF Label Processing at the Destination Node 4
3. Equal Cost Multipath (ECMP) Considerations 4
4. Operations, Administration, and Maintenance (OAM) Considerations 5
5. IANA Considerations 5
6. Security Considerations 5
7. Acknowledgements 5
8. References 5
8.1. Normative References 5
8.2. Informative References 6
Authors' Addresses 6

1. Introduction

As discussed in [RFC8300], a number of transport encapsulations for the Service Function Chaining (SFC) Network Service Header (NSH) already exist, such as Ethernet, GRE [RFC2784], and VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe].

This document describes an MPLS transport encapsulation for the NSH, and also describes how to use a Service Function Forwarder (SFF) [RFC7665] Label to indicate the presence of the NSH in the MPLS packet payload. This allows SFC packets using the NSH to be forwarded between SFFs in an MPLS transport network, where MPLS is used to interconnect the network nodes that contain one or more SFFs. The label is also used to select between multiple SFFs in the destination MPLS node.

SFF Labels are similar to other service labels at the bottom of an MPLS label stack that denote the contents of the MPLS payload being other than IP, such as a layer 2 pseudowire, an IP packet that is routed in a VPN context with a private address, or an Ethernet virtual private wire service.

This informational document follows well-established MPLS procedures and does not require any actions by IANA or any new protocol extensions.

2. MPLS Encapsulation Using an SFF Label

The encapsulation is a standard MPLS label stack [RFC3032] with an SFF Label at the bottom of the stack, followed by a NSH as defined by [RFC8300] and the NSH payload.

Much like a pseudowire label, an SFF Label is allocated by the downstream receiver of the NSH from its per-platform label space.

If a receiving node supports more than one SFF (i.e, more than one SFC forwarding instance), then the SFF Label can be used to select the proper SFF, by having the receiving node advertise more than one SFF Label to its upstream sending nodes as appropriate.

The method used by the downstream receiving node to advertise SFF Labels to the upstream sending node is out of scope of this document. That said, a number of methods are possible, such as via a protocol exchange, or via a controller that manages both the sender and the receiver using NETCONF/YANG, BGP, PCEP, etc. These are meant as possible examples and not to constrain the future definition of such advertisement methods.

While the SFF label will usually be at the bottom of the label stack, there may be cases where there are additional label stack entries beneath it. For example, when an ACH is carried that applies to the SFF, a GAL [RFC5586] will be in the label stack below the SFF. Similarly, an ELI/EL [RFC6790] may be carried below the SFF in the label stack. This is identical to the situation with VPN labels.

2.1. MPLS Label Stack Construction at the Sending Node

When one SFF wishes to send an SFC packet with the NSH to another SFF over an MPLS transport network, a label stack needs to be constructed by the MPLS node that contains the sending SFF in order to transport the packet to the destination MPLS node that contains the receiving SFF. The label can be constructed as follows:

1. Push on zero or more labels that are interpreted by the destination MPLS node, such as the Generic Associated Channel [RFC5586] label (see OAM Considerations below).

2. Push on the SFF Label to identify the desired SFF in the receiving MPLS node.

3. Push on zero or more additional labels such that (a) the resulting label stack will cause the packet to be transported to the destination MPLS node, and (b) when the packet arrives at the destination node, either:

   * the SFF Label will be at the top of the label stack, or

   * the SFF Label will rise to the top of the label stack before the packet is forwarded to another node and before the packet is dispatched to a higher layer.

2.2. SFF Label Processing at the Destination Node

The destination MPLS node performs a lookup on the SFF label to retrieve the next-hop context between the SFF and SF, e.g. to retrieve the destination MAC address in the case where native Ethernet encapsulation is used between SFF and SF. How the next-hop context is populated is out of the scope of this document.

The receiving MPLS node then pops the SFF Label (and any labels beneath it) so that the destination SFF receives the SFC packet with the NSH is at the top of the packet.

3. Equal Cost Multipath (ECMP) Considerations

As discussed in [RFC4928] and [RFC7325], there are ECMP considerations for payloads carried by MPLS.

Many existing routers use deep packet inspection to examine the payload of an MPLS packet, and if the first nibble of the payload is equal to 0x4 or 0x6, these routers (sometimes incorrectly, as discussed in [RFC4928]) assume that the payload is IPv4 or IPv6 respectively, and as a result, perform ECMP load balancing based on (presumed) information present in IP/TCP/UDP payload headers or in a combination of MPLS label stack and (presumed) IP/TCP/UDP payload headers in the packet.

For SFC, ECMP may or may not be desirable. To prevent unintended ECMP when it is not desired, the NSH Base Header was carefully constructed so that the NSH could not look like IPv4 or IPv6 based on its first nibble. See Section 2.2 of [RFC8300] for further details.

If ECMP is desired when SFC is used with an MPLS transport network, there are two possible options, Entropy [RFC6790] and Flow-Aware Transport [RFC6391] labels. A recommendation between these options, and their proper placement in the label stack, is for future study.

4. Operations, Administration, and Maintenance (OAM) Considerations

OAM at the SFC Layer is handled by SFC-defined mechanisms [RFC8300]. However, OAM may be required at the MPLS transport layer. If so, then standard MPLS-layer OAM mechanisms such as the Generic Associated Channel [RFC5586] label may be used.

5. IANA Considerations

This document does not request any actions from IANA.

Editorial note to RFC Editor: This section may be removed at your discretion.

6. Security Considerations

This document describes a method for transporting SFC packets using the NSH over an MPLS transport network. It follows well-established MPLS procedures and does not define any new protocol elements or allocate any new code points. It is therefore operationally equivalent to other existing SFC transport encapsulations as defined in [RFC8300]. As such, it should have no effect on SFC security as already discussed in Section 8 of [RFC8300].

7. Acknowledgements

The authors would like to thank Jim Guichard, Eric Rosen, Med Boucadair, Sasha Vainshtein, and Jeff Tantsura for their reviews and comments.

8. References

8.1. Normative References

[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>.

[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>.

8.2. Informative References

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-06 (work in progress), April 2018.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, DOI 10.17487/RFC4928, June 2007, <https://www.rfc-editor.org/info/rfc4928>.

[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>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <https://www.rfc-editor.org/info/rfc6391>.

[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>.

[RFC7325] Villamizar, C., Ed., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, DOI 10.17487/RFC7325, August 2014, <https://www.rfc-editor.org/info/rfc7325>.

[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

Andrew G. Malis
Huawei Technologies

Email: agmalis@gmail.com

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Joel M. Halpern
Ericsson

Email: joel.halpern@ericsson.com

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

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

N. Nainar
C. Pignataro
F. Iqbal
Cisco Systems, Inc.
A. Vainshtein
ECI Telecom

December 18, 2018

RFC8287 Sub-TLV Length Clarification
draft-nainar-mpls-rfc8287-errata-01

Abstract

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

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

Status of This Memo

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

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

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

This Internet-Draft will expire on June 21, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

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

1. Introduction

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

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

2. Terminology

This document uses the terminologies defined in [I-D.ietf-spring-segment-routing], [RFC8029], [RFC8287] and so the readers are expected to be familiar with the same.

3. Requirements notation

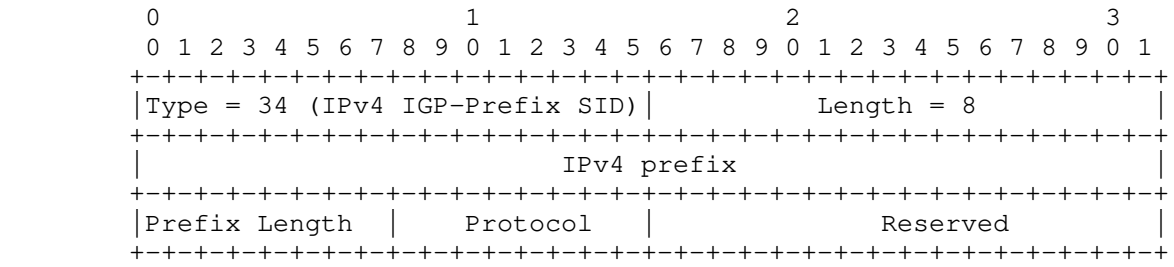
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

4. Length field clarification for Segment ID Sub-TLVs

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

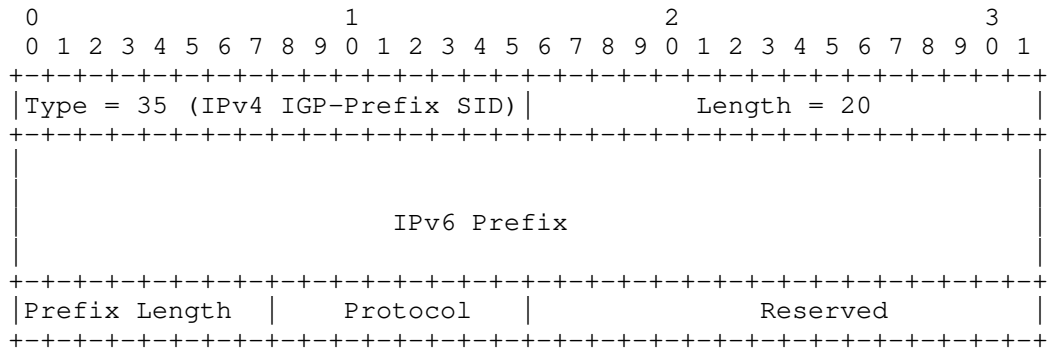
4.1. IPv4 IGP-Prefix Segment ID Sub-TLV

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



4.2. IPv6 IGP-Prefix Segment ID Sub-TLV

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

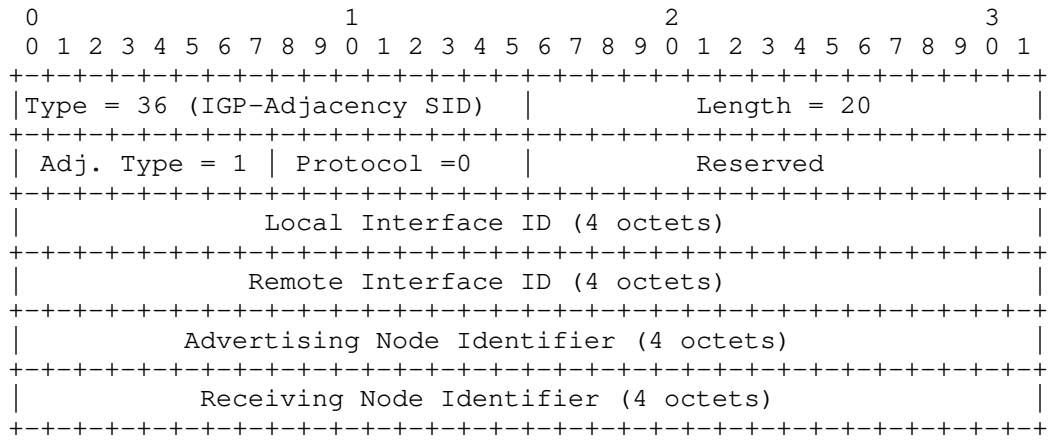


4.3. IGP-Adjacency Segment ID Sub-TLV

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

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

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



5. IANA Considerations

This document does not introduce any IANA consideration.

6. Security Considerations

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

7. Contributors

The below individuals contributed to this document:

Zafar Ali, Cisco Systems, Inc.

8. Acknowledgement

To be Updated

9. Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

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

[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>.

Authors' Addresses

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

Email: naikumar@cisco.com

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

Email: cpignata@cisco.com

Faisal Iqbal
Cisco Systems, Inc.
2000 Innovation Dr
Ottawa, ON 3E8
Canada

Email: faiqbal@cisco.com

Alexander Vainshtein
ECI Telecom
Israel

Email: vainshtein.alex@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 16, 2019

H. Song, Ed.
Z. Li
T. Zhou
Huawei
July 15, 2018

MPLS Extension Header
draft-song-mpls-extension-header-00

Abstract

Motivated by the need to support multiple in-network services and functions in an MPLS network, this document describes a generic method to encapsulate extension headers into MPLS packets. The encapsulation method allows stacking multiple extension headers and quickly accessing any of them as well as the original upper layer protocol header and payload. We show how the extension header can be used to support several new network applications.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Motivation 2
2. MPLS Extension Header 4
3. Operation on MPLS Extension Headers 7
4. Use Cases 8
5. Summary 8
6. Security Considerations 9
7. IANA Considerations 9
8. Contributors 10
9. Acknowledgments 10
10. References 10
10.1. Normative References 10
10.2. Informative References 11
Authors' Addresses 12

1. Motivation

Some applications require adding instructions and/or metadata to user packets within a network. Such examples include In-situ OAM (IOAM) [I-D.brockners-inband-oam-requirements] and Service Function Chaining (SFC) [RFC7665]. New applications are emerging. It is possible that the instructions and/or metadata for multiple applications are stacked together in one packet to support a compound service.

However, the encapsulation of the new header(s) poses some challenges to the ubiquitous MPLS networks. A problem of the MPLS protocol header is that there is no explicit indicator for the upper layer protocols. The succinct MPLS header provide little room to encode any extra information. Moreover, the backward compatibility issue discourages any attempts trying to overload the semantics of the existing MPLS header fields.

The similar "header extension" requirement for MPLS has led to several proposals. A special Generic Associated Channel Label (GAL) [RFC5586] is assigned to support the identification of an Associated Channel Header (ACH). Later, it was proposed to use GAL to indicate

the presence of a Metadata Channel Header (MCH) [I-D.guichard-sfc-mpls-metadata] as well.

GAL has several limitations:

- It must be located at the bottom of a label stack for its chief use case of MPLS-TP. An LSR needs to scan the entire label stack to be able to identify the presence of a new header. This can impact the performance when the label stack is deep.

- When GAL is present, the first nibble of the word following the GAL needs to be checked to determine the header type. Since the value of the nibble cannot be greater than 3, this approach is neither scalable nor reliable.

- By design, GAL can only indicate the presence of a single header. Therefore, the solution alone is not sufficient to support adding multiple headers at the same time.

- The presence of GAL makes the network load balancing or deep packet inspection based on upper layer protocol headers and payload difficult.

In addition to the above limitations, it is not desirable to keep overloading GAL with new semantics. Instead of trying to patch on existing schemes, we propose a general mechanism to solve the above mentioned issues and create new innovation opportunities. We derive our scheme from the experience of the IPv4 to IPv6 evolution. The adoption of IPv6 is gaining its momentum. Ironically, this is not due much to the extended address space over IPv4. One true power of IPv6 is that it supports extension headers, which offer a huge innovation potential (e.g, network security, SRv6 [I-D.ietf-spring-segment-routing], network programming [I-D.filsfils-spring-srv6-network-programming], SFC [I-D.xu-clad-spring-sr-service-chaining], etc.). It is straightforward to introduce new in-network services into IPv6 networks through extension headers. For example, it has been proposed to carry IOAM header [I-D.brockners-inband-oam-transport] and NSH as new extension headers in IPv6 networks.

Nevertheless, IPv6 is not perfect either. For one thing, IPv6’s header overhead is large compared to MPLS. We would like to retain the header compactness in MPLS networks. On the other hand, IPv6’s extension headers are chained with the original upper layer protocol headers in a flat stack. One has to scan all the extension headers to access the upper layer protocol headers and the payload. This is inconvenient and raises some performance concerns for some

applications (e.g., DPI and ECMP). The new scheme for MPLS header extension needs to address these issues too.

2. MPLS Extension Header

From the previous discussion, we have laid out the design requirements to support extension headers in MPLS networks:

Performance: If possible, unnecessary label stack scanning and extension header scanning should be avoided.

Scalability: New applications can be easily supported by introducing new extension headers. Multiple extension headers can be easily stacked together to support multiple services simultaneously.

Backward Compatibility: Legacy devices which do not recognize the extension header option should still be able to forward the packets as usual. If a device recognize some of the extension headers but not the others in an extension header stack, it can process the known headers only while ignoring the others.

To support the extension header in MPLS, we need to assign a new special label, namely the Extension Header Label (EHL). So far 8 special label values are left unsigned by IANA (which are 4 to 6 and 8 to 12). We believe this use case is significant enough to deserve one dedicated special label. Alternatively, a two label scheme with the use of the extension label (XL) plus an EHL is possible, but it does use one more label. It is also possible to use FEC labels to indicate the presence of extension headers. Although this approach avoid the need of a new special label, it introduces a good deal of complexity into the control plane. In the remaining of the document, we assume a special EHL is assigned.

The format of the MPLS packets with extension headers is shown in Figure 1. An EHL can be located in anywhere in an MPLS label stack. However, if there are legacy devices which do not recognize the EHL in the network, then for backward compatibility, the EHL must be located at the bottom of the stack (i.e., only the MPLS tunnel ends and EHL-aware nodes will look up and process it). Otherwise, the EHL can be located close to the top of the stack for better lookup performance.

The format of an EHL is the same as an MPLS label. The first 20-bit label value will be assigned by IANA. The BoS bit is used to indicate the location of the label. The other fields, CoS and TTL, are unused in the context of EHL.

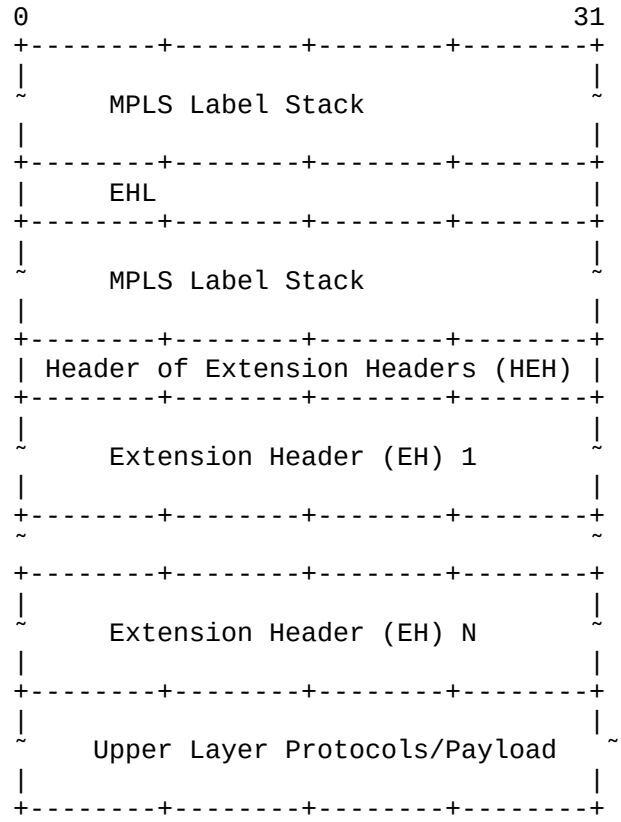


Figure 1: MPLS with Extension Header

Following the MPLS label stack is the 4-octet Header of Extension Headers (HEH), which indicates the total number of extension headers in this packet, the overall length of the extension headers, and the type of the next header. The format of the HEH is shown in Figure 2.

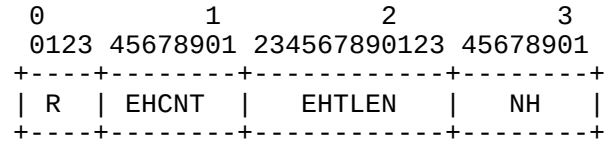


Figure 2: HEH Format

The meaning of the fields in an HEH is as follows:

R: 4-bit reserved.

EHCNT: 8-bit unsigned integer for the Extension Header Counter. This field keeps the total number of extension headers included in this packet. It does not count the original upper layer protocol headers.

EHTLEN: 12-bit unsigned integer for the Extension Header Total Length in 4-octet units. This field keeps the total length of the extension headers in this packet, not including the HEH itself.

NH: 8-bit selector for the Next Header. This field identifies the type of the header immediately following the HEH.

The EHCNT field can be used to keep track of the number of extension headers when some headers are inserted or removed at some network nodes. The EHLEN field can help to skip all the extension headers in one step if the original upper layer protocol headers or payload need to be accessed.

The format of an Extension Header (EH) is shown in Figure 3.

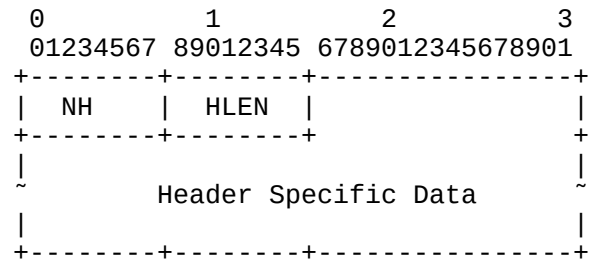


Figure 3: EH Format

The meaning of the fields in an EH is as follows:

NH: 8-bit selector for the Next Header. This field identifies the type of the EH immediately following this EH.

HLEN: 8-bit unsigned integer for the Extension Header Length in 4-octet units, not including the first 4 octets.

Header Specific Data: Variable length field for the specification of the EH. This field is 4-octet aligned.

The extension headers as well as the first original upper layer protocol header are chained together through the NH field in HEH and EHs. The encoding of NH uses the same values as the IPv4 protocol field. Values for new EH types shall be assigned by IANA.

Specifically, the NH field of the last EH in a chain can have two special values, which shall be assigned by IANA:

NONE: Indicates that there is no any other header and payload after this header. This can be used to transport packets with only extension header(s).

UNKNOWN: Indicates that the type of the header after this header is unknown. This is intended to be compatible with the original MPLS design in which the upper layer protocol type is unknown from the MPLS header alone.

3. Operation on MPLS Extension Headers

When the first EH X needs to be added to an MPLS packet, an EHL is inserted into the proper location in the MPLS label stack. A HEH is then inserted after the MPLS label stack, in which EHCNT is set to 1, EHTLEN is set to the length of X in 4-octet units, and NH is set to the header value of X. At last, X is inserted after the HEH, in which NH and HELN are set accordingly. Note that if this operation happens at a PE device, the upper layer protocol is known before the MPLS encapsulation, so its value can be saved in the NH field if desired. Otherwise, the NH field is filled with the value of "UNKNOWN".

When an EH Y needs to be added to an MPLS packet which already contains extension header(s), the EHCNT and EHTLEN in the HEH are updated accordingly (i.e., EHCNT is incremented by 1 and EHTLEN is incremented by the size of Y in 4-octet units). Then a proper location for Y in the EH chain is located. Y is inserted at this location. The NH field of Y is copied from the previous EH’s NH field (or from the HEH’s NH field, if Y is the first EH in the chain). The previous EH’s NH value, or, if Y is the first EH in the chain, the HEH’s NH, is set to the header value of Y.

Deleting an EH simply reverses the above operation. If the deleted EH is the last one, the EHL and HEH can also be deleted.

When processing an MPLS packet with extension headers, the node needs to scan through the entire EH chain and process the EH one by one. The node should ignore any unrecognized EH.

4. Use Cases

In this section, we show how MPLS extension header can be used to support several new network applications.

In-situ OAM: In-situ OAM (IOAM) records flow OAM information within user packets while the packets traverse a network. The instruction and collected data are kept in an IOAM header [I-D.ietf-ippm-ioam-data]. When applying IOAM in an MPLS network, the IOAM header can be encapsulated as an MPLS extension header.

NSH: Network Service Header (NSH) [RFC8300] provides a service plane for Service Function Chaining (SFC). NSH maintains the SFC context and metadata. If MPLS is used as the transport protocol for NSH, NSH can be encapsulated as an MPLS extension header.

Network Telemetry and Measurement: A network telemetry and instruction header can be carried as an extension header to instruct a node what type of network measurements should be done. For example, the method described in [RFC8321] can be implemented in MPLS networks since the EH provides a natural way to color MPLS packets.

Network Security: Security related functions often require user packets to carry some metadata. In a DoS limiting network architecture, a "packet passport" header is used to embed packet authentication information for each node to verify.

Segment Routing: MPLS extension header can support the implementation of a new flavor of the MPLS-based segment routing, with better performance and richer functionalities. The details will be described in another draft.

With MPLS extension headers, multiple in-network applications can be stacked together. For example, IOAM and SFC can be applied at the same time to support network OAM and service function chaining. A node can stop scanning the extension header stack if all the known headers it can process have been located. For example, if IOAM is the first EH in a stack and a node is configured to process IOAM only, it will stop searching the EH stack when the IOAM EH is found.

5. Summary

Evidenced by the existing and emerging use cases, MPLS networks need a standard way to support extension headers. In Figure 4, we summarize the potential schemes that allow MPLS packets to carry extension headers and list the main issues for each scheme.

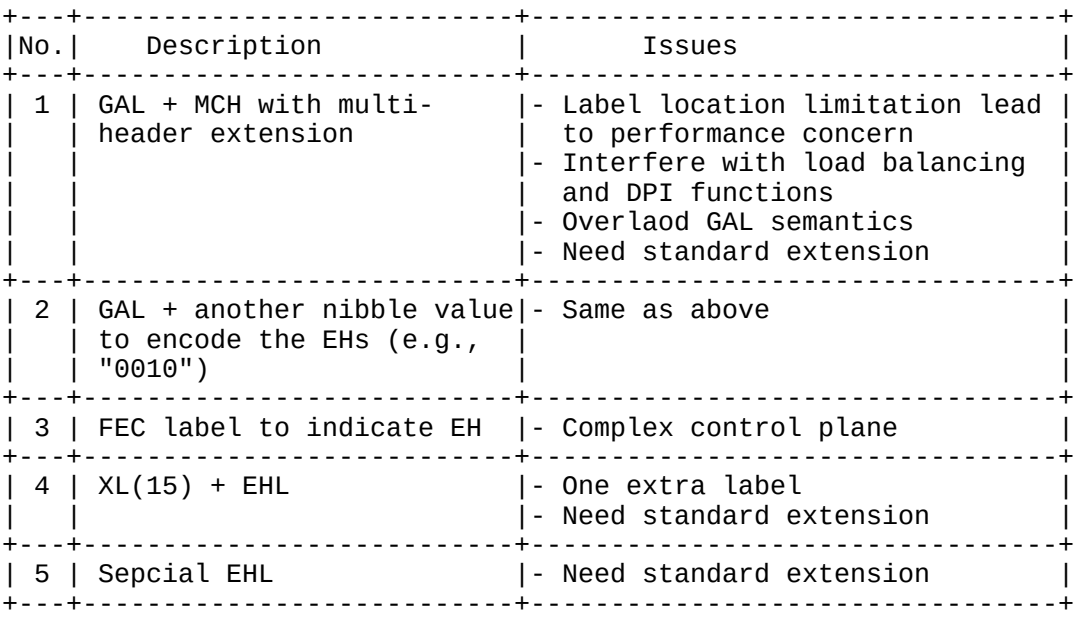


Figure 4: Potential Schemes for MPLS Extension Headers

Through comprehensive considerations on the pros and cons of each scheme, we currently recommend the scheme No.5. The proposed MPLS extension header scheme provides a generic way to support in-network services and functions in MPLS networks.

6. Security Considerations

TBD

7. IANA Considerations

This document requires IANA to assign a new special MPLS label value ("EHL") which is dedicated to indicate the presence of MPLS extension header(s).

This document also requires IANA to assign two new protocol numbers which are used to indicate no next header ("NONE") or an unknown next header ("UNKNOWN").

The new header type values shall be assigned by IANA on a case-by-case basis.

8. Contributors

The other contributors of this document are listed as follows.

- James Guichard
- Stewart Bryant
- Andrew Malis

9. Acknowledgments

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>.

[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>.

[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>.

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

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

10.2. Informative References

[I-D.brockners-inband-oam-requirements]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., <>, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

[I-D.brockners-inband-oam-transport]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport-05 (work in progress), July 2017.

[I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-05 (work in progress), July 2018.

[I-D.guichard-sfc-mpls-metadata]
Guichard, J., Pignataro, C., Spraggs, S., and S. Bryant, "Carrying Metadata in MPLS Networks", draft-guichard-sfc-mpls-metadata-00 (work in progress), September 2013.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-03 (work in progress), June 2018.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[I-D.xu-clad-spring-sr-service-chaining]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Decraene, B., Yadlapalli, C., Henderickx, W., Salsano, S., and S. Ma, "Segment Routing for Service Chaining", draft-xu-clad-spring-sr-service-chaining-00 (work in progress), December 2017.

Authors' Addresses

Haoyu Song (editor)
Huawei
2330 Central Expressway
Santa Clara
USA

Email: haoyu.song@huawei.com

Zhenbin Li
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: lizhenbin@huawei.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com

MPLS
Internet-Draft
Intended status: Informational
Expires: November 8, 2019

Z. Zhang
Juniper Networks
S. Esale
Juniper Networks, Inc.

May 7, 2019

Resilient MPLS Rings and Multicast
draft-zzhang-mpls-rmr-multicast-03

Abstract

With Resilient MPLS Rings (RMR), although all existing multicast procedures and solutions can work as is, there are optimizations that could be done for RSVP-TE P2MP tunnel signaling and Fast-ReRouting for both mLDP and RSVP-TE P2MP tunnels. This document describes RMR multicast on a high level, with detailed protocol procedure for RSVP-TE P2MP optimizations specified in a separate document. This document also discusses end to end multicast when there are RMRs.

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 November 8, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction 2
2. P2MP/MP2MP Tunnels on a Ring 3
2.1. Tunnel Protection and FRR 3
3. End to End Tunnels with Rings 4
4. End to End Native Multicast with Rings 4
4.1. Native Multicast in the Global Routing Table 4
4.2. mLDP Inband Signaling 4
4.3. Overlay Multicast Services 4
4.3.1. Tunnel Segmentation 5
5. Summary 5
6. Security Considerations 5
7. IANA Considerations 5
8. Acknowledgements 6
9. References 6
9.1. Normative References 6
9.2. Informative References 6
Authors' Addresses 7

1. Introduction

This document discusses how multicast works with Resilient MPLS Rings [I-D.ietf-mpls-rmr]. It is expected that readers are familiar with the concept and terms in [I-D.ietf-mpls-rmr].

All existing multicast procedures and solutions can work as is. This include both mpls multicast tunnels and end-to-end multicast that makes use of multicast tunnels. Ring topology is just a special case of general topologies so all existing RSVP-TE P2MP [RFC4875] and mLDP [RFC6388] tunnels can be set up using existing protocols and procedures. An Ingress Replication (IR) tunnel [RFC7988] consists a bunch of P2P LSPs, and it does not matter whether a component LSP is a plain old LSP or a Ring LSP.

On the other hand, there are optimizations that could be done for RSVP-TE P2MP tunnel signaling and Fast-ReRouting (FRR) for both mLDP and RSVP-TE P2MP tunnels. This document describes that on a high level, and discusses end to end multicast when there are RMRs even though RMR could be transparent to multicast.

2. P2MP/MP2MP Tunnels on a Ring

Because mLDP label mapping messages are merged as they propagate from the leaves towards the root, ring topology does not lead to any further optimization in tunnel signaling.

However RSVP-TE P2MP tunnel signaling and procedures can be greatly optimized, as specified in [I-D.zzhang-teas-rmr-rsvp-p2mp].

2.1. Tunnel Protection and FRR

Each node on a ring signals two counter-rotating MP2P Ring LSPs to itself. As these LSPs are self-signaled after the discovery of the ring, they can be used to protect P2MP LSPs on ring. So neither mLDP nor RSVP-TE has to setup a separate P2P bypass LSPs for link and node protection. For instance, consider a ring with 8 nodes:

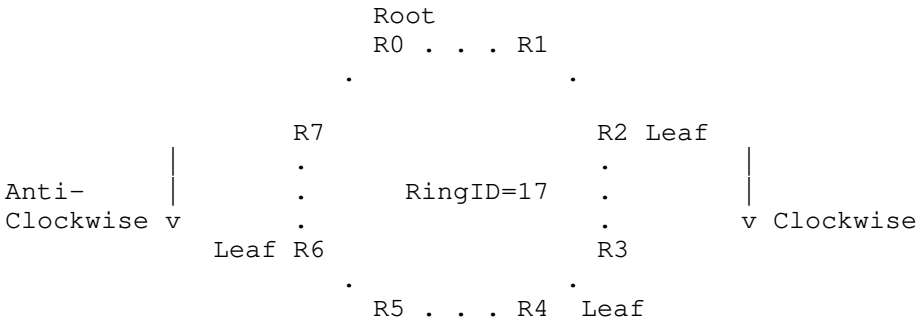


Figure 1: Ring with 8 nodes

Further, suppose a P2MP LSP is signaled with R0 as a root and R2, R4 and R6 as leafs. The P2MP LSP is formed as follows:

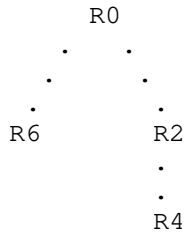


Figure 2: P2MP LSP

In the event of a link failure between R2 and R3, R2 the Point of Local Repair (PLR) tunnels P2MP LSP traffic on a anti-clockwise ring LSP to R3 the Merge Point (MP). Once the traffic is out of the ring

LSP on R3, it uses the regular P2MP LSP to reach R4. Similarly in the event of a node failure R3, R2 the PLR tunnels P2MP LSP traffic to R4 (the MP), which is also the leaf. Thus, the P2MP LSP uses the existing RSVP-TE ring LSPs for link and node protection.

3. End to End Tunnels with Rings

Consider a provider network that consists of one or more rings, optionally with a general topology connecting the rings. Multicast VPN [RFC6514], Ethernet VPN [RFC7432], VPLS [RFC4761] [RFC4762], or Global Table Multicast (GTM) via MVPN [RFC7716] overlay services are provided where end-to-end multipoint tunnels are needed across the entire network.

If the end to end tunnels are established by RSVP-TE P2MP, there is not much optimization that can be done for RMR, unless overlay-assisted tunnel segmentation is used. That is described in Section 4.3.1.

If the end to end tunnels are established by mLDP and RSVP-TE signaling is desired on part of the network, mLDP Over Targeted Sessions [RFC7060] can be used (without the help from the overlay service) to stack part of an mLDP tunnel over a RSVP-TE P2MP tunnel. If the RSVP-TE P2MP tunnel is over a ring, then the optimization described earlier can be used.

4. End to End Native Multicast with Rings

Consider a network that consists of some rings. In this network, end-to-end native multicast can take various forms described below.

4.1. Native Multicast in the Global Routing Table

This is typically signaled by PIM [RFC7761] end to end. This works for any topology and RMR does not make any difference.

4.2. mLDP Inband Signaling

This is specified in [RFC6826] [RFC7246] [RFC7438]. When part of a native (s,g) or (*,g) multicast tree needs to go over an mLDP domain, an mLDP tunnel is created for each multicast tree for the domain. RMR does not make any differences here.

4.3. Overlay Multicast Services

Overlay multicast services provided by MVPN/GTM/EVPN/VPLS use overlay multicast signaling to signal customer multicast state and tunnel binding. PE-PE multipoint underlay tunnels are used to distribute

multicast packets among PEs. Any kind of tunnel can be used, whether the provider network has rings or not, with or without the RMR related optimizations (Section 3).

4.3.1. Tunnel Segmentation

The MVPN/GTM/EVPN/VPLS PEs could span across ASes or areas. When the PE-PE multipoint tunnels cannot be signaled across AS/area boundaries, segmentation procedures can be used, as specified in [RFC6514, RFC7024] and [I-D.ietf-bess-evpn-bum-procedure-updates]. With the base MVPN/GTM/EVPN/VPLS procedures, PEs advertise I/S-PMSI A-D routes to signal traffic to tunnel binding, and the routes carry type and identification of multi-point tunnels used to carry corresponding traffic. With segmentation, the ASBRs/ABRs become segmentation points and they change the tunnel type/identification when they re-advertise the routes to the next AS/area. With this, each AS/area has its own tunnel of different type/identification, stitched together by the ASBRs/ABRs.

With segmentation, different RMRs could have their own tunnels, and RSVP-TE P2MP optimizations for RMRs could be applied. Notice that this is different from Section 3 in that overlay signaling is involved.

5. Summary

As described above, multicast in the presence of RMRs can work as is. RSVP-TE P2MP tunnel signaling can be optimized (to be specified separately). Tunnel protection/FRR can also be optimized for mLDP/ RSVP-TE P2MP tunnels.

6. Security Considerations

This is an informational document that describes how existing multicast protocols can be used with RMR, as well as possible RMR specific enhancements that will be specified separately. There are no security concerns to be discussed here, as they are already discussed in existing protocols or will be discussed in the specification for the enhancements.

7. IANA Considerations

This document does not request any allocations from IANA. The RFC Editor is requested to remove this section before publication.

8. Acknowledgements

The authors sincerely thank Loa Andersson for his careful review, comments and suggestions.

9. References

9.1. Normative References

[I-D.ietf-mpls-rmr]
Kompella, K. and L. Contreras, "Resilient MPLS Rings", draft-ietf-mpls-rmr-09 (work in progress), January 2019.

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

9.2. Informative References

[I-D.zzhang-teas-rmr-rsvp-p2mp]
Zhang, Z., Deshmukh, A., and R. Singh, "RSVP-TE P2MP Tunnels on RMR", draft-zzhang-teas-rmr-rsvp-p2mp-02 (work in progress), January 2019.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <https://www.rfc-editor.org/info/rfc4875>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC7060] Napierala, M., Rosen, E., and IJ. Wijnands, "Using LDP Multipoint Extensions on Targeted LDP Sessions", RFC 7060, DOI 10.17487/RFC7060, November 2013, <https://www.rfc-editor.org/info/rfc7060>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <https://www.rfc-editor.org/info/rfc7988>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Santosh Esale
Juniper Networks, Inc.

EMail: sesale@juniper.net

mpls
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

Z. Zhang
A. Deshmukh
R. Singh
Juniper Networks
July 2, 2018

RSVP-TE P2MP Tunnels on RMR
draft-zzhang-mpls-rmr-rsvp-p2mp-00

Abstract

This document specifies the optimization in RSVP-TE P2MP tunnel signaling over Resilient MPLS Rings (RMR).

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.

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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Specification 4
2.1. RMR Object 4
2.2. Procedures 5
2.2.1. PATH Message/State 5
2.2.2. RESV Message/State 6
3. Security Considerations 6
4. Acknowledgements 6
5. References 6
5.1. Normative References 6
5.2. Informative References 6

1. Introduction

Traditional RSVP-TE P2MP tunnel signaling could be quite involving. With RMR, this could be significantly simplifed:

There is no need for ERO/RRO/SERO/SRRO or hop by hop rouing. The tunnel ingress simply sends PATH messages in one or both directions of the ring, depending on how leaves are best reached. The <S2L Sub-LSP Descriptor List> only needs to list the tunnel leaves, and a transit router does not need to "branch" a PATH message into multiple ones Therefore, unless there are many tunnel leaves on a huge ring, a single PATH message is enough. In the rare situation of a large tunnel with many leaves to list, a small number of PATH messages should suffice. Additionally, there is no need to signal and maintain individual sub-LSPs (one for each leaf) any more. As a result, corresponding PATH/RESV state is also reduced. Each node only needs to maintain a single PATH state and a single RESV state for each P2MP tunnel, and the RESV state does not need to track individual leaves - it just need to track if a RESV is received from downstream and/or if this node itself is a leaf.

A RESV message is triggered to the PHOP when the RESV state is first created (either because the node is a leaf or because a RESV message is received from downstream) and it is refreshed periodically. A RESV Tear is sent when the RESV state is deleted (when the node is no longer a Leaf and the RESV from downstream has timed out or a RESV Tear is received).

Optionally, the tunnel ingress may not need to list any/all leaves. It could simply send the PATH message around the ring, with the <S2L Sub-LSP Descriptor List> listing the root itself. Through methods outside the scope of this document, a node determines if it is a leaf of the tunnel, and if yes, it will send back a RESV message. With this, a single PATH message is surely enough.

In this document, leaves in <S2L Sub-LSP Descriptor List> are referred to as explicit leaves, and leaves not listed there but self-determined by ring nodes are referred to as implicit leaves. There could be both explicit and implicit leaves for a tunnel. The ingress allows implicit leaves by including itself as the last one in the <S2L Sub-LSP Descriptor List>.

Optionally, the RESV message could also include a <S2L Sub-LSP Descriptor List> to list all the leaves on the established tunnel so that the each node knows its downstream leaves. In that case, when the set of downstream leaves changes, a RESV message with the new <S2L Sub-LSP Descriptor List> is triggered.

Adding/removing explicit leaves is straighforward. The ingress simply sends a triggered PATH message with new <S2L Sub-LSP Descriptor List>. As it passes around the ring, each node determines if it is an explicit leaf and updates its state accordingly. The triggered PATH message does not have to go all the way to the last leaf - if on a node the <S2L Sub-LSP Descriptor List> in the to-be-sent PATH message is the same as what was sent before, the triggered PATH message will not be sent further.

To indicate that the tunnel signaling is with above mentioned RMR optimizations, a new object is included in the PATH message to specify the Ring ID and direction.

Link/Node protection is achieved by tunneling packets to the next node using the Ring LSP to that node in the other direction. This does not need any additional signaling but is based on a reasonable premise that unicast Ring LSPs are always in place. Once the ingress learns the failure (through IGP discovery or through other error detection/notification mechanisms), global repair kicks in to reach some leaves via PATH message sent in the other direction. Before global repair is finished, traffic continues to flow in the original path except that at the failure point it is tunneled to the next node.

If an RMR is just part of a general RSVP network the optimization can also be applied on the ring nodes. If the tunnel ingress knows the leaves that are on the ring, it could put all those leaves in the single PATH message and construct the ERO/SERO only towards the entry points on the ring. The entry points then includes the RMR object in the PATH messages that they send. For leaves beyond the ring, the ingress may include the exit points on the ring as loose hops in the ERO/SERO, and when a ring node needs to send the PATH message off the ring, it removes the RMR object. Details will be provided in future revisions of this document.

2. Specification

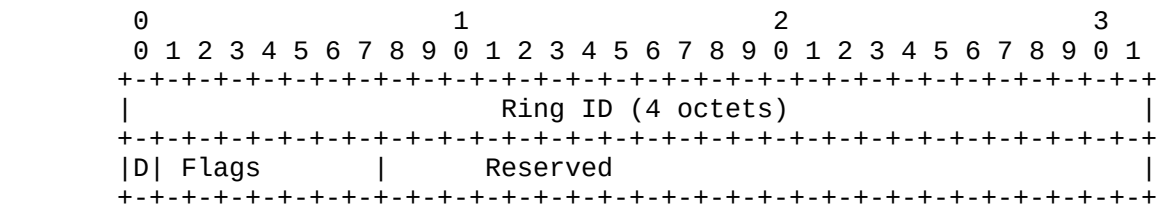
2.1. RMR Object

The RMR object is a new object of the following:

- Class Name: RMR
- Class-Num: TBA1 (to be assigned by IANA)
- C-Type: TBA2 (to be assigned by IANA)

The format of the object content following the common object header

is the folowing:



Following the 4-octect Ring ID, there is an 8-bit Flags field. The first bit of the Flags field indicates the direction. If it is set, it is clockwise direction. Otherwise, it is anti-clockwise.

2.2. Procedures

This section describes the differces in the procedures for ring nodes to set up RSVP-TE P2MP tunnels across the ring, compared to the conventional non-RMR-aware case. For now it is assumed that all nodes (ingress, tranist, and leaves) on the tunnel are on the ring.

More details will be provided in future revisions.

2.2.1. PATH Message/State

The tunnel ingress includes the RMR object with the Ring ID and the direction flag bit set accordingly. The explicit tunnel leaves are encoded in the <S2L Sub-LSP Descriptor List>, and no ERO/SERO is included. If the tunnel allows implicit leaves, the descriptor list encodes the ingress itself as the last element. The message is sent to the next node on the ring in the direction specified in the RMR object, w/o using ERO/SERO or hop-by-hop routing.

When a node recevies a PATH message with the RMR object, it checks if itself is listed in the <S2L Sub-LSP Descriptor List>, or if the <S2L Sub-LSP Descriptor List> encodes the tunnel ingress as the last element and this node itself is an implicit leaf. If yes, it creates corresponding RESV state and sends a RESV message to the PHOP.

The receiving node removes itself from the <S2L Sub-LSP Descriptor List> in the PATH message, and saves the list locally. The PATH message is sent to the next node on the ring in the specified direction if one of the following conditions is met:

- The <S2L Sub-LSP Descriptor List> encodes the tunnel ingress itself as the last element.

- The <S2L Sub-LSP Descriptor List> is not empty and either the PATH state is newly created or the <S2L Sub-LSP Descriptor List> is different from the previously saved one.

If <S2L Sub-LSP Descriptor List> is empty and different from the previously saved one, a PATH Teardown is sent instead with the saved <S2L Sub-LSP Descriptor List>.

2.2.2. RESV Message/State

A ring node may know that it is a leaf when the PATH message is first processed as described in the previous section. In case of implicit leaves, it may become a leaf after the PATH messages has been processed. A non-leaf node may also receive a RESV message from its NHOP. In all cases, the node creates RESV state and sends a RESV message to the PHOP, w/o encoding RRO/SRRO.

If a ring node was a leaf but stops being a leaf, either because it is no longer listed in the <S2L Sub-LSP Descriptor List> or it is no longer an implicit leaf, it removes/updates corresponding local state. A RESV Teardown is sent to the PHOP if there is no RESV received from its downstream.

3. Security Considerations

This document does not introduce new security risks?

4. Acknowledgements

5. References

5.1. Normative References

[I-D.ietf-mpls-rmr] Kompella, K. and L. Contreras, "Resilient MPLS Rings", draft-ietf-mpls-rmr-04 (work in progress), March 2017.

5.2. Informative References

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Abhishek Deshmukh
Juniper Networks

EMail: adeshmukh@juniper.net

Ravi Singh
Juniper Networks

EMail: ravis@juniper.net