SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
N. Vaghamshi
Reliance
M. Nagarajah
Telstra
R. Foote
Nokia

July 11, 2020

# Enhanced Performance Delay and Liveness Monitoring in Segment Routing Networks
## draft-gandhi-spring-sr-enhanced-plm-02

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document defines procedure for Enhanced Performance Delay and Liveness Monitoring (PDLM) in Segment Routing networks. The procedure uses the probe messages defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP) Light) and RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) for end-to-end SR Paths including SR Policies with both SR-MPLS and SRv6 data planes.

Table of Contents

## 1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in Liveness Monitoring for detecting faults as well as Performance Delay Measurement (DM) and Loss Measurement (LM) are essential requirements to provide Service Level Agreements (SLAs) in SR networks.

The One-Way Active Measurement Protocol (OWAMP) defined in [RFC4656] and Two-Way Active Measurement Protocol (TWAMP) defined in [RFC5357] provide capabilities for the measurement of various performance metrics in IP networks using probe messages. The TWAMP Light [Appendix I in RFC5357] and the Simple Two-way Active Measurement Protocol (STAMP) [RFC8762] provide simplified mechanisms for active performance measurement in IP networks, alleviating the need for control-channel signaling by using configuration data model to provision a test-channel.

[I-D.gandhi-spring-twamp-srpm] defines procedure for performance measurement using TWAMP Light messages with user-defined IP/UDP paths in SR networks. [I-D.gandhi-spring-stamp-srpm] defines similar procedure using STAMP messages in SR networks. The procedure for one-way and two-way modes defined for delay measurement can also be applied to liveness monitoring of SR Paths. However, it limits the scale for number of PM sessions and fault detection interval since the probe query messages need to be punted from the forwarding path (to slow path or control plane) and response messages need to be injected.

For Liveness Monitoring, Seamless Bidirectional Forwarding Detection (S-BFD) [RFC7880] can be used in Segment Routing networks. However, S-BFD requires protocol support on the reflector node to process the S-BFD packets as packets need to be punted from the forwarding path in order to send the reply thereby limiting the scale for number of PM sessions and fault detection interval. In addition, S-BFD protocol does not have the capability today to enable performance delay monitoring in SR networks. Enabling multiple protocols in SR networks, S-BFD for liveness monitoring and TWAMP Light or STAMP for performance delay monitoring increases the deployment and operational complexities in SR networks.

This document defines procedure for Enhanced Performance Delay and Liveness Monitoring (PDLM) in Segment Routing networks. The procedure uses the probe messages defined in [RFC5357] (TWAMP Light) and [RFC8762] (STAMP) for end-to-end SR Paths including SR Policies with both SR-MPLS and SRv6 data planes.

## 2. Conventions Used in This Document

### 2.1. Requirements Language

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

### 2.2. Abbreviations

BFD: Bidirectional Forwarding Detection.

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

OWAMP: One-Way Active Measurement Protocol.

PDLM: Performance Delay and Liveness Monitoring.

PM: Performance Measurement.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

STAMP: Simple Two-way Active Measurement Protocol.

TWAMP: Two-Way Active Measurement Protocol.

## 2.3. Reference Topology

In the reference topology shown below, the nodes R1 and R5 are connected via Point-to-Point (P2P) SR Path such as SR Policy [I-D.ietf-spring-segment-routing-policy] originating on node R1 with endpoint on node R5.

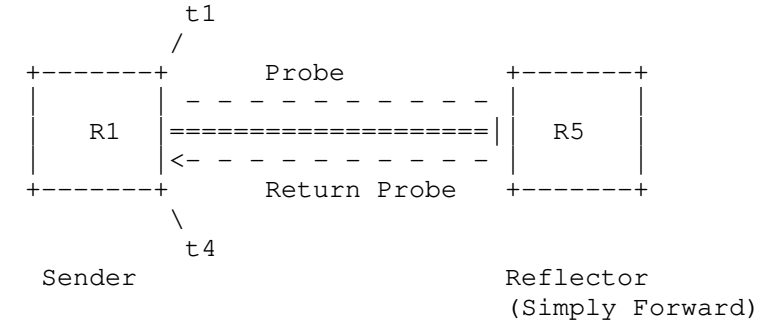


Figure 1: Reference Topology

## 2.4. Loopback Mode

In loopback mode, the sender node R1 initiates probe messages and the reflector node R5 forwards them back to the sender node R1 just like data packets for the normal traffic. The probe messages are not punted at the reflector node and it does not process them and generate response messages. The reflector node must not drop the loopback probe messages, for example, due to a local policy provisioned on the node.

# 3. Probe Messages

The TWAMP Light probe messages for delay measurement as defined in [RFC5357] or STAMP probe messages as defined in [RFC8762] are sent by the sender node R1 towards the reflector node R5 in loopback mode as shown in Figure 1. The probe messages are sent by the sender node on the congruent path of the data traffic flowing on the SR Path.

Both Source and Destination UDP ports in the probe messages are allocated dynamically or user-configured from the range specified in [RFC8762] and are different than the ports used for TWAMP Light and STAMP sessions. The Source and Destination IP addresses in the probe messages are set to the reflector and the sender node addresses,

respectively (representing the reverse path). The IPv4 Time To Live (TTL) and IPv6 Hop Limit (HL) are set to 255.

No PM session is created on the reflector node R5. As the probe message is not punted on the reflector node for processing, the Sender copies the 'Sequence Number' in 'Session-Sender Sequence Number' field directly. Also, the Sender Timestamp, Sender Error Estimate and Sender TTL fields [RFC5357] [RFC8762] in the probe message are not used. The rest of the fields are set as defined in [RFC5357] [RFC8762]

Timestamp format preferred is 64-bit PTPv2 [IEEE1588] as specified in [RFC8186], implemented in hardware. The NTP timestamp format MUST be supported [RFC5357], however, since PTPv2 is widely used, it SHOULD also be supported. In addition to adding the timestamp in the message, the "Error Estimate" field in the payload of the message can be updated using the procedure defined in [RFC4656].

## 3.1. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 2:

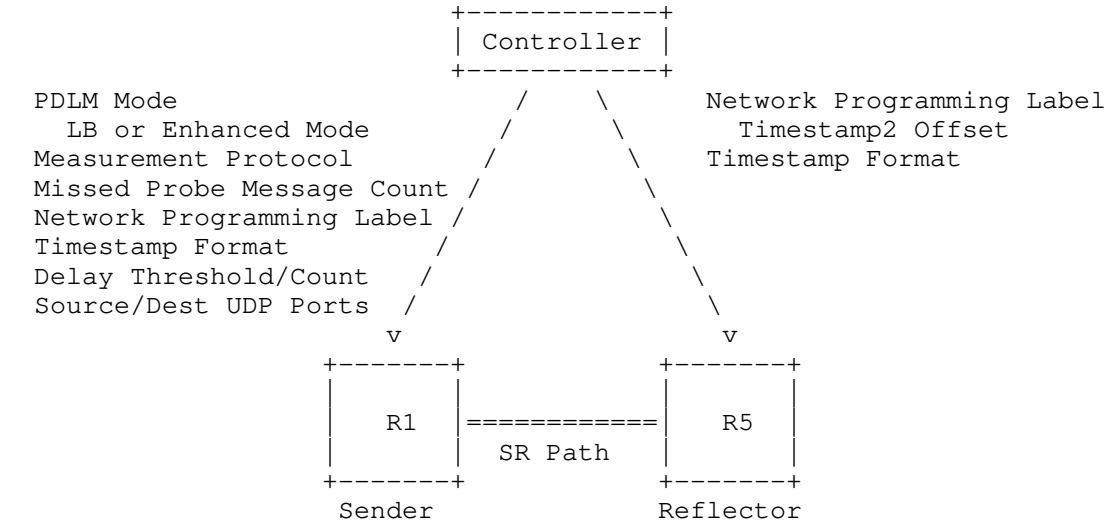


Figure 2: Example Provisioning Model

Example of Measurement Protocol is TWAMP Light and STAMP, example of Timestamp Format is 64-bit PTPv2 [IEEE1588] and NTP, etc.

The mechanisms to provision the sender and reflector nodes are outside the scope of this document.

## 4. Performance Delay and Liveness Monitoring

For performance delay and liveness monitoring of an end-to-end SR Path including SR Policy, PM probes in loopback mode is used. The PM probe messages are sent by the sender (head-end) node R1 to the reflector (endpoint) node R5 of the SR Policy as shown in Figure 1.

The probe messages are sent using the Segment List (SL) of the Candidate-paths of the SR Policy [I-D.ietf-spring-segment-routing-policy]. When a Candidate-path has more than one Segment Lists, multiple probe messages are sent, one using each Segment List. The return probe messages are received by the sender node via IP/UDP [RFC0768] return path by default. The Segment List of the return SR path can be added in the probe message header to receive the return probe message on a specific path using the mechanisms defined in [I-D.ietf-pce-binding-label-sid] and [I-D.ietf-pce-sr-bidir-path].

### 4.1. Probe Message for SR-MPLS

The TWAMP Light or STAMP probe messages for SR-MPLS data plane are sent using the MPLS header containing the label stack of the SR Policy as shown in Figure 3. In case of IP/UDP return path, the MPLS header is removed by the reflector node. The label stack can contain a reverse SR-MPLS path to receive the return probe message on a specific path. In this case, the MPLS header will not be removed by the reflector node.

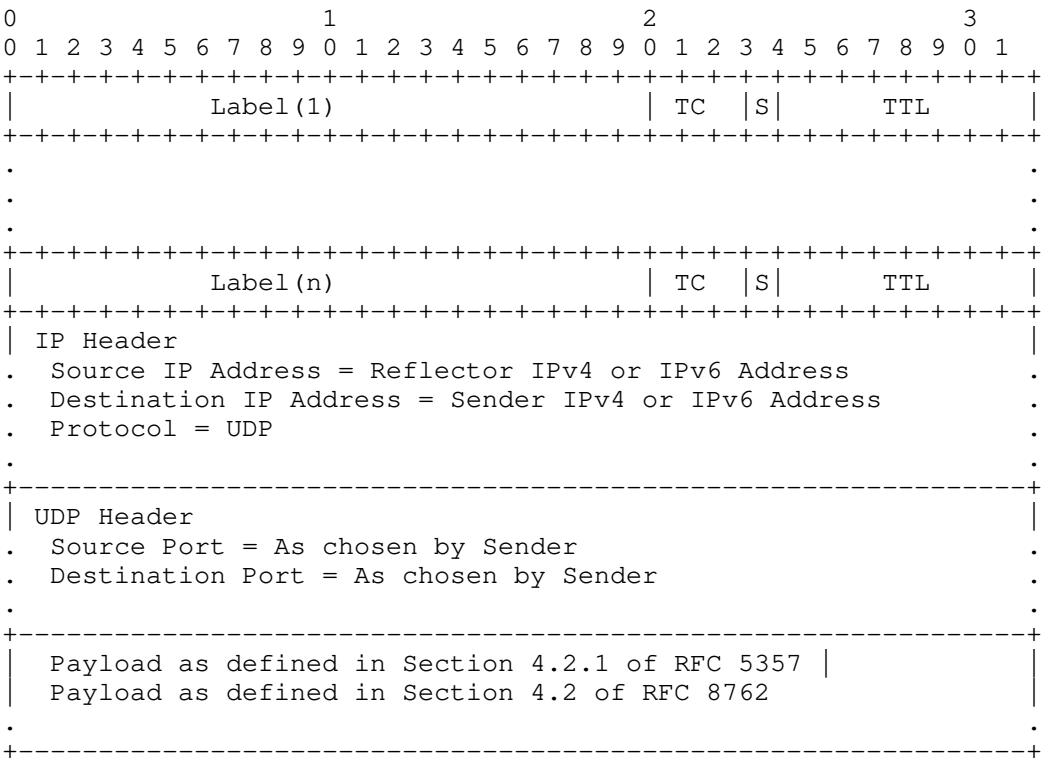


Figure 3: Example Probe Message for SR-MPLS

## 4.2. Probe Message for SRv6

The TWAMP Light or STAMP probe messages for SRv6 data plane are sent using the Segment Routing Header (SRH) [RFC8754] containing the Segment List of the SR Policy as shown in Figure 4. In case of IP/UDP return path, the SRH is removed by the reflector node. The Segment List can contain a reverse SRv6 path to receive the return probe message on a specific path. In this case, the SRH will not be removed by the reflector node.

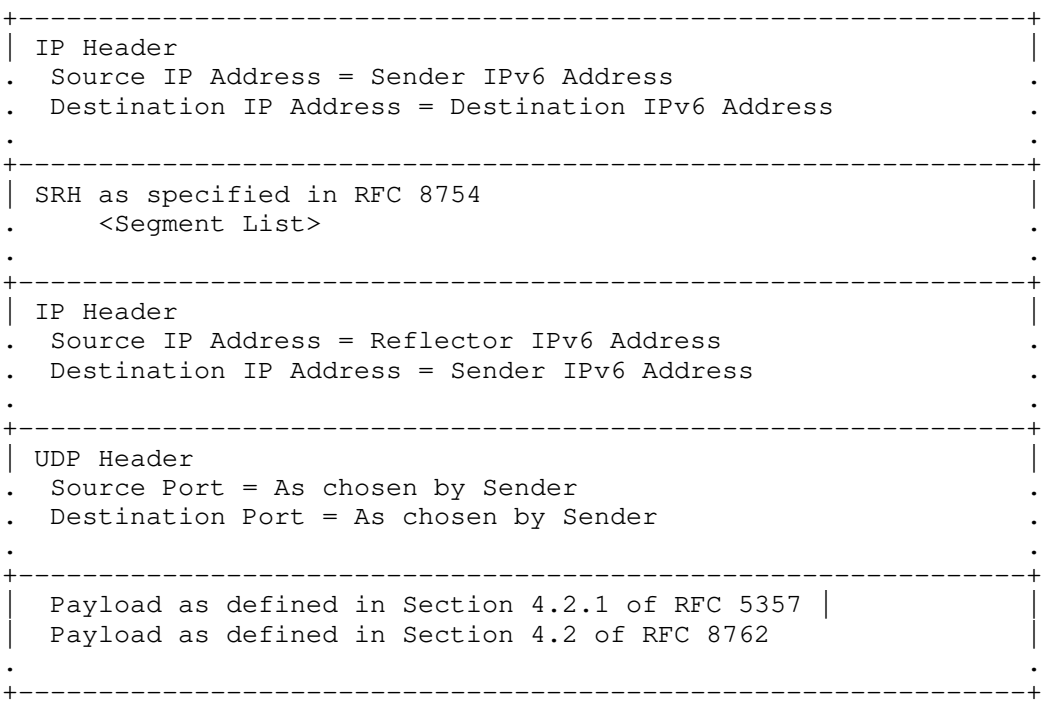


Figure 4: Example Probe Message for SRv6

## 5. Enhanced Performance Delay and Liveness Monitoring

The enhanced performance delay and liveness monitoring of an end-to-end SR Path including SR Policy is defined using the PM probes in "loopback mode enabled with network programming".

### 5.1. Loopback Mode Enabled with Network Programming

In "loopback mode enabled with network programming", both transmit (t1) and receive (t2) timestamps in data plane are collected by the probe messages sent in loopback mode as shown in Figure 5. The network programming function optimizes the "operations of punt, add receive timestamp and inject the probe packet" on the reflector node and it is implemented in hardware. The payload of the probe message is not modified by any intermediate nodes.

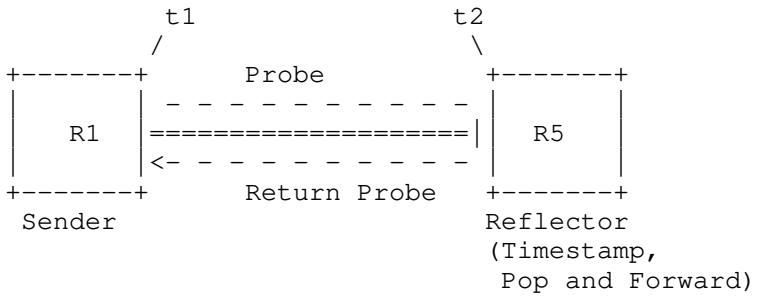


Figure 5: Loopback Mode Enabled with Network Programming

The sender node adds transmit (t1) timestamp in the payload of the TWAMP Light or STAMP probe message and clears the receive (t2) timestamp. The reflector node adds the receive timestamp in the payload of the received probe message without punting the message to slow-path (or control-plane). The reflector node only adds the receive timestamp if the source or destination address in the probe message matches the local node address to ensure that the receive timestamp is returned by the intended reflector node.

The network programming function enables the node to add receive timestamp in the payload of the probe message at a specific offset which is locally provisioned consistently in the network. In TWAMP Light message defined in Section 4.2.1 of [RFC5357] or STAMP message defined in [RFC8762] for delay measurement, the 64-bit receive timestamp is added at byte-offset 16 which is from the start of the payload.

## 5.2. Probe Message with Network Programming for SR-MPLS

In this document, new Timestamp Label (value TBD1) is defined for SR-MPLS data plane to enable network programming function for "timestamp, pop and forward" the received packet.

In the probe message for SR-MPLS, Timestamp Label is added in the MPLS header as shown in Figure 6, to collect "Receive Timestamp" field in the payload of the TWAMP Light [RFC5357] or STAMP probe message. The label stack for the reverse SR-MPLS path can be added after the Timestamp Label to receive the return probe message on a specific path. When a node receives a message with Timestamp Label, after timestamping the message at a specific offset, the node pops the Timestamp Label and forwards the message using the next label or IP header in the message (just like the data packets for the normal traffic).

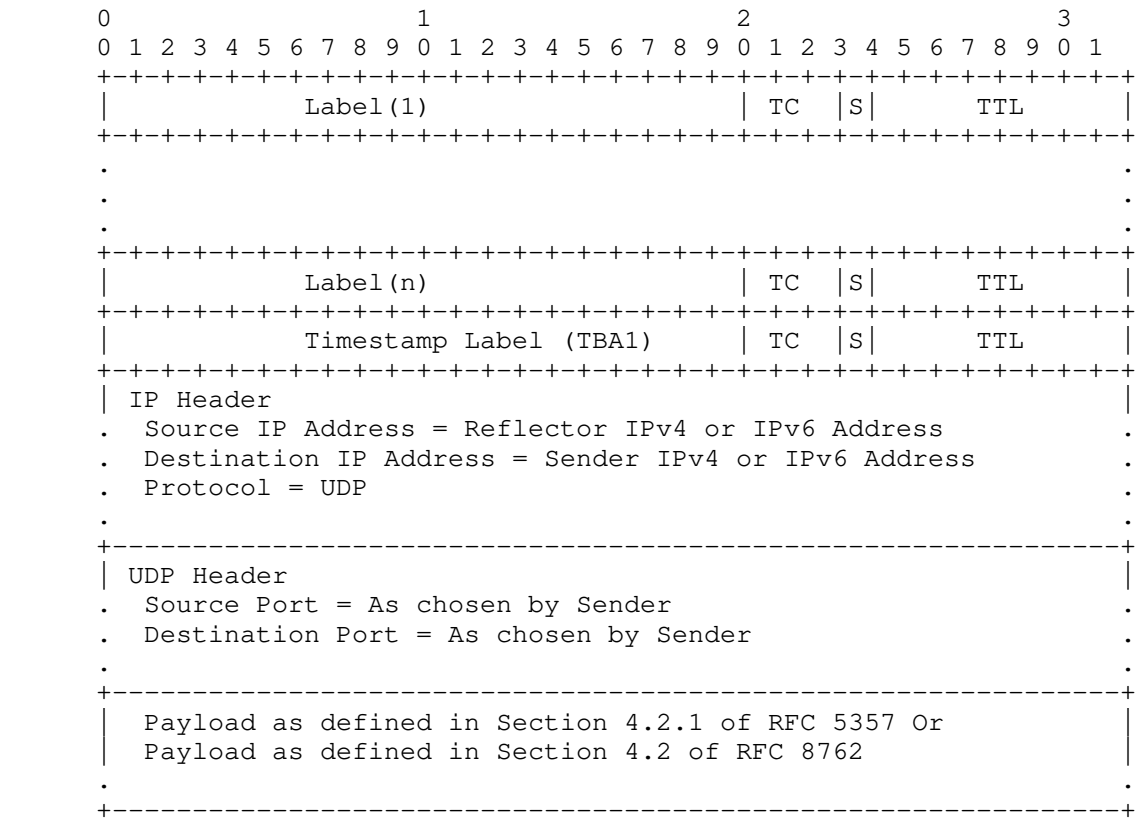


Figure 6: Example Probe Message with Timestamp Label for SR-MPLS

### 5.2.1. Node Capability for Timestamp Label

The ingress node needs to know if the egress node can process the Timestamp Label. The signaling extension for this capability exchange is outside the scope of this document.

Another way is to leverage a centralized controller (e.g., SDN controller) to program the ingress and egress nodes. In this case, the controller MUST make sure (e.g., by some capability discovery mechanisms outside the scope of this document) that the egress node can process the Timestamp Label.

### 5.2.2. Timestamp Label Allocation

Timestamp Label (value TBA1) can be allocated using one of the following methods:

- Labels assigned by IANA with value TBA1 from the Extended Special-Purpose MPLS Values [I-D.ietf-mpls-spl-terminology].

- Labels allocated by a Controller from the global table of the egress node. The Controller provisions the label on both ingress and egress nodes.

- Labels allocated by the egress node. The signaling or IGP flooding extension for this is outside the scope of this document.

### 5.3. Probe Message with Network Programming for SRv6

In this document, new Endpoint function "Timestamp and Forward (TSF)" (value TBD2) is defined for Segment Routing Header (SRH) [RFC8754] for SRv6 data plane to enable network programming function for "timestamp and forward" the received message.

In the probe message for SRv6, END.TSF function is added for the Endpoint Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 7, to collect "Receive Timestamp" field in the payload of the TWAMP Light [RFC5357] or STAMP probe message. When a node receives a packet with END.TSF function for the target SID which is local, after timestamping the packet at a specific offset, the node forwards the packet using the next SID or IP header in the packet (just like the packets for the normal traffic).

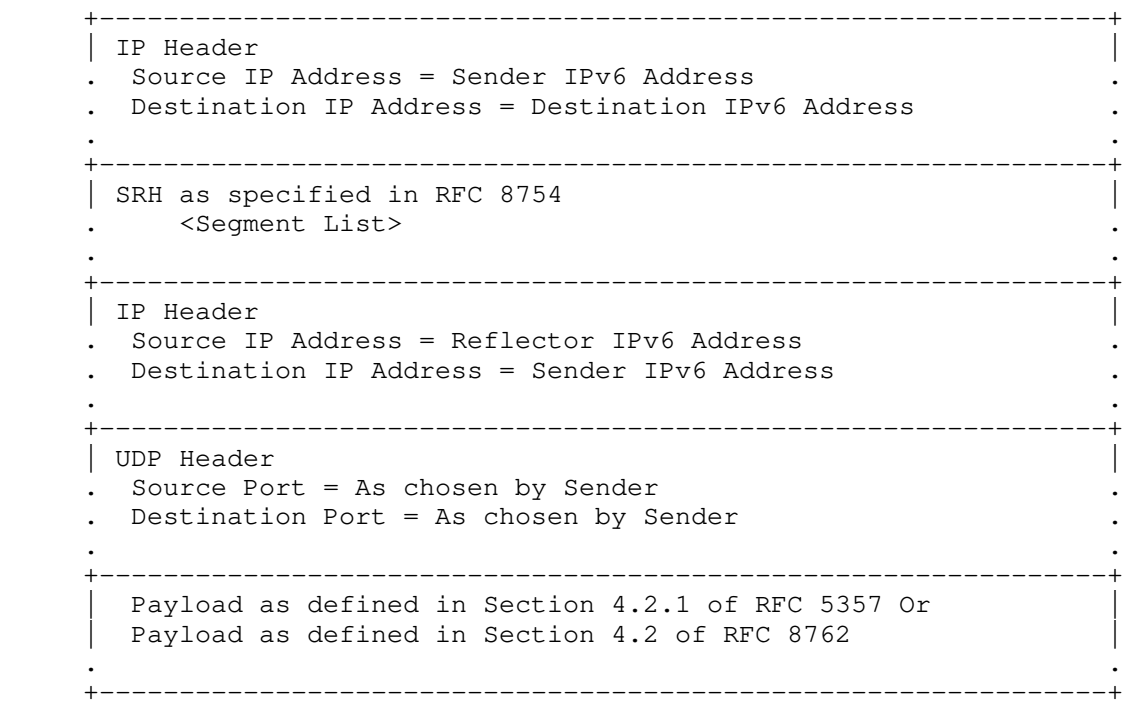


Figure 7: Example Probe Message with Endpoint Function for SRv6

## 6. ECMP Handling

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. The PM probe messages need to be sent to traverse different ECMP paths to monitor the liveness for an end-to-end SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. In IPv4 header of the PM probe messages, sweeping of Destination Address in 127/8 range can be used to exercise different ECMP paths in the loopback mode as long as the return path is also SR-MPLS. The Flow Label field in the outer IPv6 header can also be used for sweeping to exercise different ECMP paths.

## 7. Failure Notification

Liveness failure for SR Path is notified when consecutive N number of return probe messages are not received at the sender node, where N (Missed Probe Message Count) is locally provisioned value. Similarly, delay metrics are notified when consecutive M number of

probe messages have measured delay values exceed user-configured thresholds (absolute and percentage), where M is also locally provisioned value.

In loopback mode, the timestamps t1 and t4 are used to measure round-trip delay. In loopback mode enabled with network programming, the timestamps t1 and t2 are used to measure one-way delay.

## 8. Security Considerations

The Performance Delay and Liveness Monitoring is intended for deployment in the well-managed private and service provider networks. As such, it assumes that a node involved in a monitoring operation has previously verified the integrity of the path and the identity of the reflector node. If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the timestamp fields in received probe messages. The minimal state associated with these protocols also limits the extent of disruption that can be caused by a corrupt or invalid message to a single probe cycle. Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

## 9. IANA Considerations

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

| Value | Description | Reference |
|---|---|---|
| TBA1 | Timestamp Label | This document |

IANA is requested to allocate, within the "SRv6 Endpoint Behaviors Registry" sub-registry belonging to the top-level "Segment-routing with IPv6 data plane (SRv6) Parameters" registry [I-D.ietf-spring-srv6-network-programming], the following allocation:

| Value | Endpoint Behavior | Reference |
|---|---|---|
| TBA2 | END.TSF (Timestamp and Forward) | This document |

# 10. References

## 10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[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>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

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

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

## 10.2. Informative References

[IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8186] Mirsky, G. and I. Meilik, "Support of the IEEE 1588 Timestamp Format in a Two-Way Active Measurement Protocol (TWAMP)", RFC 8186, DOI 10.17487/RFC8186, June 2017, <https://www.rfc-editor.org/info/rfc8186>.

[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>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.gandhi-spring-twamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Performance Measurement Using TWAMP Light for Segment Routing Networks", draft-gandhi-spring-twamp-srpm-09 (work in progress), June 2020.

[I-D.gandhi-spring-stamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Performance Measurement Using STAMP for Segment Routing Networks", draft-gandhi-spring-stamp-srpm-01 (work in progress), June 2020.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-07 (work in progress), May 2020.

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-16 (work in progress), June 2020.

[I-D.ietf-mpls-spl-terminology] Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-02 (work in progress), May 2020.

[I-D.ietf-pce-binding-label-sid] Filsfils, C., Sivabalan, S., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-ietf-pce-binding-label-sid-03 (work in progress), June 2020.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "PCEP Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-02 (work in progress), March 2020.

## Acknowledgments

TBD

## Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Navin Vaghamshi
Reliance

Email: Navin.Vaghamshi@ril.com

Moses Nagarajah
Telstra

Email: Moses.Nagarajah@team.telstra.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2021

L. Han
China Mobile
F. Yang
Huawei Technologies

July 13, 2020

# Signal Degrade Indication Used in Segment Routing over MPLS Network
draft-han-mpls-sdi-sr-00

## Abstract

This document describes the typical use cases for signal degrade indication used in SR over MPLS networks. To satisfy the use cases and requirements of signal degrade indication, two extensions based on the BFD protocol and MPLS-TP OAM mechanisms are given respectively.

## 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 14, 2021.

## Copyright Notice

Table of Contents



## 1. Introduction

The importance and necessity of signal degrade notification used in Segment Routing over MPLS networks is discussed in [I-D.yang-mpls-ps-sdi-sr]. When signal degrade is detected, this information could be extended by means of protocols to perform the performance monitoring, fault management, and trigger of protection mechanism etc. Extensions on control plane, forwarding plane, management plane, and/or combination of any of them could be utilized to support the function of signal degrade indication. This document provides two protocol extensions used in SR over MPLS networks, by specifying the encapsulations and behaviors in detail.

In some of SR over MPLS networks, BFD [RFC5880] or enhanced SBFD [RFC7880] is widely utilized as the failure detection mechanism because of its' simplicity and efficiency characteristics. The indication of signal degrade could be adopted as one of the reasons of BFD state changes. In other SR over MPLS networks, MPLS-TP OAM [ITU-T G.8113.1] mechanisms are used instead of BFD or SBFD. In this scenario, the extension based on the OAM PDU format is proposed in this document to support the signal degarde indication.

## 2. Terminology

MPLS: Multiprotocol Label Switching

SR: Segment Routing

BFD: Bidirectional Forwarding Detection

SBFD: Seamless BFD

LER: Label Edge Router

LSR: Label Switching Router

MPLS-TP: Multiprotocol Label Switching - Tranport Profile

OAM: Operation, Administration and Maintena

GAL: Generic Associated Channel Label

G-ACh: Generic Associated Channel (G-ACh)

PDU: Protocol Data Unit

CCM: Continuity Check Message

## 3. Overview

The use cases and requirements have been discussed in [I-D.yang-mpls-ps-sdi-sr]. This document narrows the scope to the multi-hop SR over MPLS network, signal degrade is detected simply based on the physical bit error statistic on port level, no matter if the PHY is with or without forward error correction (FEC). Port level statistic is the intuitive approach to be best understood in the equipment and network systems. In practice of deployment, flexible configuration of the watermark to trigger the indication of signal degrade is preferred.

As mentioned in [I-D.yang-mpls-ps-sdi-sr], signal degrade can happen in any link or node in SR over MPLS networks, such as LERs and LSRs. LERs can detect the signal degrade fault, or directly trigger the protecton switch mechanisms once it detects the signal degrade reaches at a certain level. However, LSRs may need further considerations. In SR over MPLS networks, since only the headend LER knows all the segments in the label stack, the intermediate LSRs does not know the entire label stack. There is no other choice of forwarding path to avoid the impact of signal degrade on the LSR. Thus, the signal degrade information should be spread to other LSRs

and LERs, and consequent behaviors on LSRs or LERs are executed depending on the choices of the protection mechanisms.

The notification mechanism is best used through a forwarding protocol, not through the centralized Network Management System (NMS) or a SDN controller, to make sure the notification could be fast enough. Furthermore, carrying the signal degrade information in a control protocol is considered as well. In this case, the extensions of BFD control packet format and MPLS-TP CCM OAM PDU format are made to spread the signal degrade information.

Though the signal degrade detection is limited to be monitored based on the physical link, the indication of signal degrade is preferred at the transport path level, e.g. MPLS PWs, MPLS LSPs, or MPLS Sections. In this case, Generic Associated Channel (G-ACh) defined in [RFC5586] is proposed as the best choice to satisfy this requirement. The Generic Associated Channel packet format used in SR over MPLS network is shown in Figure 1.

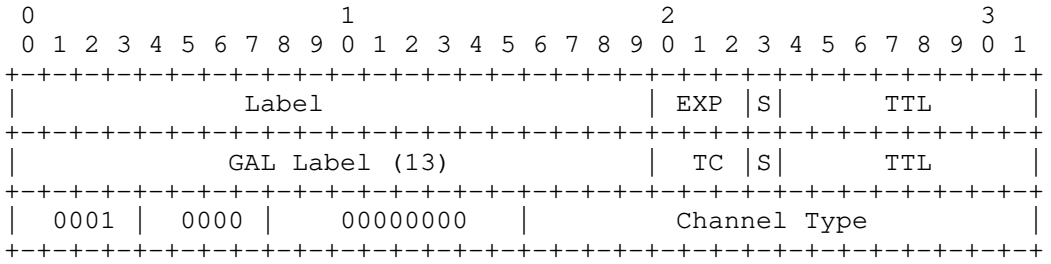


Figure 1 G-ACh Packet Format in SR over MPLS

## 4. BFD Indication Mechanism

Working together with the G-Ach, IP/UDP/BFD packet formats are encapsulated and shown in Figure 2. The IP, UDP and BFD headers stay intact within the generic associated channel. The Diagnostic code specifies the local system's reason for the last change in session state. The definition of the Values is specified in Section 4.1 of [RFC5880]. The Reserved values from 9 to 31 can be extended to support the signal degrade indication. The registration to support the indication and removal of the signal degrade indication should be applied to IANA.

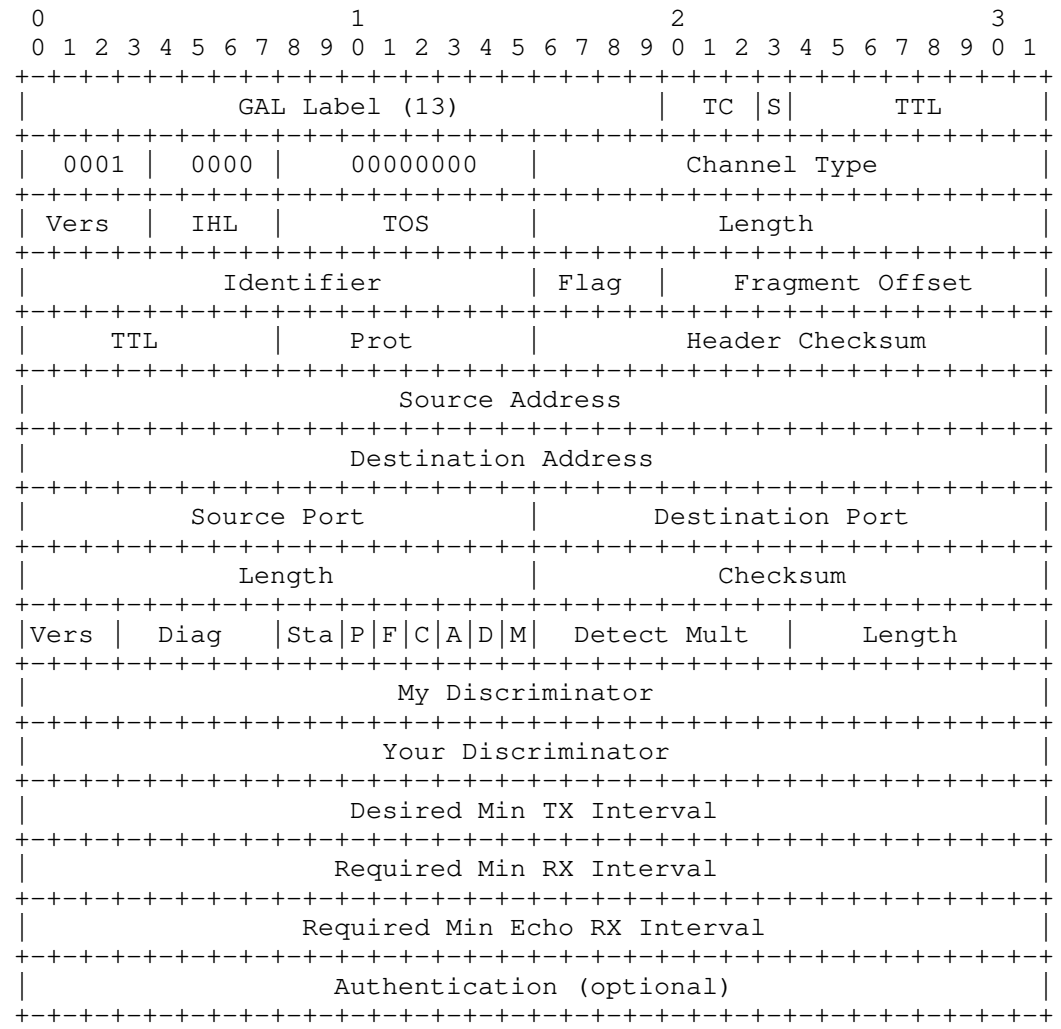


Figure 2: BFD Packet Format in SR over MPLS

## 5. MPLS-TP Indication Mechanism

ITU-T G.8113.1 defines the OAM PDU formats used in MPLS-TP networks. Figure 3 shows the OAM PDU format used within the SR over MPLS networks. If the LSR node detects the signal degrade, OAM CCM message is chosen to indicate the signal degrade via the forwarding plane. The OpCode value 0x01 in OAM PDU field indicates the CCM PDU message type.

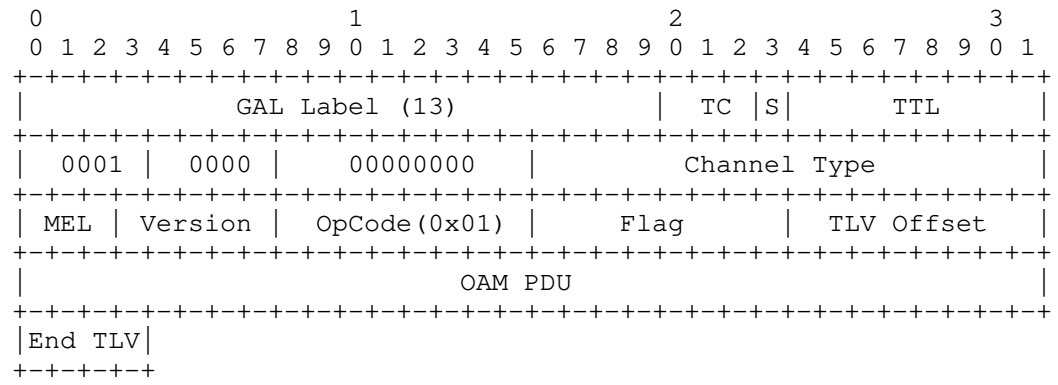


Figure 3 G.8113.1 OAM PDU Format in SR over MPLS

The reservation bits in Flag format in CCM OAM PDU message can be used as the error notification indication (EI) to indicate signal degrade, as shown in Figure 4. LSRs fills the EI field and transmits the OAM message to the other LSRs or LERs so that the degrade information can be learned.

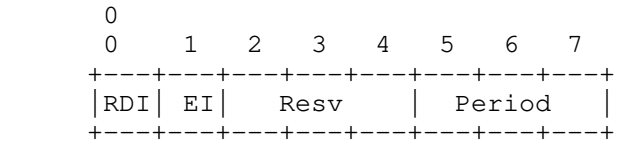


Figure 4 Extended Flags Format

EI (1 bit): Error notification indication, 0 indicates no error, 1 indicates error, to notify the signal degradation error.

## 6. IANA Considerations

The document requires the definition of the new indication and removal of the signal degrade indication in BFD Value code. Moreover, the EI bit definition is required to be assigned by ITU-T.

## 7. Security Considerations

This document has no security consideration.

## 8. Acknowledgements

TBD

# 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>.

## 9.2. Informative References

[I-D.yang-mpls-ps-sdi-sr] Yang, F., Han, L., and J. Zhao, "Problem Statement of Signal Degrade Indication for Segment Routing over MPLS Network", draft-yang-mpls-ps-sdi-sr-00 (work in progress), March 2020.

[ITU-T_G8113.1] ITU-T, "ITU-T G.8113.1: Operations, administration and maintenance mechanisms for MPLS-TP in packet and maintenance mechanisms for MPLS-TP in packet transport networks", April 2016.

[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>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

# Authors' Addresses

Liuyan Han
China Mobile
No.32 Xuanwumen west street
Beijing 100053
China

Email: hanliuyan@chinamobile.com

Fan Yang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: shirley.yangfan@huawei.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2021

K. Kompella
R. Balaji
R. Thomas
Juniper Networks
July 13, 2020

# Label Distribution Using ARP
# draft-kompella-mpls-larp-08

## Abstract

This document describes extensions to the Address Resolution Protocol to distribute MPLS labels for IPv4 and IPv6 host addresses. Distribution of labels via ARP enables simple plug-and-play operation of MPLS, which is key to deploying MPLS in data centers and enterprises.

## 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 14, 2021.

Table of Contents



## 1. Introduction

This document describes extensions to the Address Resolution Protocol (ARP) [RFC0826] to advertise label bindings for IP host addresses. While there are well-established protocols, such as LDP, RSVP and BGP, that provide robust mechanisms for label distribution, these protocols tend to be relatively complex, and often require detailed configuration for proper operation. There are situations where a simpler protocol may be more suitable from an operational standpoint. An example is the case where an MPLS Fabric is the underlay technology in a Data Center; here, MPLS tunnels originate from host machines. The host thus needs a mechanism to acquire label bindings to participate in the MPLS Fabric, but in a simple, plug-and-play manner. Existing signaling/routing protocols do not always meet this need. Labeled ARP (L-ARP) is a proposal to fill that gap.

### 1.1. Requirements Language

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

The term "server" will be used in this document to refer to an ARP/L-ARP server; the term "host" will be used to refer to a compute server or other device acting as an ARP/L-ARP client.

### 1.2. Approach

ARP is a nearly ubiquitous protocol; every device with an Ethernet interface, from hand-helds to hosts, have an implementation of ARP. ARP is plug-and-play; ARP clients do not need configuration to use ARP. That suggests that ARP may be a good fit for devices that want to source and sink MPLS tunnels, but do so in a zero-config, plug-and-play manner, with minimal impact to their code.

The approach taken here is to create a minor variant of the ARP protocol, labeled ARP (L-ARP), which is distinguished by a new hardware type, MPLS-over-Ethernet. Regular (Ethernet) ARP (E-ARP) and L-ARP can coexist; a device, as an ARP client, can choose to send out an E-ARP or an L-ARP request, depending on whether it needs Ethernet or MPLS connectivity. Another device may choose to function as an E-ARP server and/or an L-ARP server, depending on its ability to provide an IP-to-Ethernet and/or IP-to-MPLS mapping.

## 2. Overview of Ethernet ARP

In the most straightforward mode of operation [RFC0826], ARP queries are sent to resolve "directly connected" IP addresses. The ARP request is broadcast, with the Target Protocol Address field (see Section 5 for a description of the fields in an ARP message) carrying the IP address of another node in the same subnet. All the nodes in the LAN receive this ARP request. All the nodes, except the node that owns the IP address, ignore the ARP request. The IP address owner learns the MAC address of the sender from the Source Hardware Address field in the ARP request, and unicasts an ARP reply to the sender. The ARP reply carries the replying node's MAC address in the Source Hardware Address field, thus enabling two-way communication between the two nodes.

A variation of this scheme, known as "proxy ARP" [RFC2002], allows a node to respond to an ARP request with its own MAC address, even when the responding node does not own the requested IP address. Generally, the proxy ARP response is generated by routers to attract traffic for prefixes they can forward packets to. This scheme requires the host to send ARP queries for the IP address the host is trying to reach, rather than the IP address of the router. When there is more than one router connected to a network, proxy ARP enables a host to automatically select an exit router without running any routing protocol to determine IP reachability. Unlike regular ARP, a proxy ARP request can elicit multiple responses, e.g., when

more than one router has connectivity to the address being resolved. The sender must be prepared to select one of the responding routers.

Yet another variation of the ARP protocol, called 'Gratuitous ARP' [RFC2002], allows a node to update the ARP cache of other nodes in an unsolicited fashion. Gratuitous ARP is sent as either an ARP request or an ARP reply. In either case, the Source Protocol Address and Target Protocol Address contain the sender's address, and the Source Hardware Address is set to the sender's hardware address. In case of a gratuitous ARP reply, the Target Hardware Address is also set to the sender's address.

## 3. L-ARP Protocol Operation

The L-ARP protocol builds on the proxy ARP model, and also leverages gratuitous ARP model for asynchronous updates.

In this memo, we will refer to L-ARP clients (that make L-ARP requests) and L-ARP servers (that send L-ARP responses). In Figure 1, H1, H2 and H3 are L-ARP clients, and T1, T2 and T3 are L-ARP servers. T4 is a member of the MPLS Fabric that may not be an L-ARP server. Within the MPLS Fabric, the usual MPLS protocols (IGP, LDP, RSVP-TE) are run. Say H1, H2 and H3 want to establish MPLS tunnels to each other (for example, they are using BGP MPLS VPNs as the overlay virtual network technology). H1 might also want to talk to a member of the MPLS Fabric, say T. Also, the "protocol" addresses in L-ARP requests are either IPv4 or IPv6 addresses; note that while it is common to use Neighbor Discovery (ND) [RFC4861] for "regular" ARP requests when dealing with IPv6 (i.e., to obtain Ethernet MAC addresses corresponding to an IPv6 address), ND is not used when the ARP request is for an MPLS label.

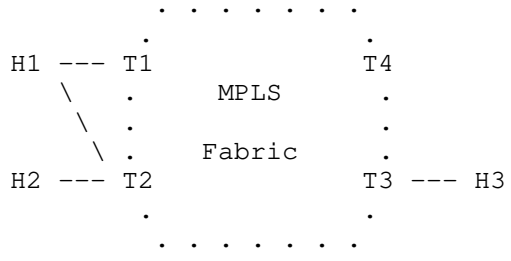


Figure 1: MPLS Fabric

### 3.1. Setup

In Figure 1, the nodes T1-T4, and those in between making up the "MPLS Fabric" are assumed to be running some protocol whereby they can signal MPLS reachability to themselves and to other nodes (like hosts H1-H3). T1-T3 are L-ARP servers; T4 need not be, since it doesn't have an attached L-ARP client. H1-H3 are L-ARP clients.

### 3.2. Egress Operation

A node (say T3) that wants an attached node (say H3) to have MPLS reachability allocates a label L3 to reach H3 and advertises this label into the MPLS Fabric. This can be triggered by configuration on T3, or when T3 first receives an L-ARP request from H3 (indicating that H3 wants MPLS connectivity), or via some other protocol. T3 then advertises (H3, L3) to its peers in the MPLS Fabric so that all members of the Fabric have connectivity to H3. This advertisement can be one of the following:

- a "proxy" LDP message (sent on behalf of H3) with prefix H3 and label L3; or
- a node SID advertised on behalf of H3; or
- a labeled BGP advertisement, with prefix H3, label L3 and next hop self.

  On receiving a packet with label L3, T3 pops the label and send the packet to H3. (In the case of labeled BGP, there would be a two-label stack, with outer label to reach T3 and inner label of L3.) This is the usual operation of an MPLS Fabric, with the addition of advertising labels for nodes outside the fabric.

### 3.3. Ingress Operation

A node (say H1, an L-ARP client) that needs an MPLS tunnel to another node (say H3) identified by a host address (either IPv4 or IPv6) broadcasts over all its interfaces an L-ARP request with the Target Protocol Address set to H3 and Hardware Type set to "MPLS-over-Ethernet". A node receiving the L-ARP request (say T1, an L-ARP server) does the following:

1. checks if it has reachability to H3. If not, it ignores the L-ARP request.
2. if it does, T1 allocates a label TL3 to reach H3 (if it doesn't already have such a label) and installs an L-FIB entry to swap L1 with the label (stack) to reach H3.

3. sends a (proxy) L-ARP reply to H1 with the Source Hardware Address (SHA) set to (L, M), where M is T1's metric to H3. T1 may also set some attribute bits in the SHA.

### 3.4. Data Plane

To send a packet to H3 over an MPLS tunnel, H1 pushes L1 onto the packet, sets the destination MAC address to M1 and sends it to T1. On receiving this packet, T1 swaps the top label with the label(s) for its MPLS tunnel to H3. If T1's reachability to H3 is via a SPRING label stack, the label L1 acts as an implicit binding SID.

If H1 and H3 have an overlay connection (say an IPVPN [RFC4364] VPN-foo) whereby VM1 on H1 wishes to talk to VM3 on H3 over VPN-foo, H1 does the following:

1. H1 learns information about VPN-foo via BGP (or an SDN controller), including the VPN label VL3 to use to talk to VM3;

2. H1 installs a VRF for VPN-foo, with prefix VM3, label VL3 and next hop H3;

3. H1 binds the local "veth" interface to VM1 to this VRF.

4. When VM1 sends a packet to dest IP address VM3 over its veth interface, H1 looks up VM3 in the corresponding VRF, gets label VL3. It then sends an L-ARP request for next hop H3, and gets TL3.

5. Finally, H1 pushes the label pair (TL3, VL3) onto the packet from VM1 and sends this to T1. This packet will then end up at VM3 on H3.

Note that H1 broadcasts its L-ARP request over its attached interfaces. H1 may receive several L-ARP replies; in that case, H1 can select any subset of these to send MPLS packets destined to H3. As described later, the L-ARP response may contain certain parameters that enable the client to make an informed choice. If the target H3 belongs to one of the subnets that H1 participates in, and H3 is capable of sending L-ARP replies, H1 can use H3's response to send MPLS packets to H3.

## 4. Attributes

In addition to carrying a label stack to be used in the data plane, an L-ARP reply carries some attributes that are typically used in the control plane. One of these is a metric. The metric is the distance from the L-ARP server to the destination. This allows an L-ARP

client that receives multiple responses to decide which ones to use, and whether to load-balance across some of them. The metric typically will be the IGP shortest path distance from server to the destination; this makes comparing metrics from different servers meaningful.

Another attribute is Entropy Label (EL) Capability. This attribute says whether the destination is EL capable (ELC). In Figure 1, if T3 advertises a label to reach H3 and T3 is ELC, T3 can include in its signaling to T1 that it is ELC. In that case, T1's L-ARP reply to H1 can have ELC bit set. This tells H1 that it may include (below the outermost label) an Entropy Label Indicator followed by an Entropy Label. This will help improve load balancing across the MPLS Fabric, and possibly on the last hop to H3.

### 4.1. Secondary Attributes

Beyond the basic attributes that are carried with every L-ARP request, there more optional attributes, for example, to ask for certain characteristics of the path traffic takes to the destination. These attributes are carried in TLVs that are carried in L-ARP requests and replies.

One such TLV is the "CT" TLV. This TLV allows the L-ARP client to request a label to a destination over a tunnel in the Transport Class given by CT [I-D.kaliraj-idr-bgp-classful-transport-planes]. To satisfy this request, the L-ARP server creates (or finds) a tunnel to the destination that is routed over the CT Transport Plane, allocates a label L, inserts an entry in the LFIB to swap L to the tunnel, and sends L to the L-ARP client in its reply.

## 5. L-ARP Message Format

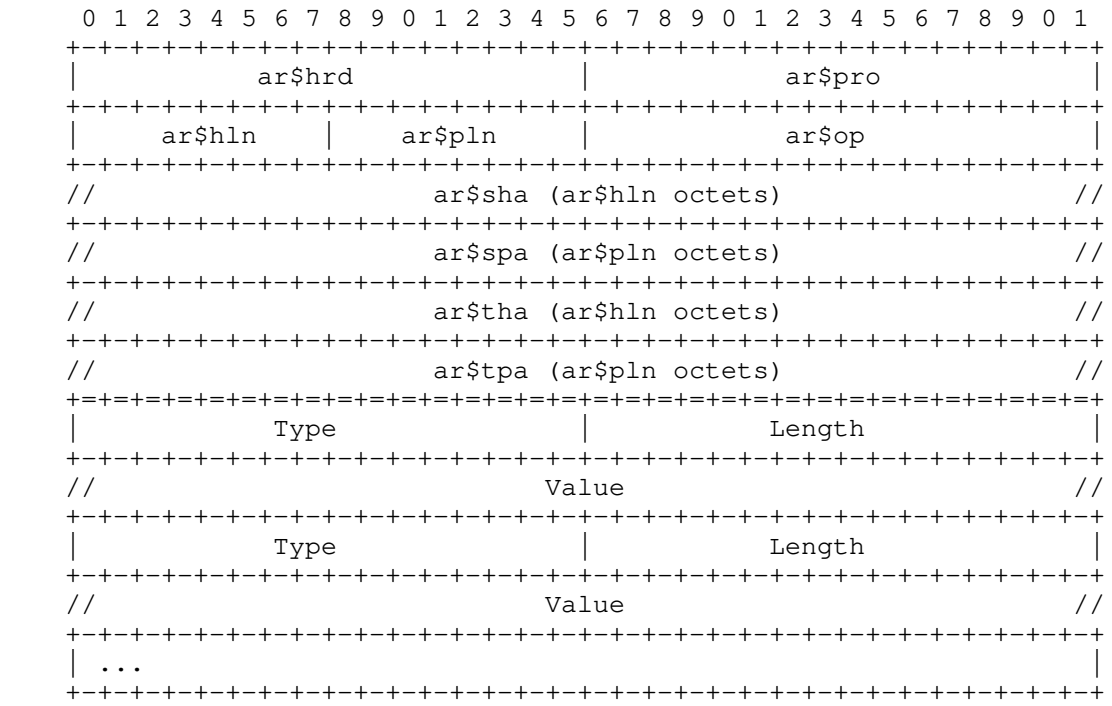


Figure 2: L-ARP Packet Format

ar$hrd: Hardware Type: MPLS-over-Ethernet. The value of the field used here is HTYPE-MPLS. To start with, we will use the experimental value HW_EXP2 (256).

ar$pro: Protocol Type: IPv4/IPv6. The value of the field used here is 0x0800 to resolve an IPv4 address and 0x86DD to resolve an IPv6 address.

ar$hln: Hardware Address Length: 6

ar$pln: Protocol Address Length: for an IPv4 address, the length is 4 octets; for an IPv6 address, it is 16.

ar$op: Operation Code: set to 1 for request, 2 for reply, and 10 for ARP-NAK. Other op codes may be used as needed.

ar$sha: Source Hardware Address: In an L-ARP request, this is usually all zeros. In an L-ARP reply, Source Hardware Address is the label to reach ar$spa, as specified in Figure 3 below.

ar$spa: Source Protocol Address: In an L-ARP request, this field carries the sender's IP address. In an L-ARP reply, this field carries the requested IP address (which may not be the sender's IP address).

ar$tha: Target Hardware Address: In an L-ARP message, this is all zeros.

ar$tpa: Target Protocol Address: In an L-ARP request, this field carries the IP address for which the client is seeking an MPLS label.

Type: a 2-octet field defining the Type of the TVL

Length: a 2-octet field defining the Length L of the TVL

Value: an L-octet field with the Value of the TLV

## 5.1. Hardware Address Format

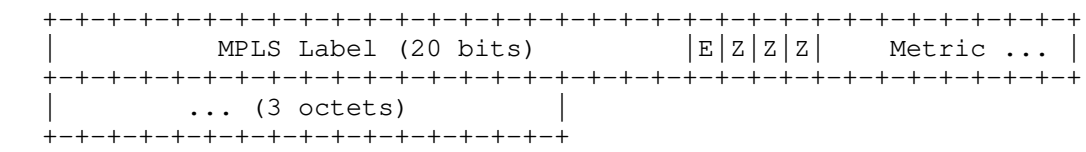


Figure 3: Label Format in L-ARP

MPLS Label: The 20-bit label

E-bit: Entropy Label Capable: this flag indicates whether the corresponding label in the label stack can be followd by an Entropy Label. If this flag is set, the client has the option of inserting ELI and EL as specified in [RFC6790]. The client can choose not to insert ELI/EL pair. If this flag is clear, the client must not insert ELI/EL after the corresponding label.

Z: These bits are not used, and SHOULD be set to zero on sending and ignored on receipt.

Metric: The IGP metric to ar$tha from the point of view of the L-ARP replier.

### 5.2. CT TLV

The CT TLV has Type (TBD) and Length 4 octets; the Value field consists of the CT attribute.

## 6. Security Considerations

There are many possible attacks on ARP: ARP spoofing, ARP cache poisoning and ARP poison routing, to name a few. These attacks use gratuitous ARP as the underlying mechanism, a mechanism used by L-ARP. Thus, these types of attacks are applicable to L-ARP. Furthermore, ARP does not have built-in security mechanisms; defenses rely on means external to the protocol.

It is well outside the scope of this document to present a general solution to the ARP security problem. One simple answer is to add a TLV that contains a digital signature of the contents of the ARP message. This TLV would be defined for use only in L-ARP messages, although in principle, other ARP messages could use it as well. Such an approach would, of course, need a review and approval by the Security Directorate. If approved, the type of this TLV and its procedures would be defined in this document. If some other technique is suggested, the authors would be happy to include the relevant text in this document, and refer to some other document for the full solution.

## 7. IANA Considerations

IANA is requested to allocate a new ARP hardware type (from registry hrd) for HTYPE-MPLS.

## 8. Acknowledgments

Many thanks to Shane Amante for his detailed comments and suggestions. Many thanks to the team in Juniper prototyping this work for their suggestions on making this variant workable in the context of existing ARP implementations. Thanks too to Luyuan Fang, Alex Semenyaka and Dmitry Afanasiev for their comments and encouragement.

## 9. References

### 9.1. Normative References

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC2002] Perkins, C., Ed., "IP Mobility Support", RFC 2002, DOI 10.17487/RFC2002, October 1996, <https://www.rfc-editor.org/info/rfc2002>.

[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>.

## 9.2. Informative References

[I-D.kaliraj-idr-bgp-classful-transport-planes] Vairavakkalai, K., Venkataraman, N., and B. Rajagopalan, "BGP Classful Transport Planes", draft-kaliraj-idr-bgp-classful-transport-planes-00 (work in progress), May 2020.

[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>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

## Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale 94089
USA

Phone: +1-408-745-2000
Email: kireeti.kompella@gmail.com

Balaji Rajagopalan
Juniper Networks
Survey No.111/1 to 115/4, Wing A & B
Bangalore 560103
India

Email: balajir@juniper.net

Reji Thomas
Juniper Networks
Survey No.111/1 to 115/4, Wing A & B
Bangalore 560103
India

Email: rejithomas@juniper.net

MPLS WG
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2020

K. Kompella
W. Lin
Juniper Networks

March 08, 2020

# No Further Fast Reroute
# draft-kompella-mpls-nffrr-00

## Abstract

There are several cases where, once Fast Reroute has taken place (for MPLS protection), a second fast reroute is undesirable, even detrimental. This memo gives several examples of this, and proposes a mechanism to prevent further fast reroutes.

## 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 9, 2020.

Table of Contents



## 1. Introduction

MPLS Fast Reroute (FRR) [RFC4090] [RFC5286] [RFC7490] is a useful and widely deployed tool for minimizing packet loss in the case of a link or node failure. This has not only proven to be very effective, it is often the reason for using MPLS as a data plane. FRR works for a variety of control plane protocols, including LDP, RSVP-TE, and SPRING. Furthermore, FRR is often used to protect MPLS services such as IP VPN and EVPN.

Having said this, there are case where, once FRR has taken place, if the packet encounters a second failure, a second FRR is not helpful, perhaps even disruptive. For example, the packet may loop until TTL expires. This can lead to link congestion and further packet loss. Thus, the attempt to prevent a packet from being dropped may instead affect many other packets. Note that the "second" failure may simply be another manifestation of the same failure; see Figure 1.

This memo proposes a mechanism for preventing further FRR once in cases where such further protection may be harmful. Several examples where this is the case are demonstrated as motivation. A solution using special-purpose labels (SPLs) is then offered. Some mechanisms

for distributing the capability to avoid further fast reroutes are also discussed, although these may be better placed in other documents in other Working Groups.

### 1.1. Terminology

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

## 2. Motivation

A few cases are given where "further fast reroute" is harmful. Some of the cases are for MPLS services; others for "plain" MPLS forwarding.

### 2.1. EVPN (VPN/VPLS) Active-active Multihoming

Consider the following topology for multihoming an Ethernet VPN (EVPN [RFC7432]) Customer Edge (CE) device for protection against the failure of a Provider Edge (PE) device or a PE-CE link. To do so, there is a backup MPLS path between PE2 and PE3 (denoted by the starred line).

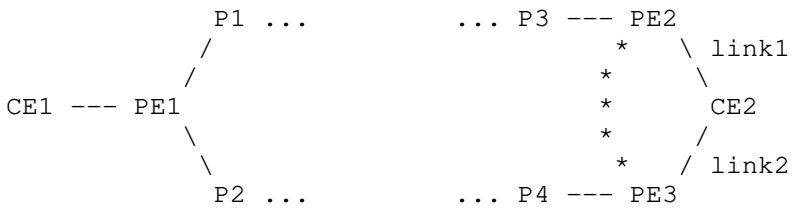


Figure 1: EVPN Multihoming

Suppose (known unicast) traffic goes from CE1 to CE2. With active-active multihoming, this traffic will be load-balanced between PE2 (to CE2 via link link1) and PE3 (to CE2 via link2). If link1 were to fail, PE2 can still get traffic for CE2 by sending it over the backup path to PE3 (and similarly for PE3 if link2 fails).

However, suppose CE2 is down. PE2 will assume link1 is down and send traffic for CE2 to PE3 over the backup path. PE3 (which thinks that link2 is down; note that the single real failure of CE2 being down is manifested as separate failures to PE2 and PE3) will protect this "second" failure by sending traffic for CE2 over the backup path to

PE2. Thus, traffic will ping-pong between PE2 and PE3 until TTL expires.

Thus, the attempt to protect traffic to CE2 may end up doing more harm than good, by congesting the backup path between PE2 and PE3 and by giving PE2 and PE3 useless work to do.

A similar topology can be used in EVPN-Etree [RFC8317], EVPN-VPWS [RFC8214], IP VPN [RFC4364] or VPLS [RFC4761] [RFC4762]. In all these cases, the same looping behavior would occur for unicast traffic if CE2 is down.

## 2.2. RMR Protection

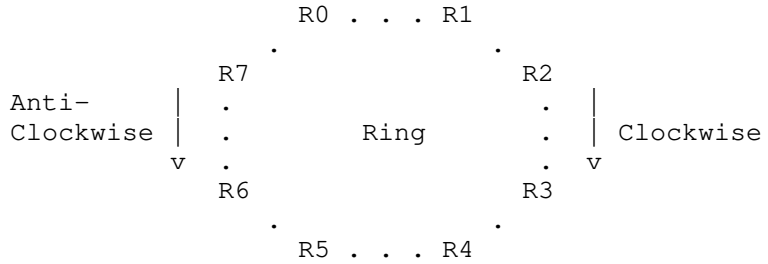


Figure 2: RMR Looping

In Resilient MPLS Rings (RMR), suppose traffic goes from a node, say R0, to a node, say R4, over a clockwise path. Protection consists of switching this traffic onto the anti-clockwise path to R4. This works well if a node or link between R0 or R4 is down. However, if node R4 itself is down, its adjacent neighbor R3, will send the traffic anti-clockwise to R4; when this traffic reaches R4's other neighbor R5, it will return to N3, and so on, until TTL expires. [I-D.ietf-mpls-rmr] provides more details, and offers some means of mitigation. This memo offers a more elegant solution.

## 2.3. General MPLS forwarding

Consider the following topology:

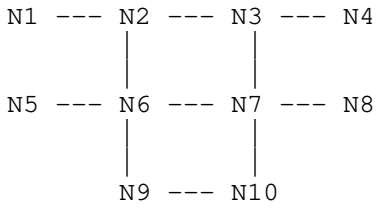


Figure 3: General MPLS Forwarding

Say link protection is configured for links N2-N3 and N6-N7. Link N2-N3 is protected by a bypass tunnel N2-N6-N7-N3, and link N7-N3 is protected by a bypass tunnel N7-N6-N2-N3. (These bypass tunnels may be set up using RSVP-TE [RFC3209] or via SPRING stacks [RFC8660].) Say furthermore that there is an LSP from N1 to N4 with path N1-N2-N3-N4, which asks for link protection. If link N2-N3 fails, traffic will take the path N1-N2-N6-N7-N3-N4.

Suppose, however, links N2-N3 and N7-N3 fail simultaneously. This may happen if they share fate (e.g., go over a common fiber conduit); it may also appear to happen if node N3 fails. Either way, first, the bypass protecting link N2-N3 kicks in, and traffic is sent to N3 via N6 and N7. However, when the traffic hits N7, the bypass for N7-N3 kicks in, and traffic is sent back to N2. Thus the traffic will loop between N2 and N7 until TTL expires, in the process congesting links N2-N6 and N6-N7.

Now consider an LSP: N5-N6-N7-N8. The link N6-N7 may be protected by the bypass N6-N2-N3-N7 or by N6-N9-N10-N7, or by load-balancing between these two bypasses. If both links N2-N3 and N6-N7 fail, then traffic that is protected via bypass N6-N2-N3-N7 will ping-pong between N6 and N2 until TTL expires; traffic protected via bypass N6-N9-N10-N7 will successfully make it to N8. If link N6-N7 is protected by load-balancing across the two bypass paths, then about half the traffic will loop between N6 and N2, and the rest will make it to N8.

While the above description is for protection using a bypass tunnel, the same principle applies to protection using Loop-Free Alternates [RFC5286] [RFC7490] or any of its variants (such as Topology Independent LFA).

## 3. Solution

To address this issue, we suggest the use of a SPL [RFC7274] called NFFRR (value TBD; suggested: 8). An alternate would be to use an extended SPL, whereby a pair of labels indicates that no further fast route is desired. However, in the case of SPRING MPLS bypass tunnels

(Section 3.2.1) of depth N, this would triple the label stack size. Using regular SPLs instead would only double the stack size.

## 3.1. NFFRR for MPLS forwarding

To illustrate, we'll first take the example of Figure 3, with MPLS paths signaled using RSVP-TE. This method can be used for paths that use SPRING stacks, but this will be detailed in a later version.

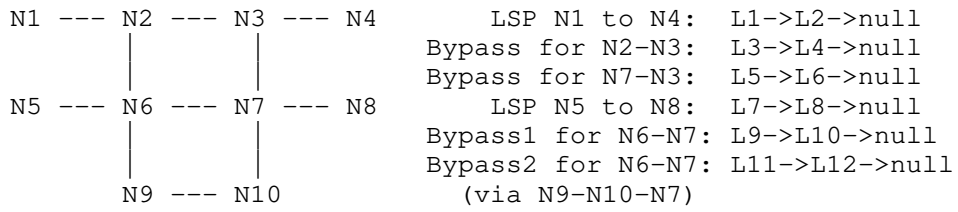


Figure 4: Example Using RSVP-TE LSPs

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N1 | push L1 | N2 | [L1] pkt | ingress |
| N2 | L1 -> L2 | N3 | [L2] pkt | |
| N3 | pop L2 | N4 | pkt | PHP |
| N4 | fwd pkt | - | - | continue |

Table 1: Forwarding from N1 to N4

Note 1: "[L1 ...]" denotes the label stack on the packet; pkt is the original packet received at ingress. "L1 -> L2" means swap label L1 with L2. "pop L2" means pop the top label L2. "fwd pkt" means forward the packet as usual.

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N2 | push L3 | N6 | [L3] pkt | ingress |
| N6 | L3 -> L4 | N7 | [L4] pkt | |
| N7 | pop L4 | N3 | pkt | PHP |

Table 2: Forwarding over the bypass for link N2-N3

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N7 | push L5 | N6 | [L5] pkt | ingress |
| N6 | L5 -> L6 | N2 | [L6] pkt | |
| N2 | pop L6 | N3 | pkt | PHP |

Table 3: Forwarding over Bypass1 for link N7-N3

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N1 | push L1 | N2 | [L1] pkt | ingress |
| N2 | L1 -> L2 | N3 | [L2] pkt | N3 X |
| N2 | push L3 | N6 | [L3 L2] pkt | PLR |
| N6 | L3 -> L4 | N7 | [L4 L2] pkt | |
| N7 | pop L4 | N3 | [L2] pkt | merge |
| N3 | pop L2 | N4 | pkt | PHP |
| N4 | fwd pkt | - | - | continue |

Table 4: Forwarding from N1 to N4 if link N2-N3 fails

Table 4 is obtained by composing Table 1 and Table 2.

Note 2: "N3 X" means "next hop N3 unavailable (because link N2-N3 failed)".

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N1 | push L1 | N2 | [L1] pkt | ingress |
| N2 | L1 -> L2 | N3 | [L2] pkt | N3 X |
| N2 | push L3 | N6 | [L3 L2] pkt | PLR |
| N6 | L3 -> L4 | N7 | [L4 L2] pkt | |
| N7 | pop L4 | N3 | [L2] pkt | N3 X’ |
| N7 | push L5 | N6 | [L5 L2] pkt | |
| N6 | L5 -> L6 | N2 | [L6 L2] pkt | PLR |
| N2 | pop L6 | N3 | [L2] pkt | N3 X |
| N2 | push L3 | N6 | [L3 L2] | PLR |
| etc | | | | loop! |

Table 5: Forwarding from N1 to N4 if links N2-N3 and N7-N3 fail

Table 5 is obtained by composing Table 1, Table 2 and Table 3.

Note 3: "N3 X’" means "next hop N3 unavailable because link N7-N3 is down.

Note 4: While the impact of a loop is pretty bad, the impact of an ever-growing label stack (not illustrated here) and possible associated fragmentation on transit nodes may be worse.

## 3.2. Proposal

An LSR (typically a PLR) that wishes to prevent further FRRs after the first one can push an SPL, namely NFFRR, onto the label stack as follows:

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N1 | push L1 | N2 | [L1] pkt | ingress |
| N2 | L1 -> L2 | N3 | [L2] pkt | N3 X |
| N2 | push L3, NFFRR | N6 | [L3 NFFRR L2] pkt | PLR |
| N6 | L3 -> L4 | N7 | [L4 NFFRR L2] pkt | |
| N7 | pop L4, NFFRR | N3 | [L2] pkt | merge |
| N3 | pop L2 | N4 | pkt | PHP |
| N4 | fwd pkt | - | - | continue |

Table 6: Forwarding from N1 to N4 if link N2-N3 fails with NFFRR

Note 5: N2 can insert an NFFRR label only if it knows that all LSRs in the path can process it correctly. See Section 4 for some details on how this capability is communicated.

| Node | Action | Next | New Pkt | Comment |
|---|---|---|---|---|
| N1 | push L1 | N2 | [L1] pkt | ingress |
| N2 | L1 -> L2 | N3 | [L2] pkt | N3 X |
| N2 | push L3, NFFRR | N6 | [L3 NFFRR L2] pkt | PLR |
| N6 | L3 -> L4 | N7 | [L4 NFFRR L2] pkt | |
| N7 | pop L4 | N3 | [NFFRR L2] pkt | N3 X |
| N7 | check NFFRR | - | - | drop pkt |

Table 7: Forwarding from N1 to N4 if links N2-N3 and N7-N3 fail with NFFRR

Note 6: "check NFFRR" means that, before N7 applies FRR (because link N7-N3 is down), N7 checks the label below the top label (or in this case, because of PHP, the top label itself). If this is the NFFRR label, N7 drops the packet rather than apply FRR.

#### 3.2.1. NFFRR and SPRING

Suppose that, to protect link N2-N3, a bypass tunnel N2-N6-N7-N3 were instantiated using SPRING MPLS [RFC8660], in particular, using adjacency SIDs. If the corresponding labels for links N6-N7 and N7-N3 were L20 and L21, the bypass would consist of pushing the label stack [L20 L21] onto the packet and sending the packet to N6. To indicate that FRR has already occurred and to drop the packet rather than to try to protect the packet again, N2 would have to push [L20 NFFRR L21 NFFRR] onto the packet before sending it to N6. If the packet came from N1 with label L1, N2 would send a packet with label stack [L20 NFFRR L21 NFFRR L2] to N6.

N6 would see L20, pop it, note the NFFRR label and pop it, then attempt to send the packet to N7. If the link N6-N7 is down, N6 drops the packet. Otherwise, N7 gets the packet, sees L21, pops it, sees NFFRR, pops it and tries to send the packet to N3. If link N7-N3 is down, N7 drops the packet. Otherwise, N3 gets the packet with L2, swaps with with L3 and sends it to N4.

Note that with SPRING MPLS, the NFFRR label needs to be repeated for each label in the bypass stack. Hence the request for a "regular" SPL rather than an extended SPL.

### 3.3. NFFRR for MPLS Services

First, we illustrate known unicast EVPN forwarding:

| Node | Action | Next | Packet | Comment |
|---|---|---|---|---|
| PE1 | send to CE2 | PE2 | [T1 S2] pkt | EVPN |
| PE2 | send to CE2 | link1 | pkt | done! |

Note: T1/T2/T3 are the transport labels for PE1/PE3/PE2 to reach PE2/PE2/PE3 respectively. S2/S3 are the service labels announced by PE2/PE3 for CE2.

Then, we show what happens when CE2 is down without NFFRR:

| Node | Action | Next | Packet | Comment |
|---|---|---|---|---|
| PE1 | send to CE2 | PE2 | [T1 S2] pkt | EVPN |
| PE2 | send to CE2 | link1 | -- | link1 X |
| PE2 | send to CE2 | PE3 | [T3 S3] pkt | eFRR |
| PE3 | send to CE2 | link2 | -- | link2 X |
| PE3 | send to CE2 | PE2 | [T2 S2] pkt | eFRR |
| PE2 | send to CE2 | link1 | -- | link1 X |
| PE2 | send to CE2 | PE3 | [T3 S3] pkt | eFRR |
| ... | | | | loop! |

Note: link1/link2 X means link1/link2 is down. eFRR refers to EVPN multihoming FRR.

In the case of MPLS services such as EVPN Figure 1, the NFFRR label is inserted below the service label, as shown below:

| Node | Action | Next | Packet | Comment |
|---|---|---|---|---|
| PE1 | send to CE2 | PE2 | [T1 S2] pkt | EVPN |
| PE2 | send to CE2 | link1 | -- | link1 X |
| PE2 | send to CE2 | PE3 | [T3 S2 NFFRR] pkt | eFRR |
| PE3 | send to CE2 | link2 | -- | link2 X |
| PE3 | drop pkt | -- | -- | check NFFRR |

Note: "check NFFRR" is as above.

### 3.4. NFFRR for RMR

As described in Figure 2, packets will loop until TTL expires if the destination node in an RMR ring (here, R4) fails. The solution in this case is that the first node to apply RMR protection (R3) pops the current RMR transport label being used, sees that the next label

is not NFFRR (so protection is allowed), pushes an NFFRR label and then the RMR transport label for the reverse direction.

When R5 receives the packet, it sees that the next link is down, pops the RMR transport label, sees the NFFRR label and drops the packet. Thus, the loop is avoided.

# 4. Signaling NFFRR Capability

## 4.1. Signaling NFFRR Capability for MPLS Services with BGP

The ideal choice would be an attribute consisting of a bit vector of node capabilities, one bit of which would be the capability of processing the NFFRR SPL below the BGP service label. This would be used by BGP L2VPN, BGP VPLS, EVPN, E-Tree and E-VPWS. An alternative is to use the BGP Capabilities Optional Parameter [I-D.ietf-idr-next-hop-capability]. Details to be worked out.

## 4.2. Signaling NFFRR Capability for MPLS Services with Targeted LDP

One approach to signaling NFFRR capability for MPLS services signaled with targeted LDP is to introduce a new LDP TLV called the NFFRR Capability TLV as an Optional Parameter in the Label Mapping Message [RFC5036]. This TLV has Type TBD (suggested: 0x0207) and Length 0.

Another approach is to use LDP Capabilities [RFC5561]; this approach has the advantage that it deals with capabilities on a node basis rather than on a per label mapping basis. However, there don't appear to be other documents using this approach.

## 4.3. Signaling NFFRR Capability for MPLS Forwarding

The authors suggest signaling a router's ability to process the NFFRR SPL using the Link State Router TE Node Capabilities [RFC5073], which works for both IS-IS and OSPF. A new TE Node Capability bit, the N bit (suggested value 5) indicates that the advertising node is capable of processing the NFFRR SPL.

# 5. IANA Considerations

If this draft is deemed useful, an SPL for NFFRR will need to be allocated. We suggest the early allocation of label 8 for this.

Furthermore, means of signaling the ability to process the NFFRR SPL should be defined for IS-IS, OSPF, LDP and BGP.

The following update is suggested for the Link State Router TE Node Capabilities registry:

| Bit | Name | Reference |
|---|---|---|
| 5 | NFFRR | This docusment |

The following update is suggested for the TLV Type Name Space of the Label Distribution Protocol (LDP) Parameters registry:

| Type | Name | Reference |
|---|---|---|
| 0x0207 | NFFRR | This docusment |

## 6. Security Considerations

A malicious or compromised LSR can insert NFFRR into a label stack, preventing FRR from occurring. If so, protection will not kick in for failures that could have been protected, and there will be unnecessary packet loss.

## 7. References

### 7.1. Normative References

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

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[RFC5073] Vasseur, J., Ed. and J. Le Roux, Ed., "IGP Routing Protocol Extensions for Discovery of Traffic Engineering Node Capabilities", RFC 5073, DOI 10.17487/RFC5073, December 2007, <https://www.rfc-editor.org/info/rfc5073>.

[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>.

### 7.2. Informative References

[I-D.ietf-idr-next-hop-capability]
Decraene, B., Kompella, K., and W. Henderickx, "BGP Next-Hop dependent capabilities", draft-ietf-idr-next-hop-capability-05 (work in progress), June 2019.

[I-D.ietf-mpls-rmr]
Kompella, K. and L. Contreras, "Resilient MPLS Rings", draft-ietf-mpls-rmr-12 (work in progress), October 2019.

[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>.

[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>.

[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>.

[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>.

[RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, DOI 10.17487/RFC5561, July 2009, <https://www.rfc-editor.org/info/rfc5561>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[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>.

[RFC8214] Boutros, S., Sajassi, A., Salam, S., Drake, J., and J. Rabadan, "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, DOI 10.17487/RFC8214, August 2017, <https://www.rfc-editor.org/info/rfc8214>.

[RFC8317] Sajassi, A., Ed., Salam, S., Drake, J., Uttaro, J., Boutros, S., and J. Rabadan, "Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) and Provider Backbone Bridging EVPN (PBB-EVPN)", RFC 8317, DOI 10.17487/RFC8317, January 2018, <https://www.rfc-editor.org/info/rfc8317>.

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

## Authors' Addresses

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

Email: kireeti.kompella@gmail.com

Wen Lin
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States

Email: wlin@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2021

Y. Liu
G. Mirsky
ZTE Corporation

July 12, 2020

# MPLS-based Service Function Path(SFP) Consistency Verification
# draft-lm-mpls-sfc-path-verification-00

## Abstract

This document proposes a method to verify the correlation between Service Function Chaining control and/or management plane view of the specified Service Function Path and the state of its data. It works for both SR service programming and MPLS-based NSH SFC.

## 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 13, 2021.

Table of Contents



## 1. Introduction

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

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

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

## 2. Conventions used in this document

### 2.1. Acronyms

SFC: Service Function Chain

SFF: Service Function Forwarder

SF: Service Function

SFP: Service Function Path

RSP: Rendered Service Path

## 3. MPLS-based SFP Consistency Verification

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

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

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

- [RFC8595] provides another method to realize SFC in an MPLS network by means of using a logical representation of the Network Service Header (NSH) in an MPLS label stack. When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used, which can be present one or more times in the label stack. This unit comprises two MPLS labels, one carries a label to provide a context within the SFC scope (the SFC Context Label), and the other carries a label to show which SF is to be enacted (the SF Label). This two-label unit is shown in Figure 1.

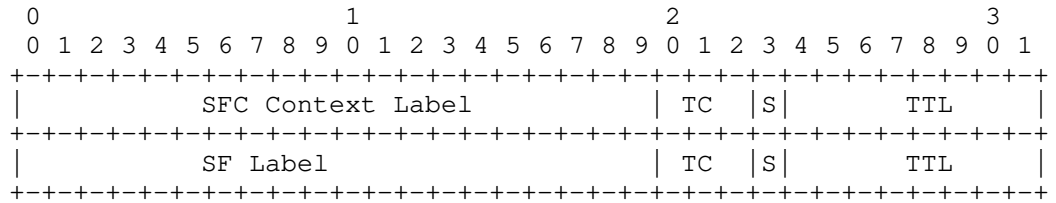


Figure 1: The Basic Unit of MPLS Label Stack for SFC

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

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

Generally, except for the designed specific functions, the packet processing functions supported by SFs are limited. SFs may not support MPLS OAM protocols like LSP ping, so SFFs are responsible for MPLS echo request processing.

### 3.1. MPLS-based SFP Consistency Verification

An MPLS SFC validation request/reply is an MPLS echo request/reply that includes an SFC validation TLV.

Nodes examine and process the TLV only if configured to do so; other nodes MUST ignore the TLV and process the packet as a standard MPLS echo packet.

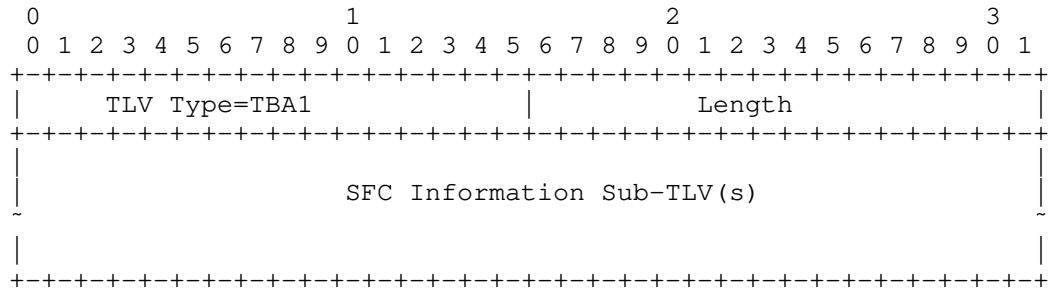


Figure 2: SFC Validation TLV

SFC Information Sub-TLV: The Sub-TLV, as defined in Figure 3, MUST NOT be included in an MPLS SFC validation request.

### 3.2. SFC Info Sub-TLV

Upon receiving the SFC validation request, the SFF MUST respond with an echo reply, which includes the SFC detailed information.

The SFC detailed information is recorded in SFC info sub-TLV.

Two types of sub-TLVs are defined in this section, and those are used in MPLS-based service programming [I-D.ietf-spring-sr-service-programming] and MPLS-based NSH [RFC8595] respectively.

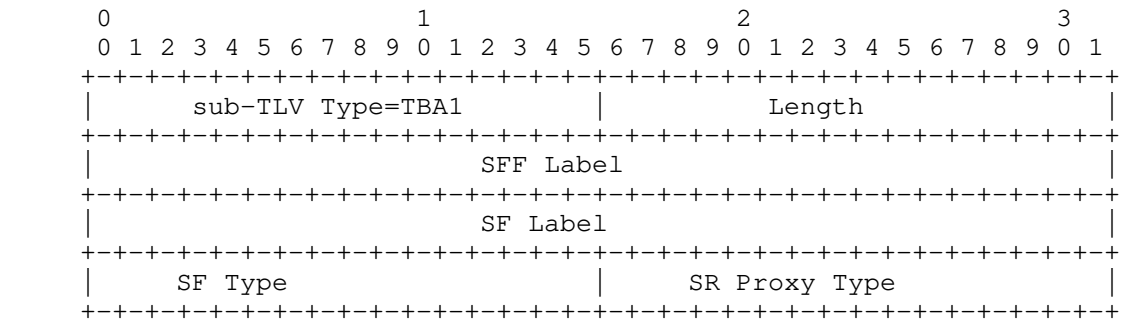


Figure 3: SFC Info Sub-TLV for SR-MPLS-based Service Programming

Type TBA1 sub-TLV: used in SR-MPLS-based service programming

SFF Label: represents the SID of the SFF

SF Label: represents the service SID of the SF or SR proxy

SF Type: indicates the type of SF, such as DPI, firewall, etc.

SR Proxy Type: It is defined in [I-D.ietf-spring-sr-service-programming] and indicates the type of SR proxy if it exists.

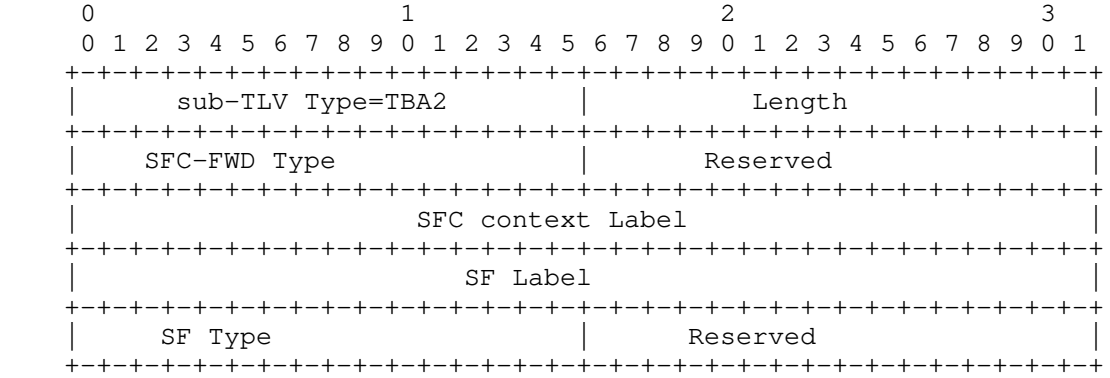


Figure 4: SFC Info Sub-TLV for MPLS-based NSH

Type TBA2 sub-TLV: used in MPLS-based NS

SFC-FWD Type: indicates the forwarding type of the data plane, and has the following values:

0x10: MPLS-based NSH [RFC8595] label swapping

0x11: MPLS-based NSH [RFC8595] label stacking

SFC context Label: The meaning of the SFC context label depends on the SFC Type. If SFC-FWD Type is 0x10, the SFC context Label represents SPI. If SFC-FWD Type is 0x11, the SFC context Label represents the context label [RFC8595].

SF Label: The meaning of the SF label depends on the SFC-FWD Type. If SFC Type is 0x10, the SF Label represents SI. If SFC Type is 0x11, the SF Label represents the SFI index [RFC8595].

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

## 3.3. SFC Basic Unit FEC Sub-TLV

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

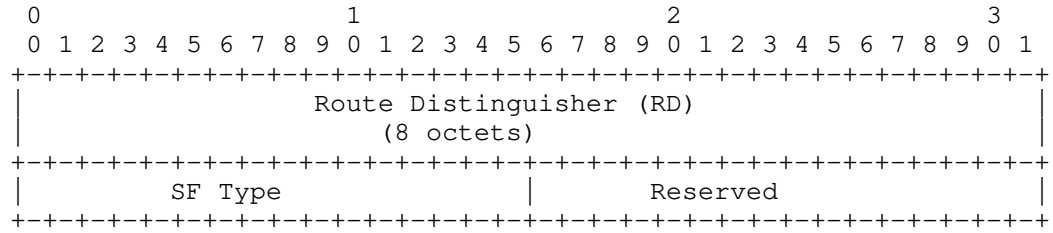


Figure 5: SFC Basic Unit sub-TLV

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

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

A node that receives an LSP ping with the new FEC will check if it is its Route Distinguisher and whether it advertised that Service Function Type.

## 3.4. Theory of Operation

An MPLS SFC validation request is an MPLS echo request with an SFC validation TLV, and the echo request is sent with a label stack corresponding to the SFP being tested.

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

After general packet sanity verifying[RFC8029], section 3.4.1 and section 3.4.2 in this document separately describe the following processing procedures in service programming and MPLS-based NSH.

After all SFFs on the SFP send back MPLS echo reply, the sender collects information about all traversed SFFs and SFs on the rendered service path (RSP).

### 3.4.1. MPLS-based service programming

[I-D.ietf-spring-sr-service-programming] describes how a service can be associated with a SID to achieve service function chaining. In an SR-MPLS network, the SFP is encoded as a stack of MPLS labels. That stack is pushed on top of the packet.

If an SFC validation TLV is present in the received echo request, an SFF MUST parse through the label stack until the next label is not a local service SID to get all the SFs attached to the SFF on the SFP and record the corresponding Label-stack-depth.

The SFF then sends an MPLS echo reply with all the SF information recorded in SFC Information Sub-TLV, including the service SID and the SF type.

### 3.4.2. RFC 8595 path consistency

[RFC8595] describes how Service Function Chaining (SFC) can be achieved in an MPLS network using a logical representation of the Network Service Header (NSH) in an MPLS label stack.

SFC forwarding can be achieved by label swapping, label stacking, or the mix of both. When an SFF receives a packet containing an MPLS label stack, it examines the top basic unit of MPLS label stack for SFC, {SPI, SI} or {context label, SFI index}, to determine where to send the packet next.

Upon receiving the SFC validation request, an SFF checks the MPLS label stack to get all the locally attached basic units for SFC. Then, the SFF sends back a reply message, including SFC info sub-TLVs, for each basic unit local to the SFF.

### 3.5. Discussion

In [RFC8595], it says, "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". To trace SFC, it should be changed to allow punting the packet to the control plane though under throttling control.

## 4. References

### 4.1. Normative References

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", draft-ietf-bess-nsh-bgp-control-plane-15 (work in progress), June 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-02 (work in progress), March 2020.

[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>.

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

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

### 4.2. Informative References

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

Authors' Addresses

Liu Yao
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: liu.yao71@zte.com.cn

Greg Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com

MPLS
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

Q. Xiong
G. Mirsky
ZTE Corporation
W. Cheng
China Mobile

July 13, 2020

# The Use of Path Segment in SR-MPLS and MPLS Interworking
## draft-xiong-mpls-path-segment-sr-mpls-interworking-02

Abstract

This document illustrates the SR-MPLS and MPLS interworking scenarios to support end-to-end bidirectional tunnel across multiple domains with the use of Path Segments.

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 14, 2021.

Table of Contents



## 1. Introduction

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an SR Policy instantiated as an ordered list of instructions called "segments". SR supports a per-flow explicit routing while maintaining per-flow state only at the ingress nodes of the SR domain. Segment Routing can be instantiated on MPLS data plane which is referred to as SR-MPLS [RFC8660]. SR-MPLS leverages the MPLS label stack to construct the SR path.

IP/MPLS technology can be deployed in domains, which may serve as an access, aggregation, or core network. Further, using SR architecture, the IP/MPLS network may be upgraded to support the SR-MPLS technology. As such transformation is performed incrementally, by one domain at the time, operators are faced with a requirement to support the interworking between MPLS and SR-MPLS networks at the boundaries to provide the end-to-end bidirectional service. As defined in [RFC8402], the headend of an SR Policy binds a Binding Segment ID(B-SID) to its policy. The B-SID could be bound to a SID List or selected path and used to stitch the SR list and the SR Label Switched Paths (LSP) across multiple domains. The use of the B-SID is recommended to reduce the size of the label stack and stitch the SR LSPs.

In some scenarios, for example, a mobile backhaul transport network, it is required to provide end-to-end bidirectional path across SR and MPLS networks. The Path Segment as defined in [I-D.ietf-spring-mpls-path-segment] can be used to support bidirectional tunnel scenarios such as SR path Performance Measurement (PM), end-to-end 1+1 SR path protection and bidirectional SR paths correlation.

This document illustrates the SR-MPLS and MPLS interworking scenarios to support end-to-end bidirectional tunnel across multiple domains with the use of Path Segments.

## 2. Conventions used in this document

### 2.1. Terminology

ABR: Area Border Routers. Routers used to connect two IGP areas (areas in OSPF or levels in IS-IS).

AS: Autonomous System. An Autonomous System is composed by one or more IGP areas.

ASBR: Autonomous System Border Router. A router used to connect together ASes of the same or different service providers via one or more inter-AS links.

Border Node: An ABR that interconnects two or more IGP areas.

Border Link: Two ASes are interconnected with ASBRs.

B-SID: Binding Segment ID.

Domains: Autonomous System (AS) or IGP Area. An Autonomous System is composed of one or more IGP areas.

e-PSID: end-to-end Path Segment.

IGP: Interior Gateway Protocol.

N-PSID: Nesting of Path Segments.

PM: Performance Measurement.

SID: Segment ID.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

S-PSID: Stitching of Path Segments.

VPN: Virtual Private Network.

### 2.2. Requirements Language

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

## 3. SR-MPLS Interworking with MPLS

It is required to establish the end-to-end Virtual Private Network (VPN) service across the access network, aggregation network, and core network. For example, SR-MPLS may be deployed in access and core network, and MPLS may be deployed in the aggregation network. The network interworking should be taken into account in deployment are the following:

- Border Node or Border Link
- Stitching of Path Segments or Nesting of Path Segments
- End-to-end Path Monitoring

The domains of the networks may be IGP Areas or ASes. The SR-MPLS and MPLS networks can be interconnected with a border node between IGP areas or border links between ASes. MPLS domain can be deployed between two SR-MPLS domains, as Figure 1 shows. The packets being transmitted along the SR path in SR-MPLS domains by using the SID list at the ingress node. And the path in MPLS domains can be pre-configuration either via NMS or via the MPLS control plane signaling. This document takes border node scenarios across IGP Areas domains for example. The border link scenarios are in future discussion.

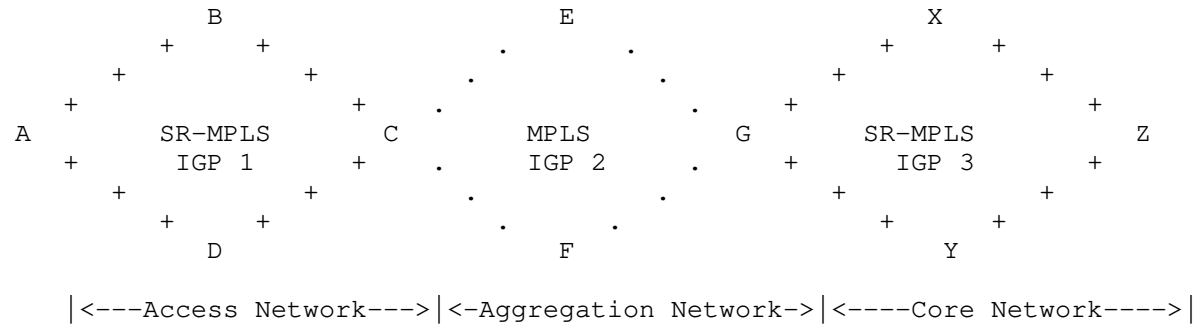


Figure 1: SR-MPLS and MPLS interworking Scenario

The VPN service across the SR-MPLS and MPLS domains is an end-to-end bidirectional path. In the SR-MPLS network, a Path Segment uniquely identifies an SR path and can be used for the end-to-end bidirectional path. This document illustrates the end-to-end Path Segment used in the interworking scenario including the stitching and nesting models. As described in [I-D.ietf-spring-mpls-path-segment], an end-to-end path segment or PSID (e-PSID), is also referred to as Nesting of Path SID (N-PSID) in nesting model or Stitching of Path SID (S-PSID) in stitching model.

## 3.1. Stitching of Path Segments

It is a common requirement that SR-MPLS needs to interwork with MPLS when SR is incrementally deployed in the MPLS domain. Figure 2 shows the stitching of Path Segments in SR-MPLS interworking with MPLS. The SR-LSPs and IP/MPLS LSPs are established independently in each domain which consist of SID list or MPLS label. The end-to-end bidirectional path acrossing the SR-MPLS and MPLS networks is split into multiple segments which can be identified by the S-PSID. The end-to-end path is terminated at the egress node in egress domain. The S-PSID will be popped out at the border node in each domain and correlated to the S-PSID of next domain.

The correlation of S-PSIDs can bind the segments of end-to-end path. The S-PSIDs are valid in the corresponding domain and the border nodes maintain the forwarding entries of that S-PSID segment that maps to the next S-PSID and the related path segments. In the headend node, the S-PSID can correlate the inter-domain path of reverse direction and bind the two unidirectional paths. The stitching of Path Segments can support the end-to-end path stitching and monitoring.

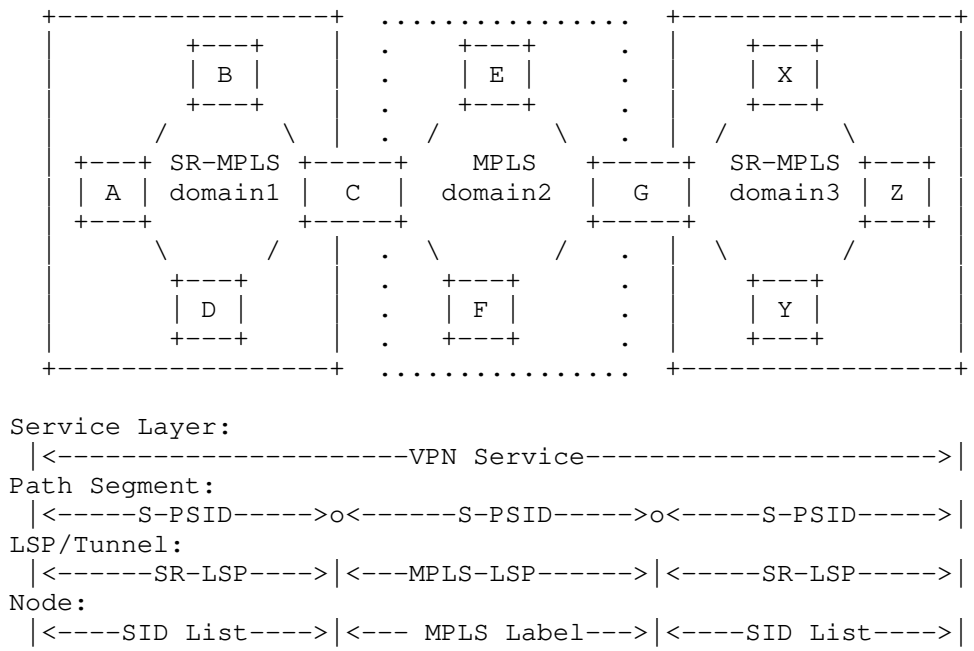


Figure 2: Stitching of Path Segments in SR-MPLS and MPLS interworking

## 3.2. Nesting of Path Segments

Figure 3 displays the nesting of Path Segments in SR-MPLS and MPLS interworking. The SR-LSPs and IP/MPLS LSPs are established in respective domain which consist of SID list or MPLS label. The SR-LSPs and IP/MPLS LSPs may be stitched across domains with B-SID. Comparing with S-PSID in the stitching model, the N-PSID presents end-to-end encapsulation in the packet from an SR-MPLS domain to an MPLS domain which is encapsulated at the ingress nodes and decapsulated at the egress nodes. The transit nodes, even the border nodes of domains, are not aware of the N-PSID.

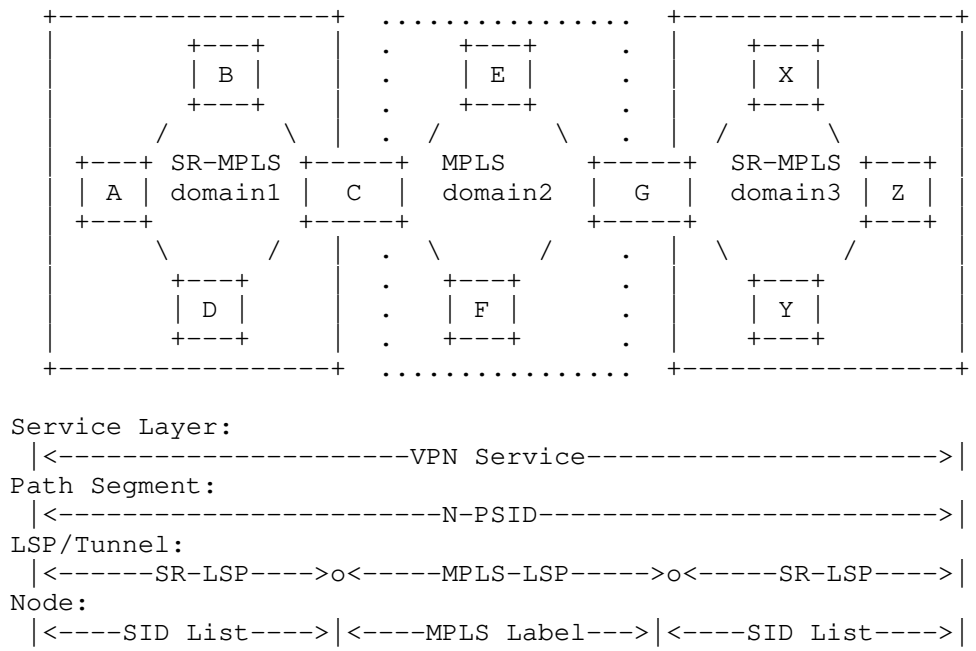


Figure 3: Nesting of Path Segments in SR-MPLS and MPLS interworking

## 4. Security Considerations

TBA

## 5. Acknowledgements

TBA

## 6. IANA Considerations

TBA

## 7. Normative References

[I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-02 (work in progress), February 2020.

[I-D.xiong-spring-path-segment-sr-inter-domain]
Xiong, Q., Mirsky, G., and W. Cheng, "The Use of Path Segment in SR Inter-domain Scenarios", draft-xiong-spring-path-segment-sr-inter-domain-01 (work in progress), October 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>.

[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>.

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

## Authors' Addresses

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

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

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com