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.

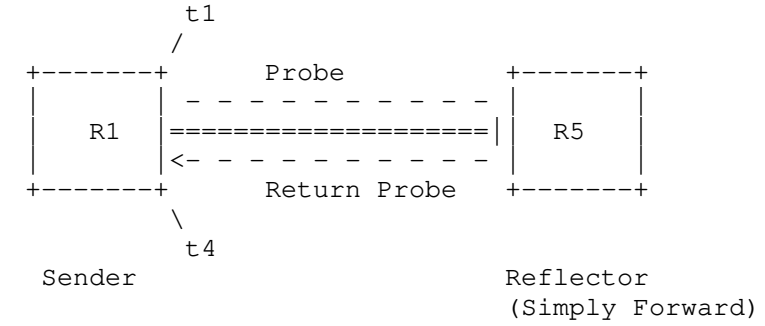


Figure 1: Reference Topology

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:

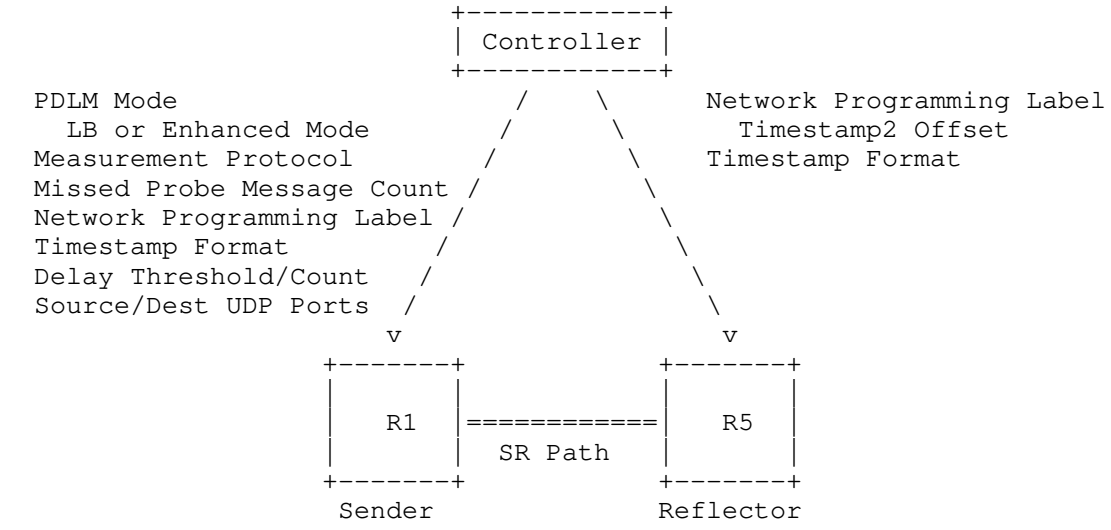


Figure 2: Example Provisioning Model

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.

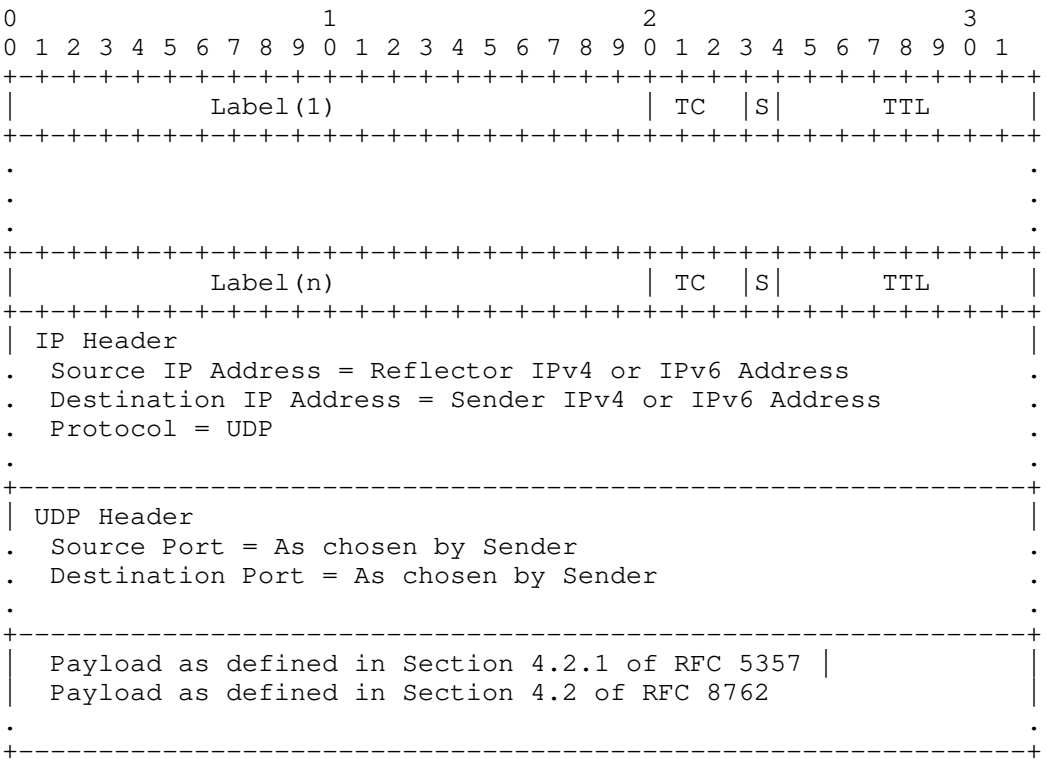


Figure 3: Example Probe Message for SR-MPLS

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.

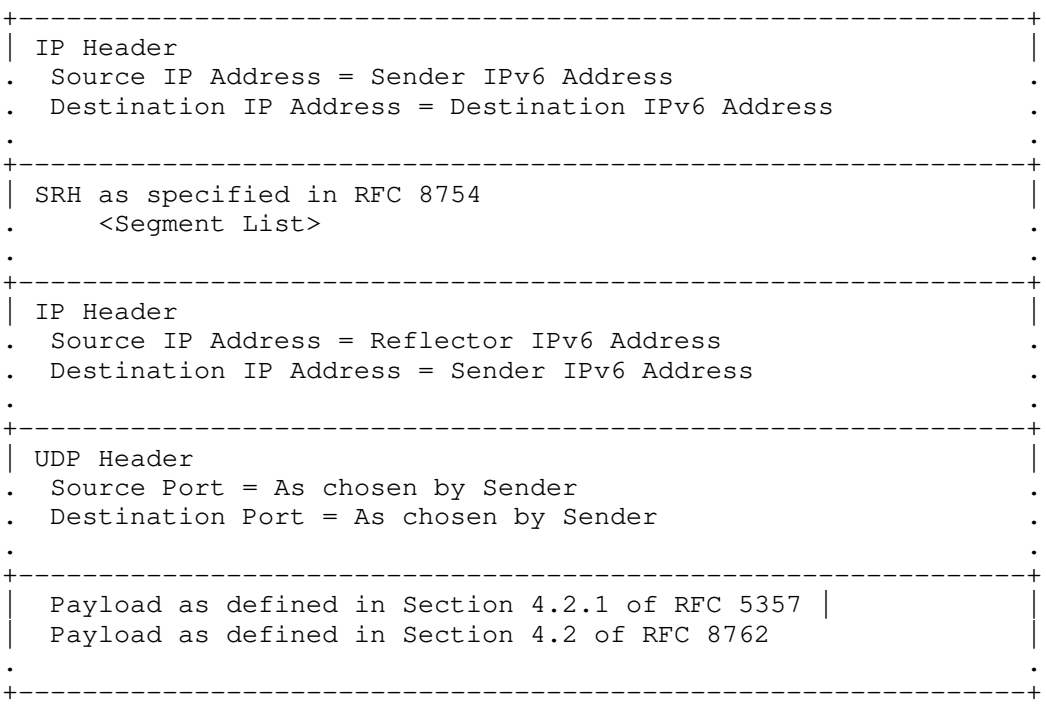


Figure 4: Example Probe Message for SRv6

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.

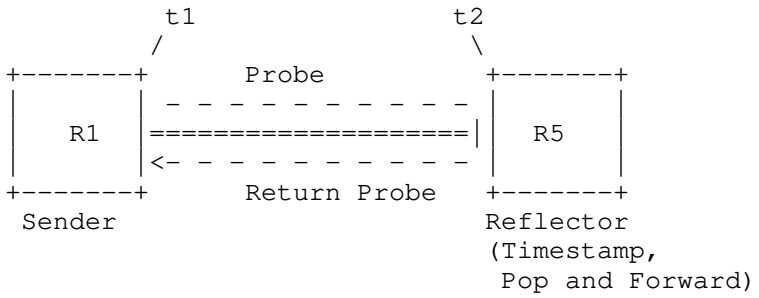


Figure 5: Loopback Mode Enabled with Network Programming

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).

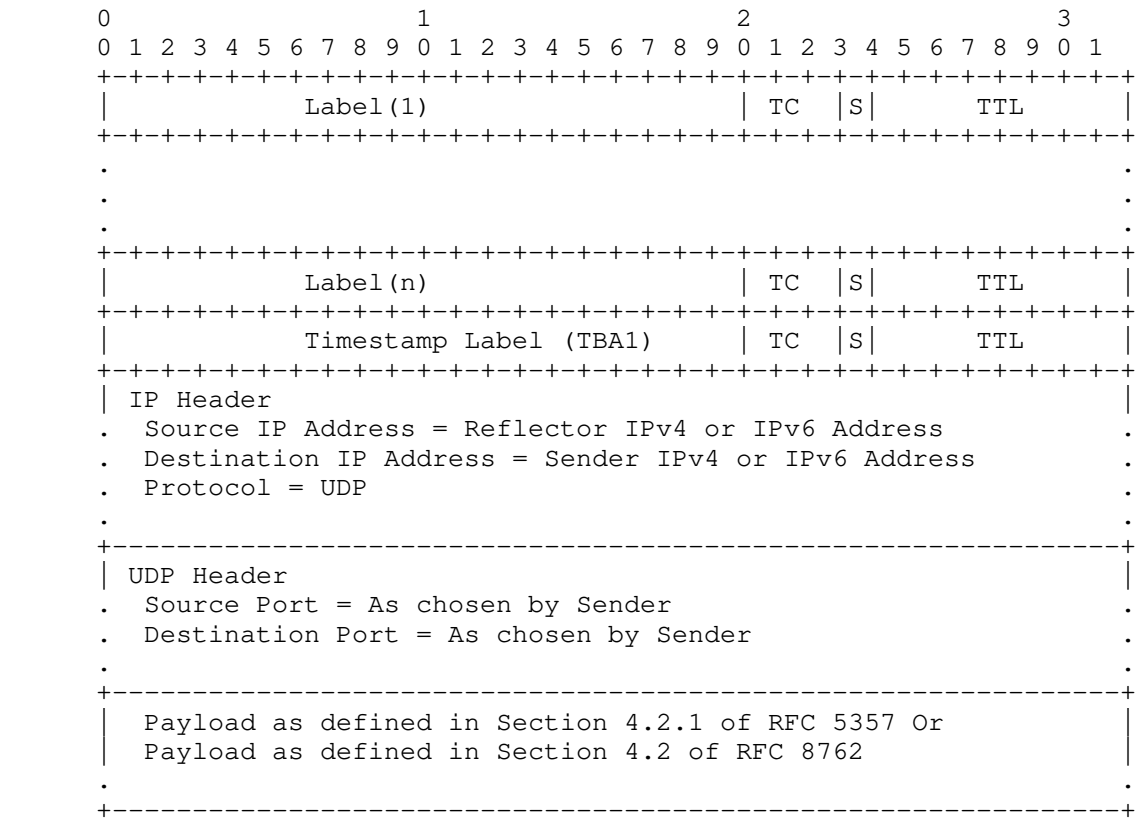


Figure 6: Example Probe Message with Timestamp Label for SR-MPLS

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).

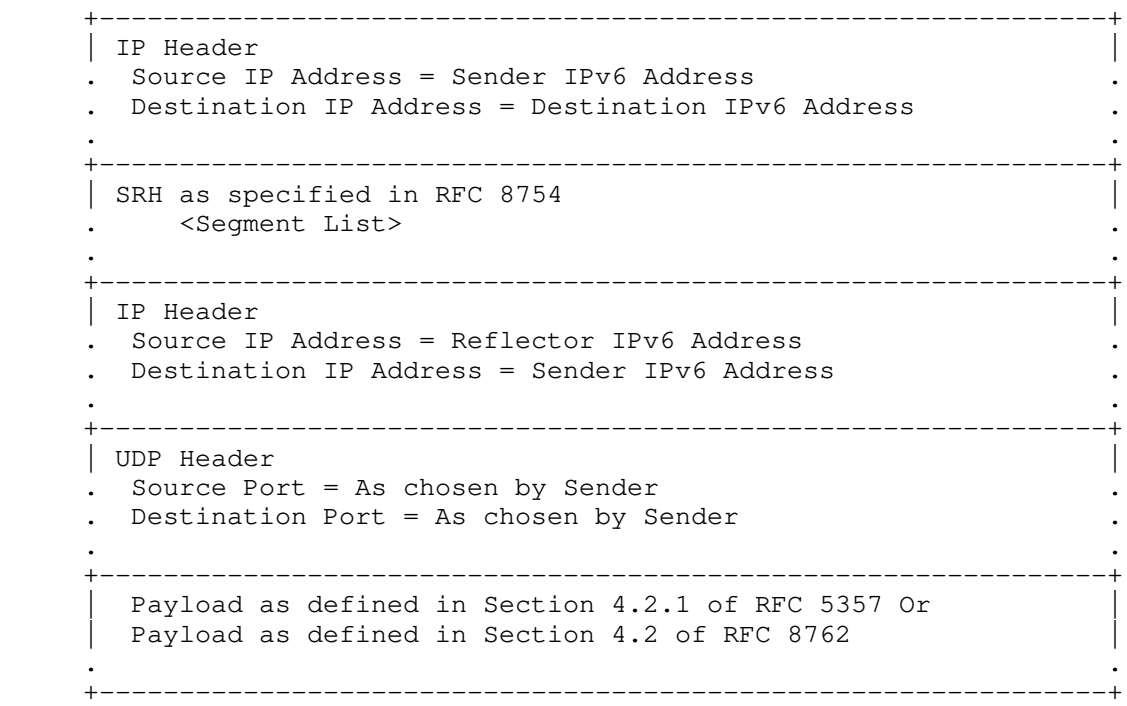


Figure 7: Example Probe Message with Endpoint Function for SRv6

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

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 30, 2021

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

September 26, 2020

Enhanced Performance Delay and Liveness Monitoring in Segment Routing Networks
draft-gandhi-spring-sr-enhanced-plm-03

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document defines procedure for Enhanced Performance Delay and Liveness Monitoring (PDLM) in Segment Routing networks. The procedure leverages the probe messages compatible with the delay measurement message formats defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP)) and RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) and is applicable to end-to-end SR Paths including SR Policies for 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 Performance Delay Measurement (DM) as well as Liveness Monitoring for Connectivity Verification (CV) and Continuity Check (CC) 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 leverages the probe messages compatible with the delay measurement message formats defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP)) and RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) and is applicable to end-to-end SR Paths including SR Policies for 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 in Figure 1, 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.

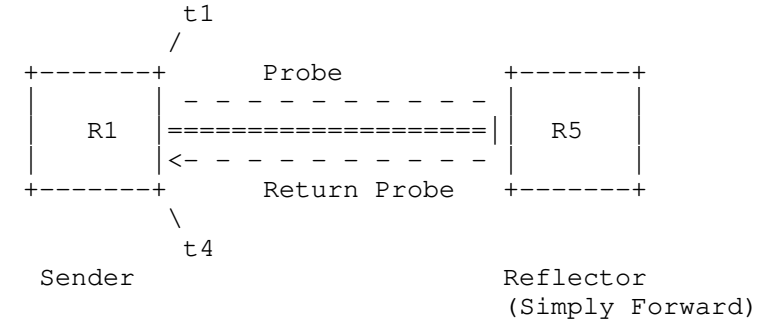


Figure 1: Reference Topology

3. Overview

3.1. Loopback Mode

In loopback mode, the sender node R1 initiates probe messages and the reflector node R5 forwards them just like data packets for the normal traffic back to the sender node R1. 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. No PM session is created on the reflector node.

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). Both Source and Destination UDP ports in the probe messages are allocated dynamically or user-configured from the range specified in [RFC8762]. The UDP ports used in loopback mode are different than the ports used for TWAMP and STAMP sessions. The IPv4 Time To Live (TTL) and IPv6 Hop Limit (HL) are set to 255.

3.2. Loopback Mode Enabled with Network Programming Function

In "loopback mode enabled with network programming function", both transmit (t1) and receive (t2) timestamps in data plane are collected by the probe messages sent in loopback mode as shown in Figure 2. The network programming function optimizes the "operations of punt and inject the probe packet" on the reflector node as timestamping is implemented in hardware. This helps to achieve higher scale and faster rate, resulting in faster failure detection.

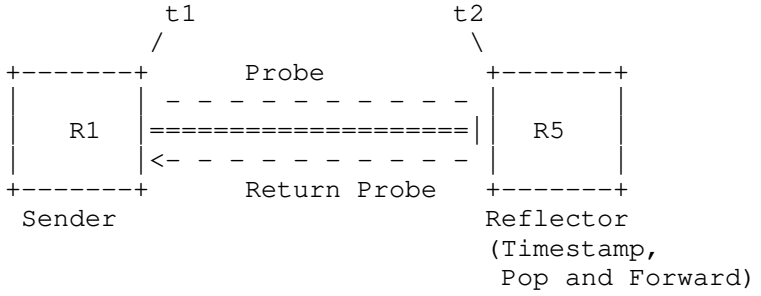


Figure 2: Loopback Mode Enabled with Network Programming Function

The sender node adds transmit (t1) timestamp in the payload of the probe message and clears the receive (t2) timestamp. The reflector node adds the receive timestamp in the payload of the received probe message in hardware 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 probe message reaches the intended reflector node and the receive timestamp is returned by the that node. The payload of the probe message is not modified by any intermediate nodes.

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 the probe message defined in Figure 4 for delay measurement, the 64-bit receive timestamp is added at byte-offset 16 which is from the start of the payload.

3.3. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 3:

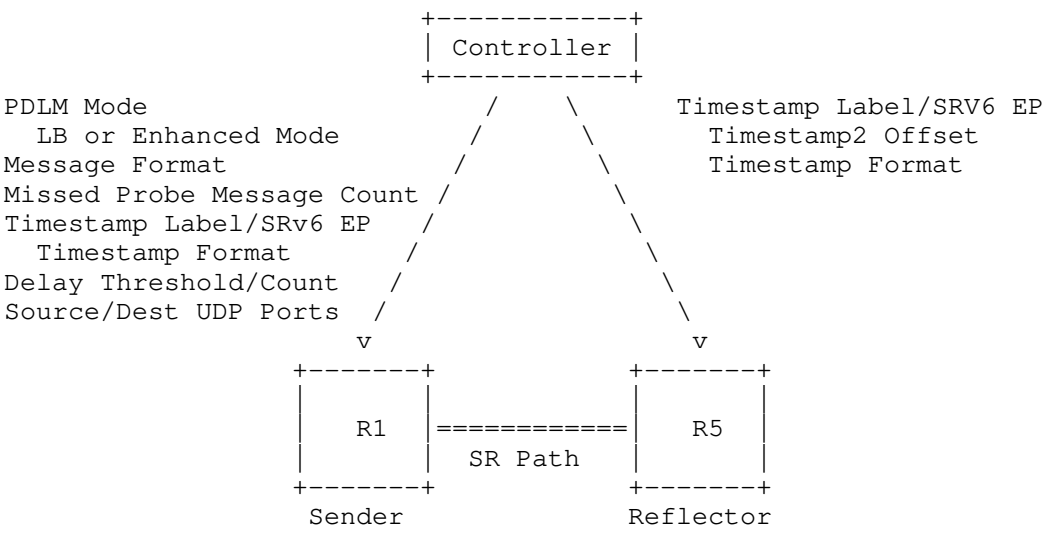


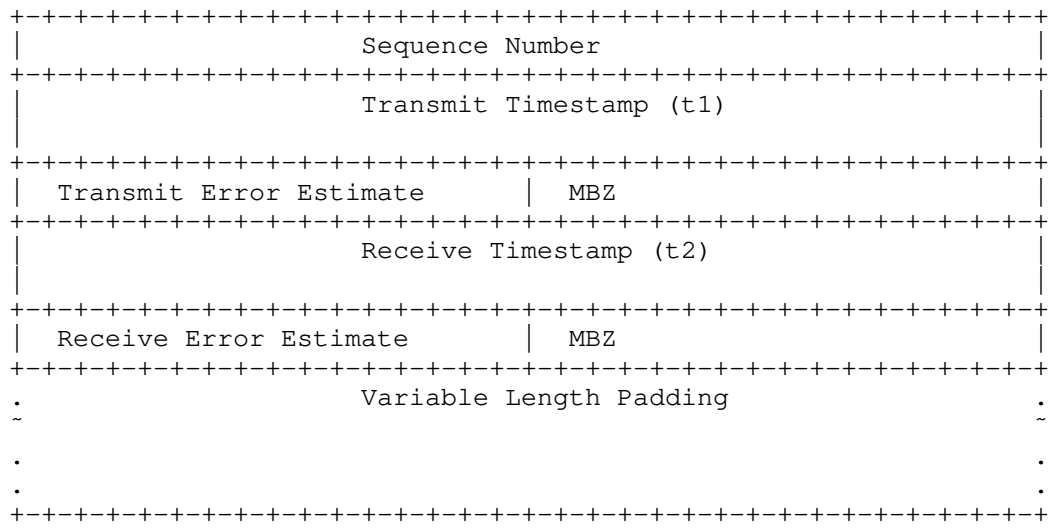
Figure 3: Example Provisioning Model

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

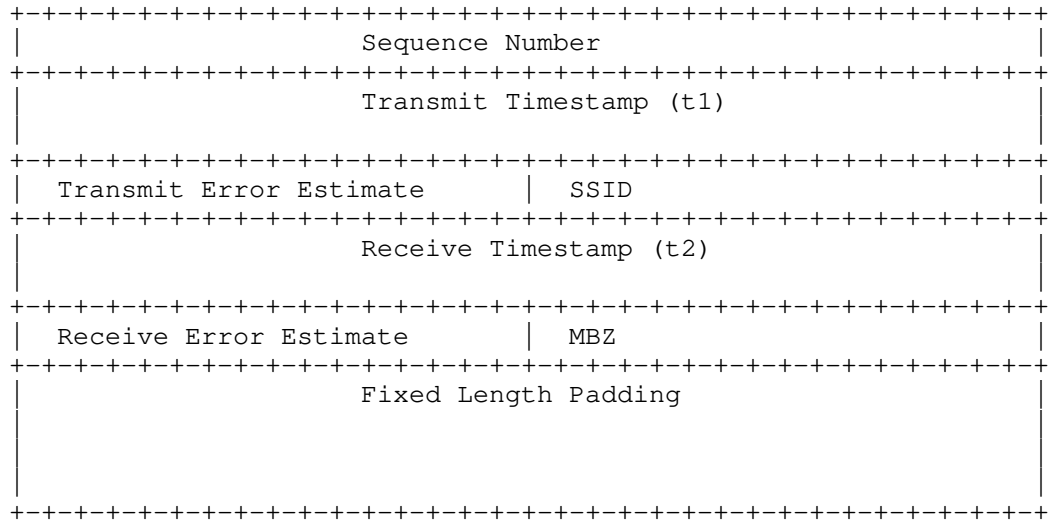
The mechanisms to provision the sender and reflector nodes are outside the scope of this document.

4. Probe Message Formats

The probe messages compatible with the delay measurement message formats defined in TWAMP [RFC5357] and STAMP [RFC8762] are specified in Figure 4.



TWAMP Compatible Probe Packet Format



STAMP Compatible Probe Packet Format

Figure 4: Probe Packet Formats

Sequence Number is the sequence number of the probe packet according to its transmit order. It starts with zero and is incremented by one for each subsequent packet.

Transmit Timestamp and Transmit Error Estimate are the Sender's transmit timestamp and error estimate for the probe packet, respectively. Similarly, Receive Timestamp and Receive Error Estimate are the Reflector's receive timestamp and error estimate, respectively. The timestamp and error estimate fields follow the definition and formats defined in Section 4.1.2 in [RFC4656]. Timestamp format preferred is 64-bit PTPv2 [IEEE1588] as specified in [RFC8186], implemented in hardware.

5. 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.

For SR Policy, the probe messages are sent using the Segment List (SL) of the Candidate-path [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 Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path].

5.1. Probe Message for SR-MPLS

The probe messages are sent using the MPLS header containing the label stack of the SR Policy as shown in Figure 5. 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.

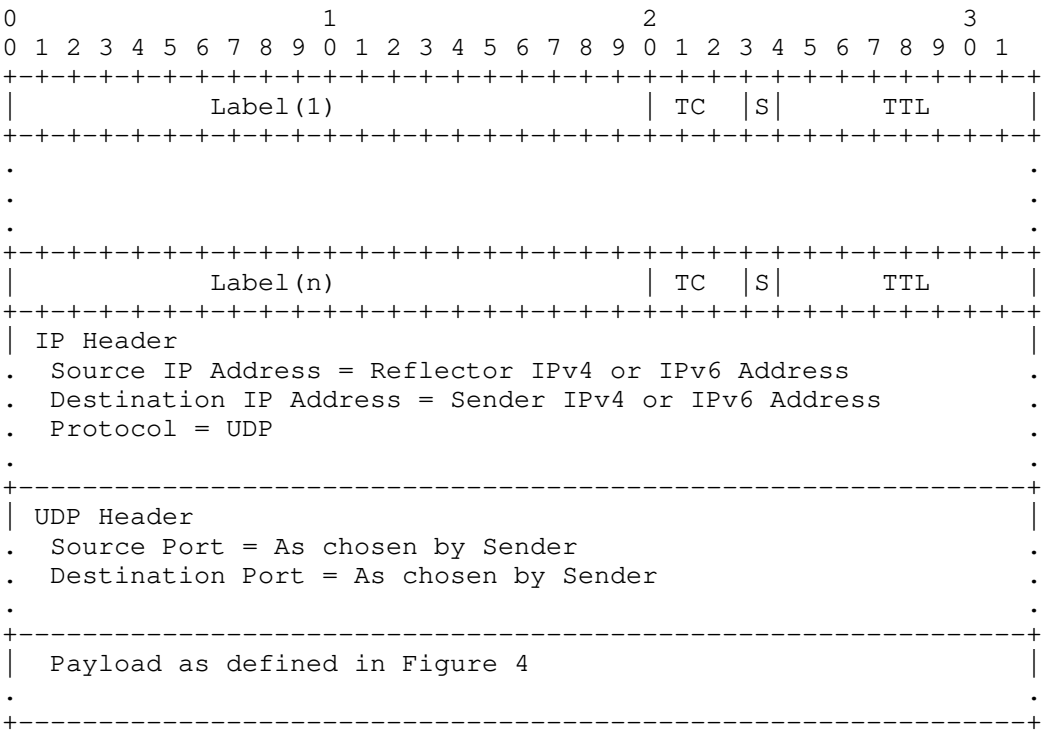


Figure 5: Example Probe Message for SR-MPLS

5.2. Probe Message for SRv6

The 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 6. 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. When the return probe message contains an SRH at the sender node, the procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe messages.

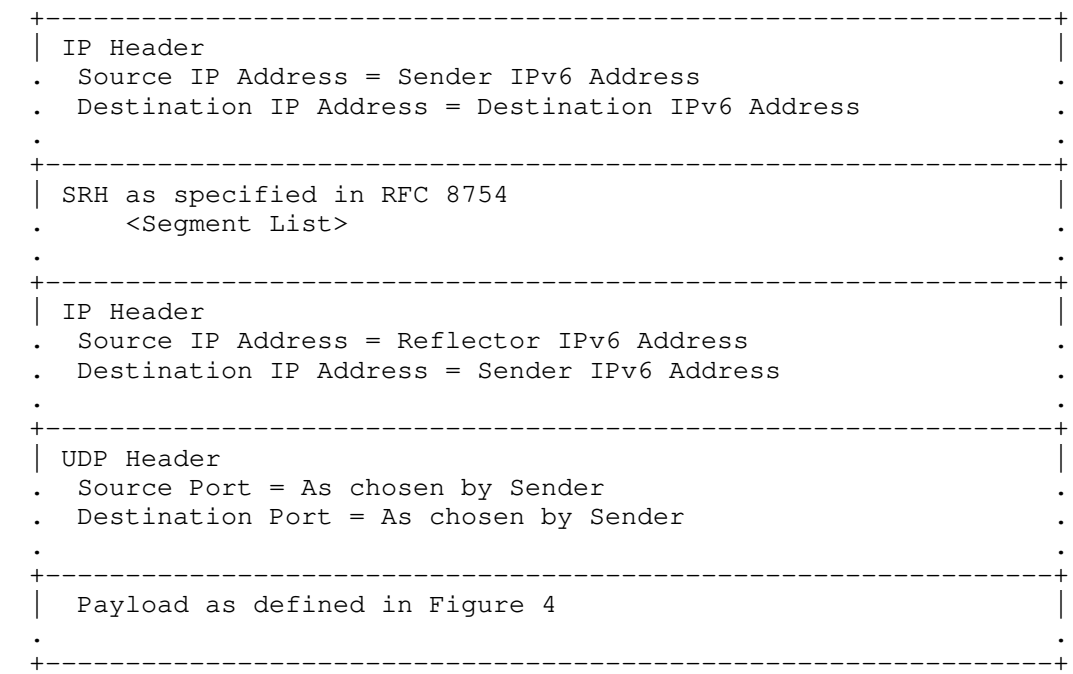


Figure 6: Example Probe Message for SRv6

6. 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 function".

6.1. Probe Message with Timestamp Label for SR-MPLS

In this document, new Timestamp Label (Extended Special-Purpose 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 7, to collect "Receive Timestamp" field in the payload of the 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 packet 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).

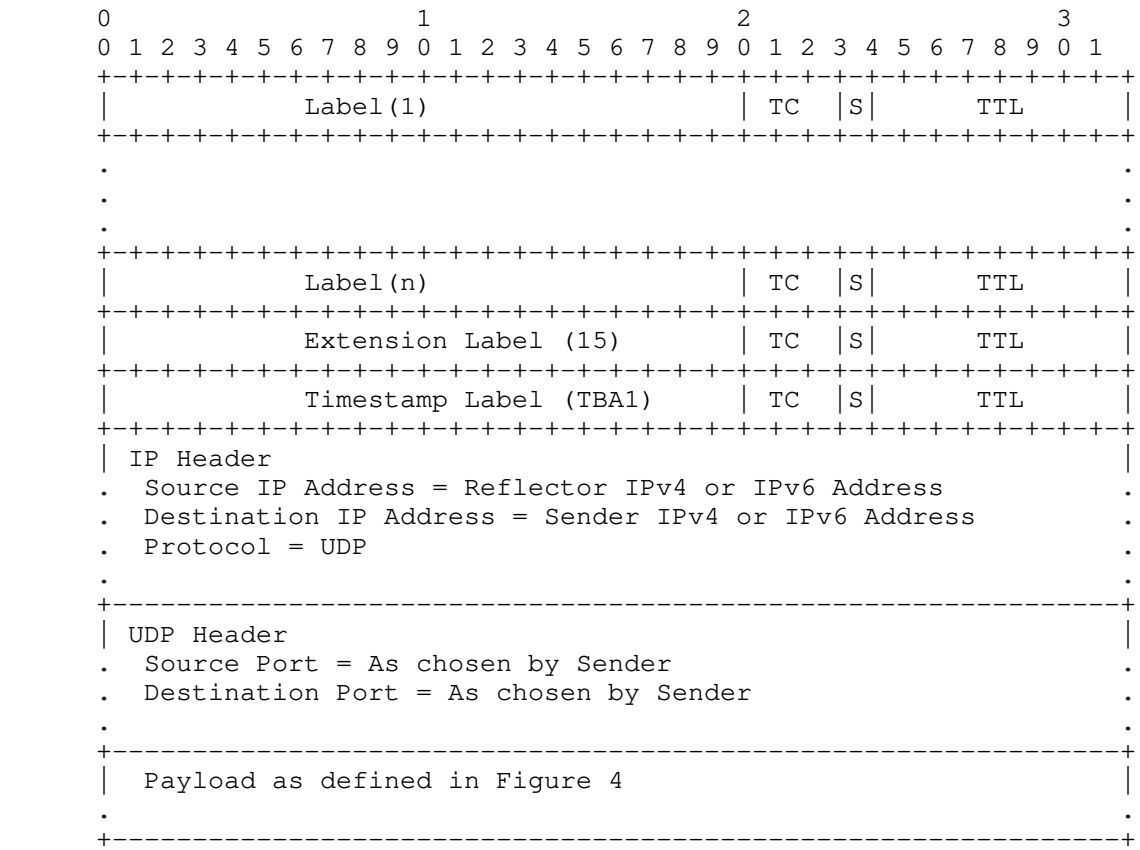


Figure 7: Example Probe Message with Timestamp Label for SR-MPLS

6.1.1. Timestamp Label Allocation

Timestamp Label can be allocated using one of the following methods:

- Label (value TBA1) assigned by IANA from the "Extended Special-Purpose MPLS Values" [I-D.ietf-mpls-spl-terminology]. The timestamp offset is fixed at byte-offset 16 from the start of the payload and timestamp format is 64-bit PTPv2 for this label.

- Label allocated by a Controller from the global table of the egress node. The Controller provisions the label on both ingress and egress nodes, as well as timestamp offset and timestamp format.

- Label allocated by the egress node. The signaling and IGP flooding extension for the label (including timestamp offset and timestamp format) are outside the scope of this document.

6.1.2. Node Capability for Timestamp Label

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

6.2. Probe Message with Timestamp Endpoint Function for SRv6

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

In the probe message for SRv6, End.TSF function is added for the target Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 8, to collect "Receive Timestamp" field in the payload of the probe message. The Segment List for the reverse path can be added after the target SID to receive the return probe message on a specific path. When a reflector node receives a message 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 message (just like the data packets for the normal traffic).

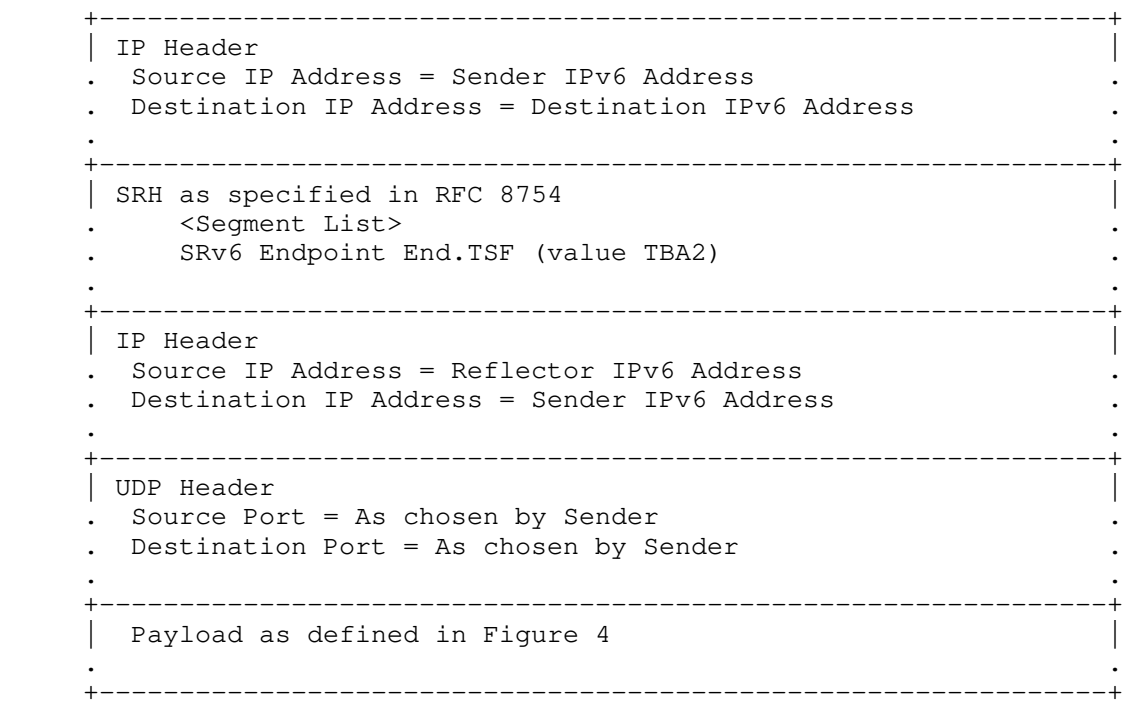


Figure 8: Example Probe Message with Endpoint Function for SRv6

6.2.1. Timestamp Endpoint Function Assignment

Timestamp endpoint function for "Timestamp and Forward" can be signaled using one of the following methods:

- Timestamp endpoint function (value TBA2) assigned by IANA from the "SRv6 Endpoint Behaviors Registry". The timestamp offset is fixed at byte-offset 16 from the start of the payload and timestamp format is 64-bit PTPv2 for this endpoint function.

- Timestamp endpoint function assigned by a Controller. The Controller provisions the value on both ingress and egress nodes, as well as timestamp offset and timestamp format.

- Timestamp endpoint function assigned by the egress node. The signaling and IGP flooding extension for the endpoint function (including timestamp offset and timestamp format) are outside the scope of this document.

6.2.2. Node Capability for Timestamp Endpoint Function

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

7. 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.

8. Failure Notification

Liveness success for SR Path is initially notified as soon as one or more return probe messages are received at the sender node.

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 as an example 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.

For the probe messages generated by the Sender node R1 in the loopback mode, a failure on the reverse direction path can also cause the return probe messages to not reach the Sender node. This is also true in case of the probe response messages generated by the Reflector node R5 e.g. to indicate node R1 of any failure on the forward direction path. As such, the probe-based methods have this limitation for the liveness monitoring of the forward direction path.

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

9. 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. Cryptographic measures may be used by the correct configuration of access-control lists and firewalls.

10. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>). IANA is requested to allocate 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 Parameters" registry [I-D.ietf-spring-srv6-network-programming], the following allocation:

Value	Endpoint Behavior	Reference
TBA2	End.TSF (Timestamp and Forward)	This document

11. References

11.1. 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>.

11.2. 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-10 (work in progress), August 2020.

[I-D.gandhi-spring-stamp-srpm]
Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks", draft-gandhi-spring-stamp-srpm-02 (work in progress), August 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 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-20 (work in progress), September 2020.

[I-D.ietf-mpls-spl-terminology]
Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-04 (work in progress), September 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-03 (work in progress), September 2020.

Acknowledgments

The authors would like to thank Greg Mirsky, Mach Chen, Kireeti Kompella, and Adrian Farrel for providing the review comments.

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

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 7, 2020

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

June 5, 2020

Performance Measurement Using STAMP for Segment Routing Networks
draft-gandhi-spring-stamp-srpm-01

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the mechanisms defined in RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) for Delay Measurement, and uses the mechanisms defined in this document for Loss Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 7, 2020.

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. 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 SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks using probe messages [RFC8762]. It eliminates the need for control-channel signaling by using configuration data model to provision a test-channel (e.g. UDP paths). [I-D.ietf-ippm-stamp-option-tlv] defines TLV extensions for STAMP messages.

The STAMP message with a TLV for "direct measurement" can be used for combined Delay + Loss measurement [I-D.ietf-ippm-stamp-option-tlv]. However, in order to use only for loss measurement purpose, it requires the node to support the delay measurement messages and support timestamp for these messages (which may also require clock synchronization). Furthermore, for hardware-based counter collection for direct-mode loss measurement, the optional TLV based processing adds unnecessary overhead (as counters are not at well-known locations).

This document specifies procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the mechanisms defined in [RFC8762] (STAMP) (including the TLV extensions) for Delay Measurement (DM), and uses the mechanisms defined in this document for Loss Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies. This document also defines mechanisms for handling ECMPs of SR Paths for performance delay measurement. Unless otherwise specified, the mechanisms defined in [RFC8762] and [I-D.ietf-ippm-stamp-option-tlv] are not modified by this document.

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

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

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.

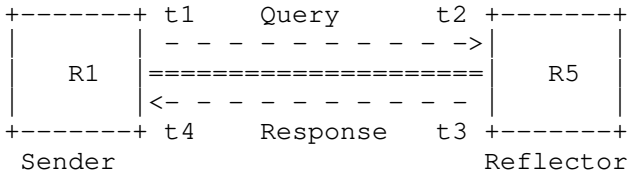
SSID: STAMP Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a probe query for performance measurement and the reflector node R5 sends a probe response for the query message received. The probe response is sent to the sender node R1. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.voyer-spring-sr-replication-segment].



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC8762] are used. For direct-mode and inferred-mode loss measurements in Segment Routing networks, the messages defined in this document are used. Separate UDP destination port numbers are user-configured for delay and loss measurements from the range specified in [RFC8762]. As specified in [RFC8762], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages defined in this document. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. For both Links and end-to-end SR Paths including SR Policies, no PM session for delay or loss measurement is created on the reflector node R5 [RFC8762].

For Performance Measurement, probe query and response messages are sent as following:

- For Delay Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Paths.

- For Loss Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM session state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

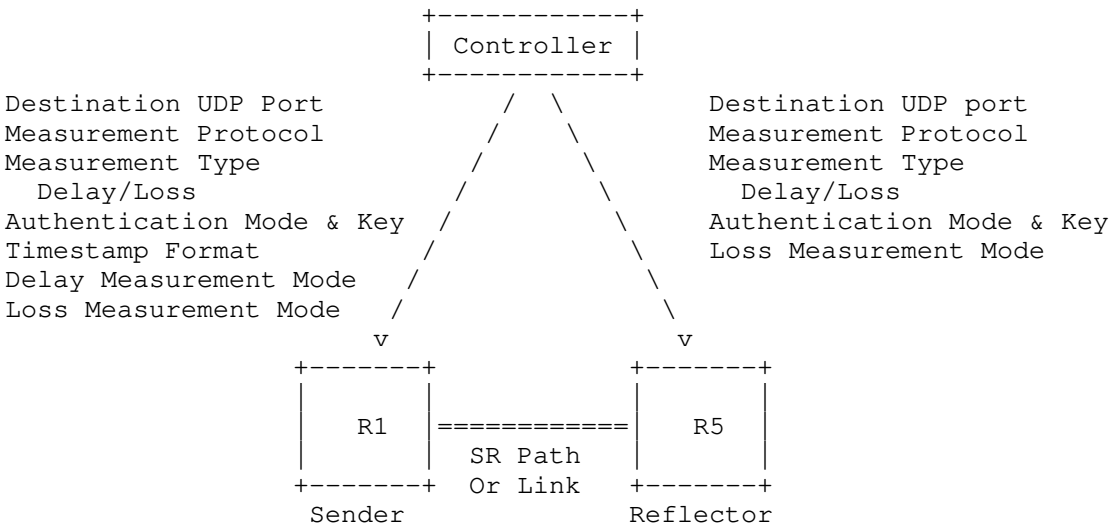


Figure 1: Example Provisioning Model

Example of Measurement Protocol is STAMP, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

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

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC8762] are used for Delay Measurement for Links and end-to-end SR Paths including SR Policies. For Loss Measurement, the probe messages defined in this document are used.

The Sender IPv4 or IPv6 address is used as the source address. The reflector IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the address in the range of 127/8 for IPv4 or ::FFFF:127/104 for IPv6 is used as the destination address, respectively.

4.1.1. Delay Measurement Query Message

The message content for Delay Measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in [RFC8762].

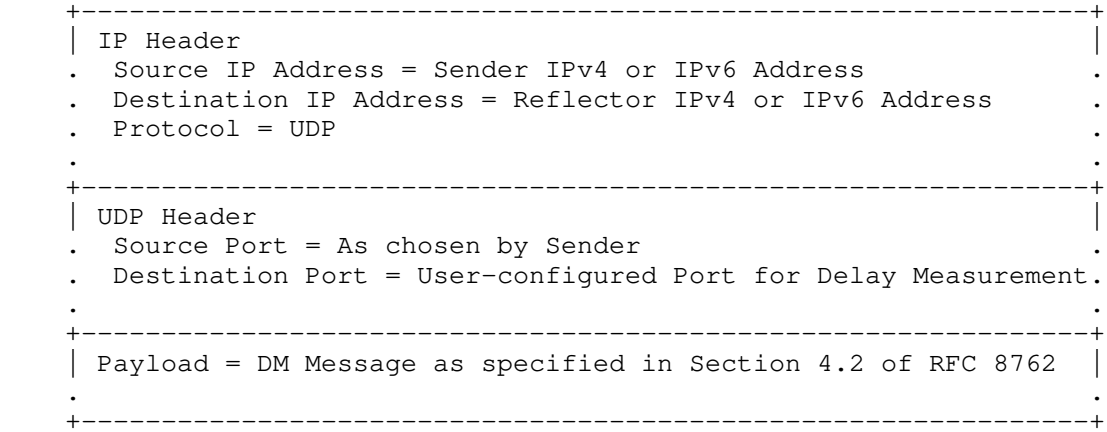


Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC8762]. It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for Loss Measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in Figure 7 and Figure 8.

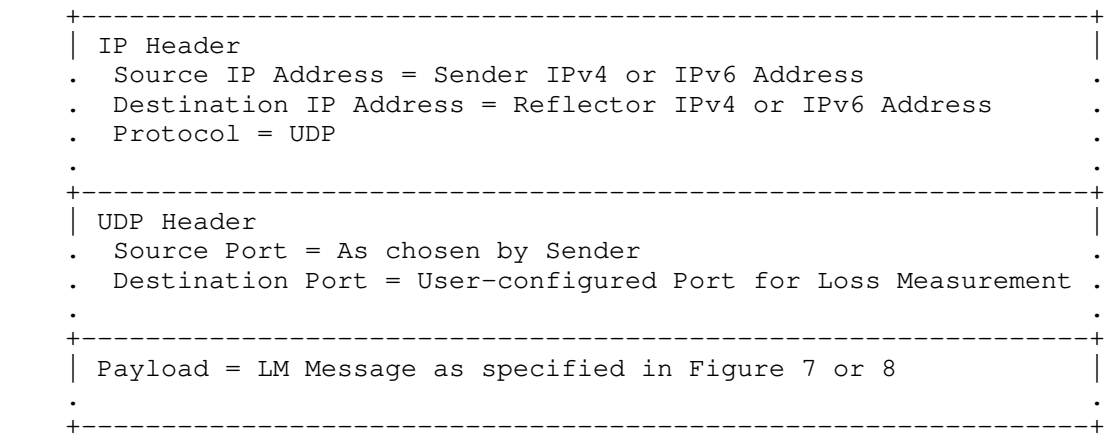


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement is sent on the congruent path of the data traffic. The probe messages are routed over the Link for both delay and loss measurement.

4.1.4. Probe Query for End-to-end Measurement for SR Policy

The performance delay and loss measurement for segment routing is applicable to both SR-MPLS and SRv6 Policies.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for end-to-end performance measurement of an SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

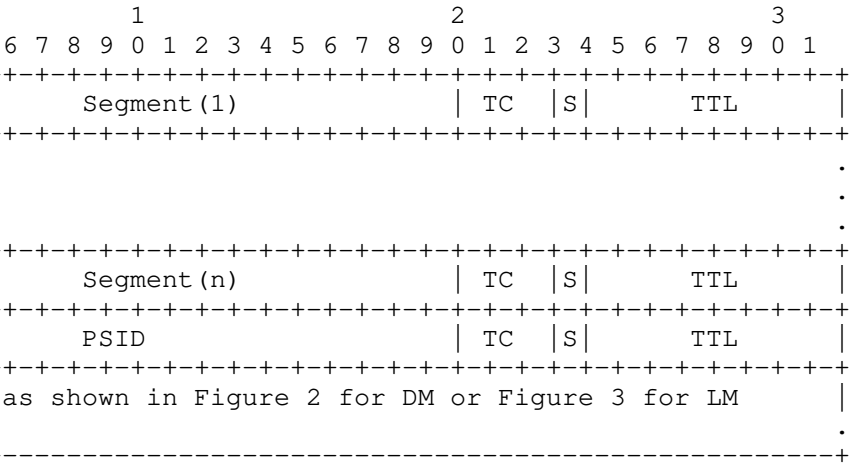


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. For SRv6, network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for end-to-end performance measurement of an SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5.

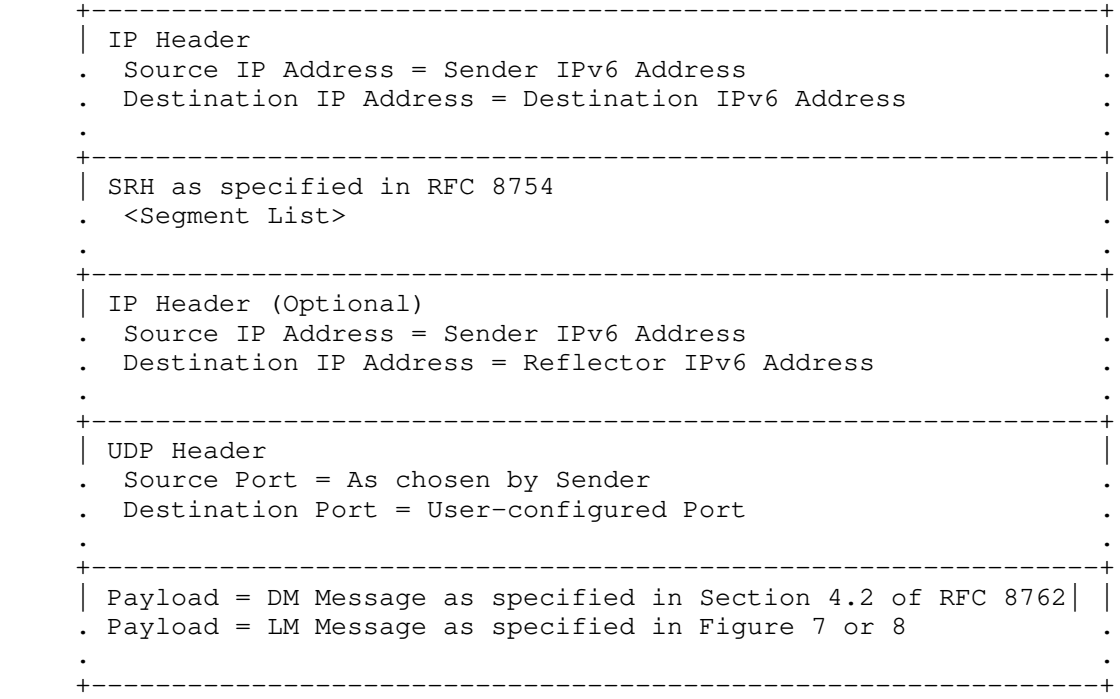


Figure 5: Example Probe Query Message for SRv6 Policy

4.1.5. Control Code Field for STAMP Messages

The Control Code field is defined for delay and loss measurement probe query messages for STAMP protocol in unauthenticated and authenticated modes. The modified delay measurement probe query message format is shown in Figure 6. This message format is backwards compatible with the message format defined in STAMP [RFC8762] as its reflector MUST ignore the received field (previously

identified as MBZ). The usage of the Control Code is not limited to the SR paths and can be used for non-SR paths in a network.

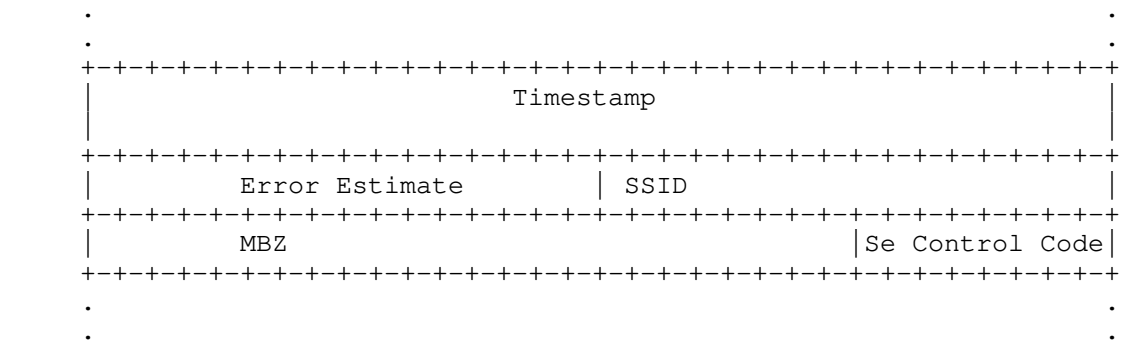


Figure 6: Sender Control Code in STAMP DM Message

Sender Control Code: Set as follows in STAMP probe query message.

In a Query:

0x0: Out-of-band Response Requested. Indicates that the probe response is not required over the same path in the reverse direction. This is also the default behavior.

0x1: In-band Response Requested. Indicates that this query has been sent over a bidirectional path and the probe response is required over the same path in the reverse direction.

0x2: No Response Requested.

4.1.6. Loss Measurement Query Message Formats

In this document, STAMP probe query messages for loss measurement are defined as shown in Figure 7 and Figure 8. The message formats are hardware efficient due to well-known locations of the counters and payload small in size. They are stand-alone and similar to the delay measurement message formats (e.g. location of the Counter and Timestamp). They also do not require backwards compatibility and support for the existing DM message formats from [RFC8762] as different user-configured destination UDP port is used for loss measurement.

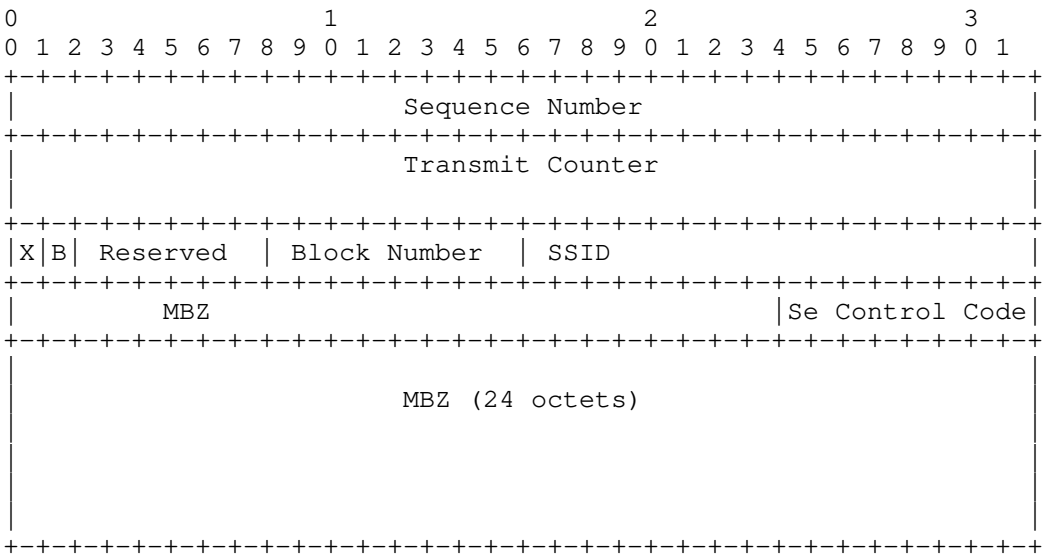


Figure 7: STAMP LM Probe Query Message - Unauthenticated Mode

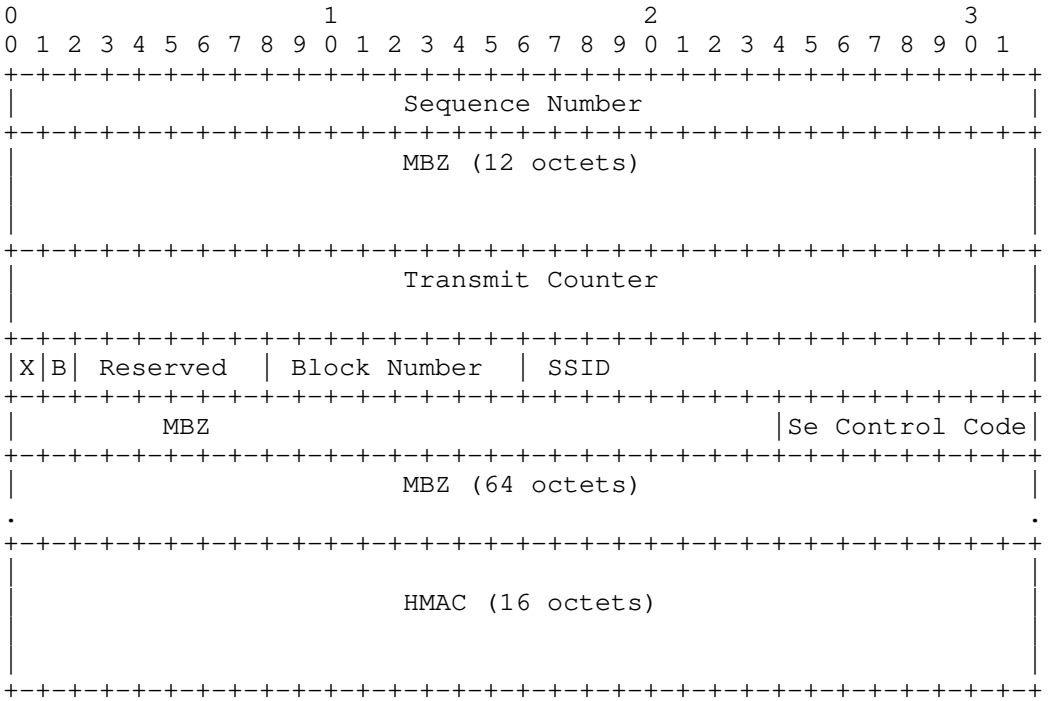


Figure 8: STAMP LM Probe Query Message - Authenticated Mode

Sequence Number (32-bit): As defined in [RFC8762].

Transmit Counter (64-bit): The number of packets or octets sent by the sender node in the query message and by the reflector node in the response message. The counter is always written at the well-known location in the probe query and response messages.

Receive Counter (64-bit): The number of packets or octets received at the reflector node. It is written by the reflector node in the probe response message.

Sender Counter (64-bit): This is the exact copy of the transmit counter from the received query message. It is written by the reflector node in the probe response message.

Sender Sequence Number (32-bit): As defined in [RFC8762].

Sender TTL: As defined in Section 7.1.

LM Flags: The meanings of the Flag bits are:

X: Extended counter format indicator. Indicates the use of extended (64-bit) counter values. Initialized to 1 upon creation (and prior to transmission) of an LM Query and copied from an LM Query to an LM response. Set to 0 when the LM message is transmitted or received over an interface that writes 32-bit counter values.

B: Octet (byte) count. When set to 1, indicates that the Counter 1-4 fields represent octet counts. The octet count applies to all packets within the LM scope, and the octet count of a packet sent or received includes the total length of that packet (but excludes headers, labels, or framing of the channel itself). When set to 0, indicates that the Counter fields represent packet counts.

Block Number (8-bit): The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to color the data traffic. To be able to compare the transmit and receive traffic counters of the matching color, the Block Number (or color) of the traffic counters is carried by the probe query and response messages for loss measurement.

HMAC: The PM probe message in authenticated mode includes a key Hashed Message Authentication Code (HMAC) [RFC2104] hash. Each probe query and response messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It can use HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in

IPSec defined in [RFC4868]); hence the length of the HMAC field is 16 octets.

HMAC uses its own key and the mechanism to distribute the HMAC key is outside the scope of this document.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe message MAY include Comp.MBZ (Must Be Zero) variable length field to align the packet on 16 octets boundary.

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 9.

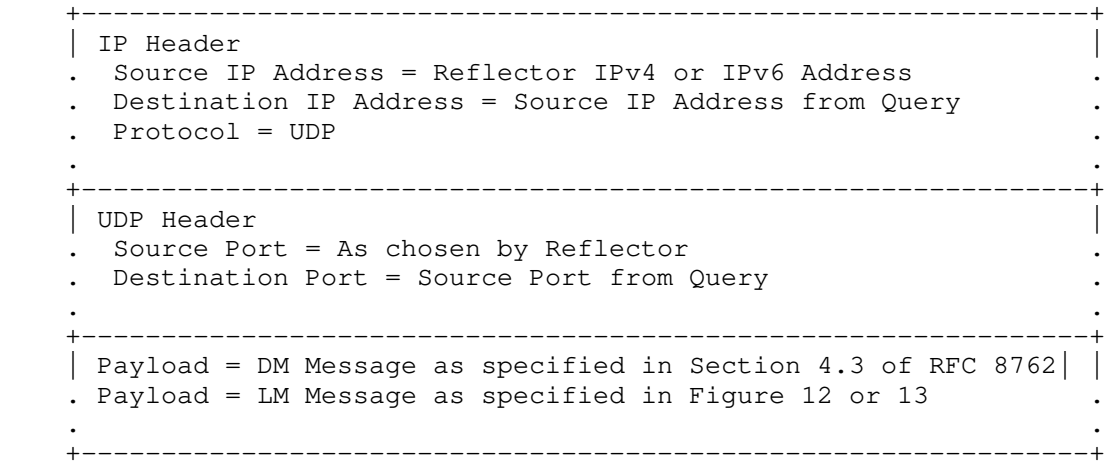


Figure 9: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way performance measurement mode, the probe response message as defined in Figure 9 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. However, only timestamps t1 and t2 are used to measure one-way delay.

4.2.2. Two-way Measurement Mode

In two-way performance measurement mode, when using a bidirectional path, the probe response message as defined in Figure 9 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. All four timestamps are used to measure two-way delay.

Specifically, the probe response message is sent back on the incoming physical interface where the probe query message is received. This is required for example, in case of two-way measurement mode for Link delay.

4.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 10.

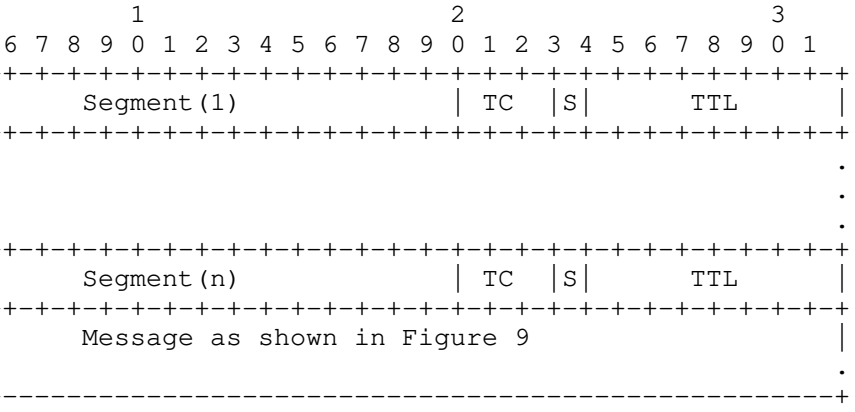


Figure 10: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR Policy in the probe query can be used to find the associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path] to send the probe response message for two-way measurement of SR Policy unless when using STAMP message with Return Path TLV.

4.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way end-to-end performance measurement of an SRv6 Policy with SRH is shown in Figure 11.

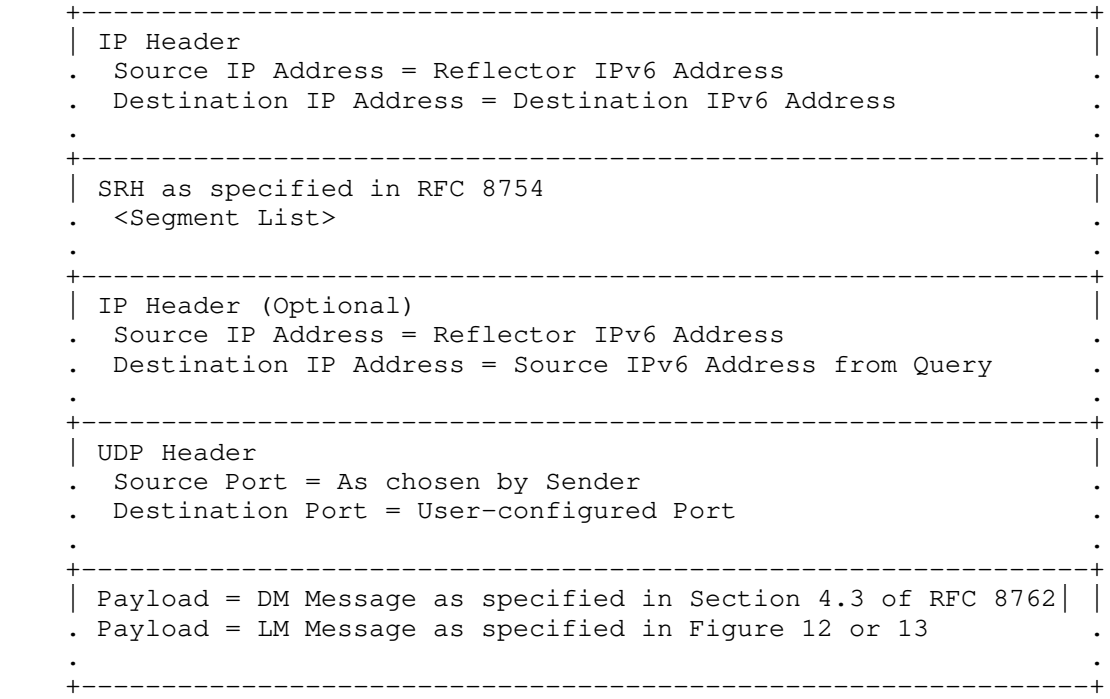


Figure 11: Example Probe Response Message for SRv6 Policy

4.2.3. Loss Measurement Response Message Formats

In this document, STAMP probe response message formats are defined for loss measurement as shown in Figure 12 and Figure 13.

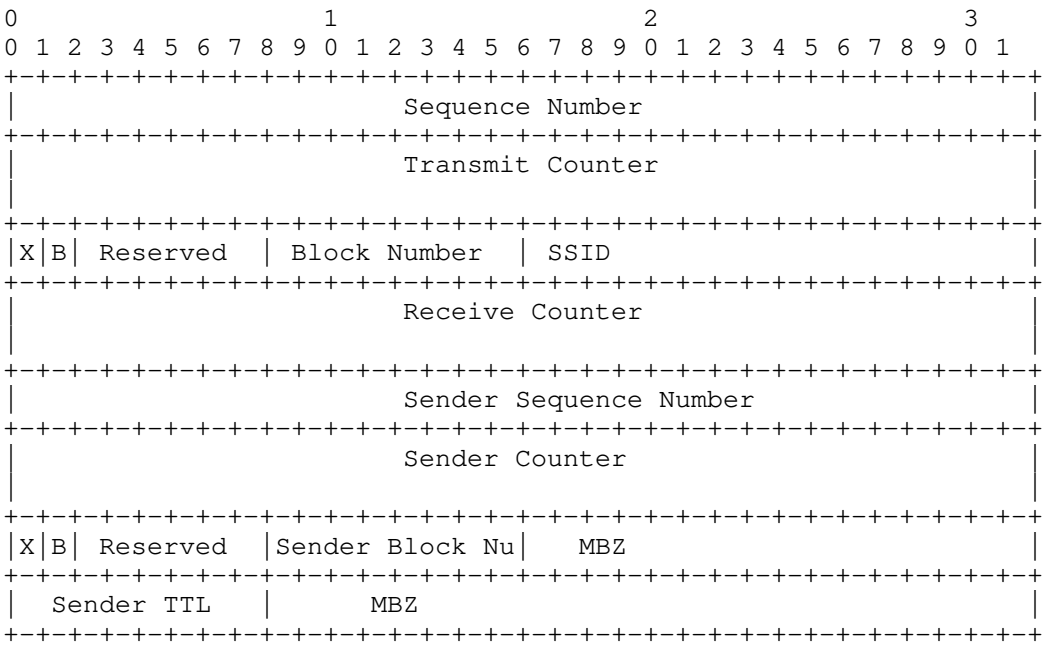


Figure 12: STAMP LM Probe Response Message - Unauthenticated Mode

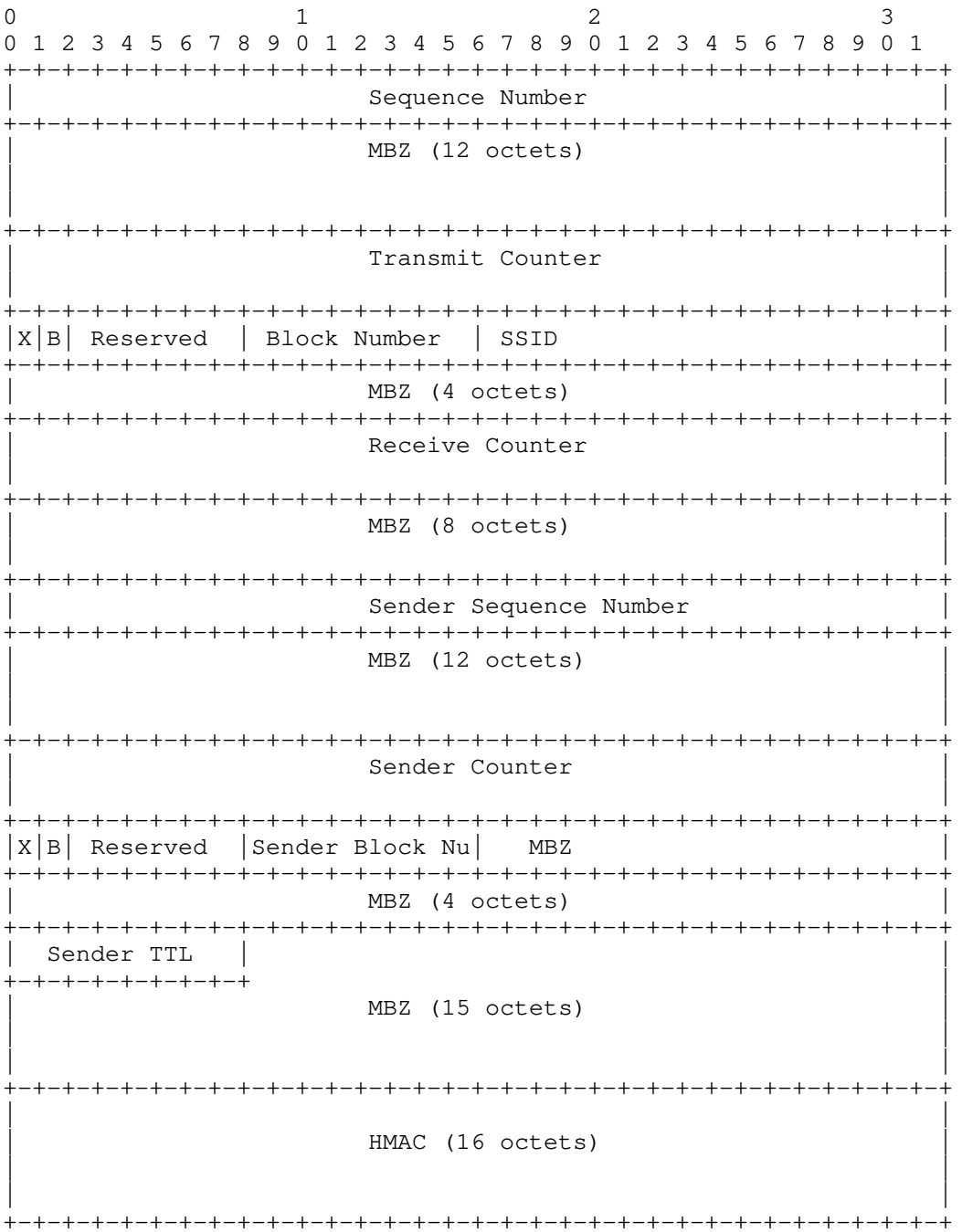


Figure 13: STAMP LM Probe Response Message - Authenticated Mode

4.3. Node Address TLV

In this document, Node Address TLV is defined for STAMP message [I-D.ietf-ippm-stamp-option-tlv] and has the following format shown in Figure 14:

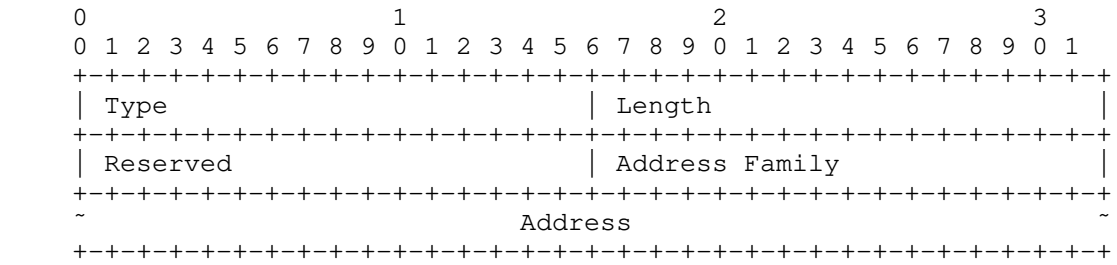


Figure 14: Node Address TLV Format

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

The following Type is defined and it contains Node Address TLV:

Destination Node Address (value TBA1):

The Destination Node Address TLV is optional. The Destination Node Address TLV indicates the address of the intended recipient node of the probe message. The reflector node MUST NOT send response if it is not the intended destination node of the probe query message. This check is useful for example, for performance measurement of SR Policy when using the destination address in 127/8 range for IPv4 or in ::FFFF:127/104 range for IPv6.

4.4. Return Path TLV

For two-way performance measurement, the reflector node needs to send the probe response message on a specific reverse path. The sender node can request in the probe query message to the reflector node to send a response back on a given reverse path (e.g. co-routed bidirectional path). This way the destination node does not require any additional SR Policy state.

For one-way performance measurement, the sender node address may not be reachable via IP route from the reflector node. The sender node in this case needs to send its reachability path information to the reflector node.

[I-D.ietf-ippm-stamp-option-tlv] defines STAMP probe query messages that can include one or more optional TLVs. The TLV Type (value TBA2) is defined in this document for Return Path that carries reverse path for STAMP probe response messages (in the payload of the message). The format of the Return Path TLV is shown in Figure 15:

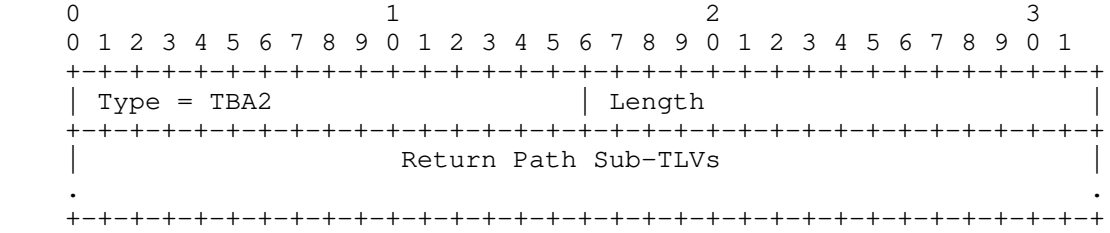


Figure 15: Return Path TLV

The following Type defined for the Return Path TLV contains the Node Address sub-TLV using the format shown in Figure 14:

- Type (value 0): Return Address. Target node address of the response different than the Source Address in the query

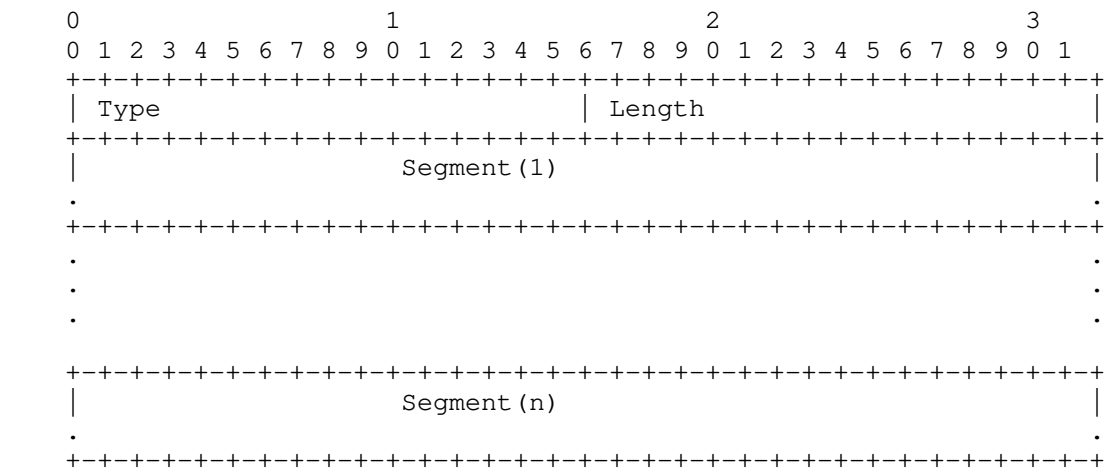


Figure 16: Segment List Sub-TLV in Return Path TLV

The Segment List Sub-TLV (shown in Figure 16) in the Return Path TLV can be one of the following Types:

- Type (value 1): SR-MPLS Label Stack of the Reverse SR Path

- Type (value 2): SR-MPLS Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy
- Type (value 3): SRv6 Segment List of the Reverse SR Path
- Type (value 4): SRv6 Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

The Return Path TLV is optional. The PM sender node MUST only insert one Return Path TLV in the probe query message and the reflector node MUST only process the first Return Path TLV in the probe query message and ignore other Return Path TLVs if present. The reflector node MUST send probe response message back on the reverse path specified in the Return Path TLV and MUST NOT add Return Path TLV in the probe response message.

4.5. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to the STAMP messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.5.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC8762]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC8762].

When using the Destination IPv4 Address from the 127/8 range, the TTL in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.5.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.5.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for PM delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR Policies [I-D.ietf-spring-segment-routing-policy] are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies as following:

- The sender root node sends probe query messages using the Replication Segment defined in [I-D.voyer-spring-sr-replication-segment] for the P2MP SR Policy as shown in Figure 17.

- Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 9. This allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the end-to-end delay and loss performance for each P2MP leaf node of the P2MP SR Policy.

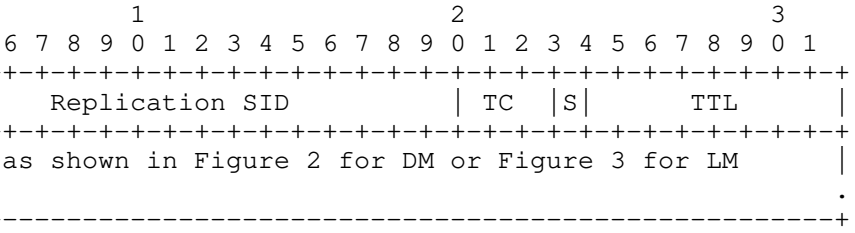


Figure 17: Example Query with Replication Segment for SR-MPLS Policy

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the PM probe messages, sweeping of Destination Address in 127/8 range can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

The procedure defined in this document for delay measurement using the STAMP probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that detection interval and scale for number of sessions need to account for the processing of the probe messages which are punted out of fast path in forwarding (to slow path or control plane), and re-injected back on the reflector node.

8. Security Considerations

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

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, PM probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

IANA will create a "STAMP TLV Type" registry for [I-D.ietf-ippm-stamp-option-tlv]. IANA is requested to allocate a value for the following mandatory Destination Address TLV Type from this registry. This TLV is to be carried in PM probe messages.

- Type TBA1: Destination Node Address TLV

IANA is also requested to allocate a value for the following mandatory Return Path TLV Type from the same registry. This TLV is to be carried in PM probe query messages.

- Type TBA2: Return Path TLV

IANA is requested to create a sub-registry for "Return Path Sub-TLV Type". All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

Value	Description	Reference
0- 32767	Mandatory TLV, unassigned	IETF Review
32768 - 65279	Optional TLV, unassigned	First Come First Served
65280 - 65519	Experimental	This document
65520 - 65534	Private Use	This document
65535	Reserved	This document

Table 1: Return Path Sub-TLV Type Registry

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

- Type (value 0): Return Address

- Type (value 1): SR-MPLS Label Stack of the Reverse SR Path

- Type (value 2): SR-MPLS Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

- Type (value 3): SRv6 Segment List of the Reverse SR Path

- Type (value 4): SRv6 Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

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>.

[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>.

[I-D.ietf-ippm-stamp-option-tlv] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-way Active Measurement Protocol Optional Extensions", draft-ietf-ippm-stamp-option-tlv-04 (work in progress), March 2020.

10.2. Informative References

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

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[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>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[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.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.voyer-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-voyer-spring-sr-replication-segment-03 (work in progress), June 2020.

[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.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-15 (work in progress), March 2020.

[I-D.ietf-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-ietf-pce-binding-label-sid-02 (work in progress), March 2020.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-02 (work in progress), March 2020.

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.

[I-D.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

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document. The authors would like to acknowledge the earlier work on the loss measurement using TWAMP described in draft-xiao-ippm-twamp-ext-direct-loss. The authors would also like to thank Sam Aldrin for the discussions to check for broken path.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 7, 2021

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

August 6, 2020

Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks
draft-gandhi-spring-stamp-srpm-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 specifies procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the mechanisms defined in RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) for Delay Measurement, and uses the mechanisms defined in this document for Loss Measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

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. 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 SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks using probe messages [RFC8762]. It eliminates the need for control-channel signaling by using configuration data model to provision a test-channel (e.g. UDP paths). [I-D.ietf-ippm-stamp-option-tlv] defines TLV extensions for STAMP messages.

The STAMP message with a TLV for "direct measurement" can be used for combined Delay + Loss measurement [I-D.ietf-ippm-stamp-option-tlv]. However, in order to use only for loss measurement purpose, it requires the node to support the delay measurement messages and support timestamp for these messages (which may also require clock synchronization). Furthermore, for hardware-based counter collection for direct-mode loss measurement, the optional TLV based processing adds unnecessary overhead (as counters are not at well-known locations).

This document specifies procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the mechanisms defined in [RFC8762] (STAMP) (including the TLV extensions) for Delay Measurement (DM), and uses the mechanisms defined in this document for Loss Measurement (LM). The procedure specified is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths. This document also defines mechanisms for handling ECMPs of SR Paths for performance delay measurement. Unless otherwise specified, the mechanisms defined in [RFC8762] and [I-D.ietf-ippm-stamp-option-tlv] are not modified by this document.

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

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

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.

SSID: STAMP Session Identifier.

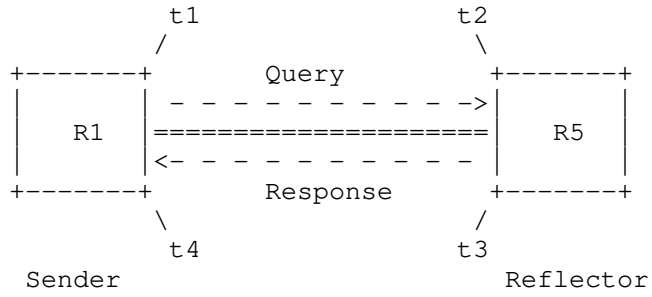
STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a performance measurement probe query message and the reflector node R5 sends a probe response message for the query message received. The probe response message is typically sent to the sender node R1.

SR is enabled on nodes R1 and R5. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R5 (called tail-end).



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC8762] are used. For direct-mode and inferred-mode loss measurements, the messages defined in this document are used. For both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths, no PM state for delay or loss measurement need to be created on the reflector node R5.

Separate UDP destination port numbers are user-configured for delay and loss measurements from the range specified in [RFC8762]. As specified in [RFC8762], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages defined in this document. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. The same destination UDP port is used for Links and SR Paths and the reflector is unaware if the query is for the Links or SR Paths. The number of UDP ports with PM functionality needs to be minimized due to limited hardware resoucres.

For Performance Measurement, probe query and response messages are sent as following:

- For delay measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Paths.

- For loss measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

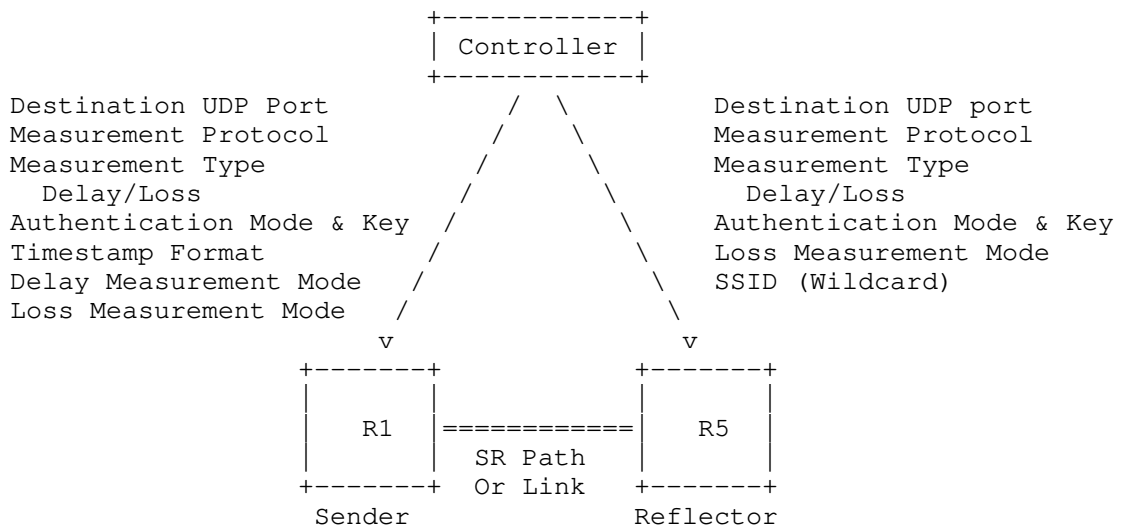


Figure 1: Example Provisioning Model

Example of Measurement Protocol is STAMP, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

The mechanisms to provision the sender and reflector nodes are outside the scope of this document. The provisioning model is not used for signaling the PM parameters between the reflector and sender nodes in SR networks.

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC8762] are used for delay measurement for Links and end-to-end SR Paths including SR Policies. For loss measurement, the probe messages defined in this document are used.

The sender IPv4 or IPv6 address is used as the source address. The reflector IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the address in the range of 127/8 for IPv4 or ::FFFF:127/104 for IPv6 is used as the destination address, respectively.

4.1.1. Delay Measurement Query Message

The message content for delay measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in [RFC8762].

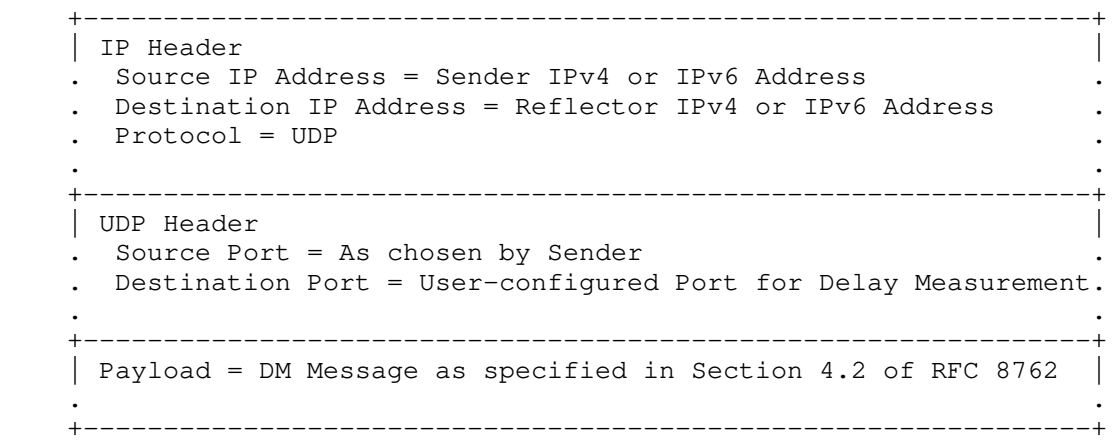


Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC8762]. It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for loss measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in Figure 7 and Figure 8.

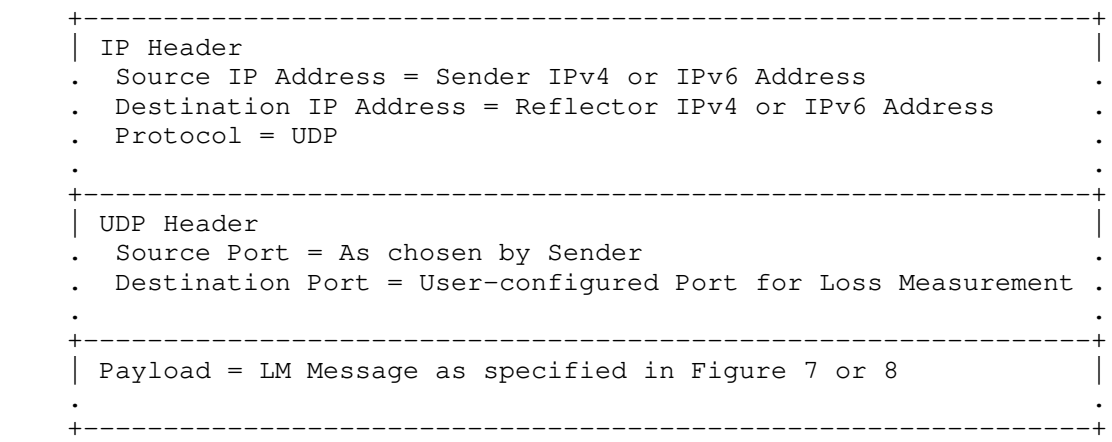


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement are used for Links which may be physical, virtual or LAG (bundle), LAG (bundle) member, numbered/ unnumbered Links. The probe messages are pre-routed over the Link for both delay and loss measurement.

4.1.4. Probe Query for SR Policy

The performance delay and loss measurement for segment routing is applicable to both end-to-end SR-MPLS and SRv6 Policies.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for performance measurement of an end-to-end SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

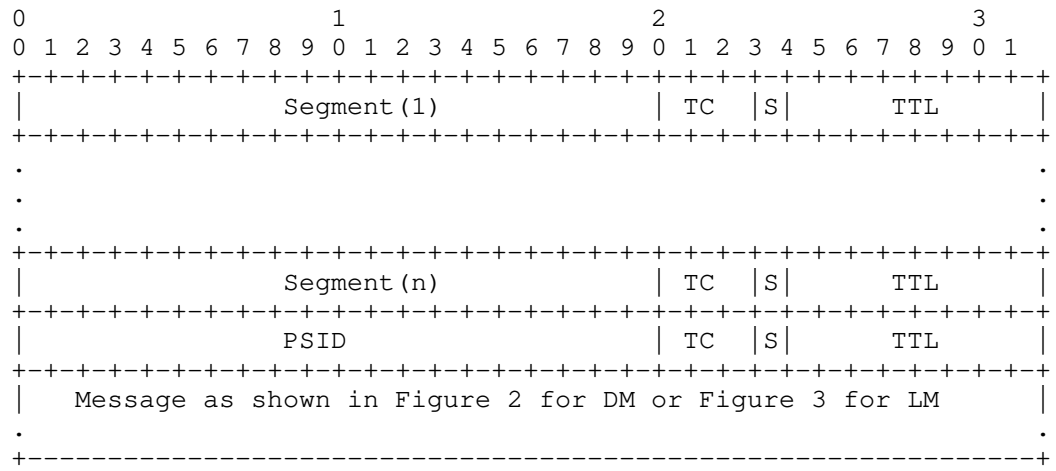


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. The SRv6 network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for performance measurement of an end-to-end SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe query messages.

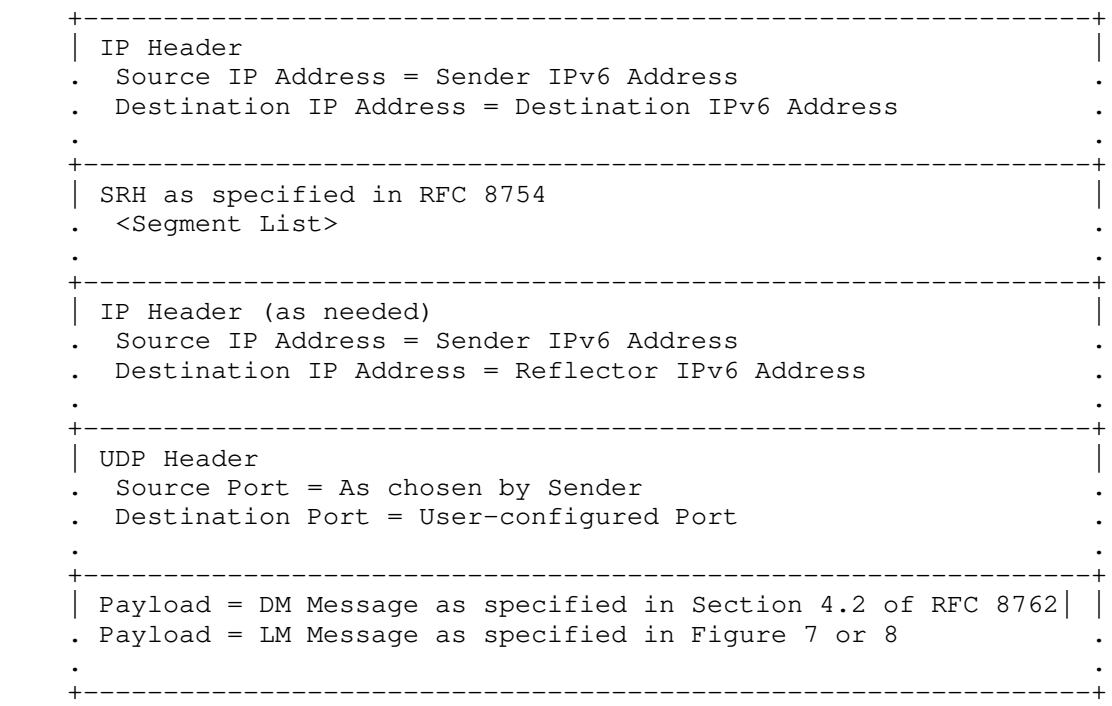


Figure 5: Example Probe Query Message for SRv6 Policy

4.1.5. Control Code Field Extension for STAMP Messages

In this document, the Control Code field is newly defined for delay and loss measurement probe query messages for STAMP protocol in unauthenticated and authenticated modes. The modified delay measurement probe query message format is shown in Figure 6. This message format is backwards compatible with the message format defined in STAMP [RFC8762] as its reflector MUST ignore the received field (previously identified as MBZ). With this field, the reflector node does not require any additional SR state for PM (recall that in SR networks, the state is in the probe packet and signaling of the parameters is avoided). The usage of the Control Code is not limited to the SR paths and can be used for non-SR paths in a network.

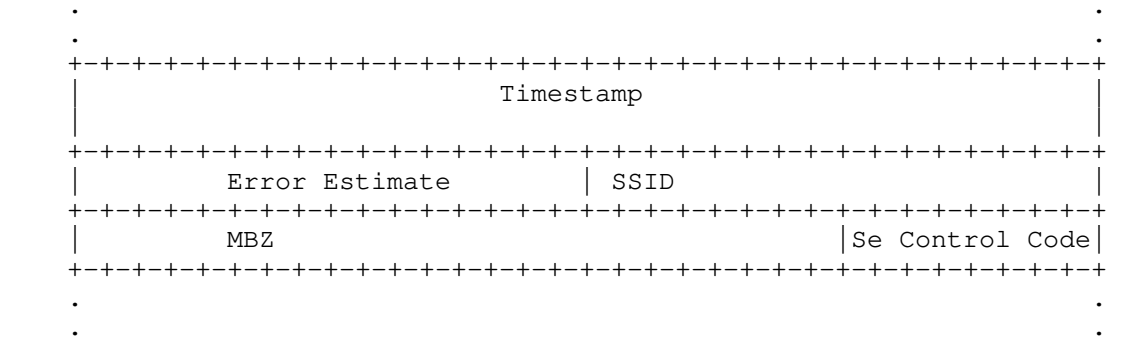


Figure 6: Sender Control Code in STAMP DM Message

Sender Control Code: Set as follows in STAMP probe query message.

In a Query:

0x0: Out-of-band Response Requested. Indicates that the probe response is not required over the same path in the reverse direction. This is also the default behavior.

0x1: In-band Response Requested. Indicates that this query has been sent over a bidirectional path and the probe response is required over the same path in the reverse direction.

0x2: No Response Requested.

4.1.6. Loss Measurement Query Message Extensions

In this document, STAMP probe query messages for loss measurement are defined as shown in Figure 7 and Figure 8. The message formats are hardware efficient due to well-known locations of the counters and payload small in size. They are stand-alone and similar to the delay measurement message formats (e.g. location of the Counter and Timestamp). They also do not require backwards compatibility and support for the existing DM message formats from [RFC8762] as different user-configured destination UDP port is used for loss measurement.

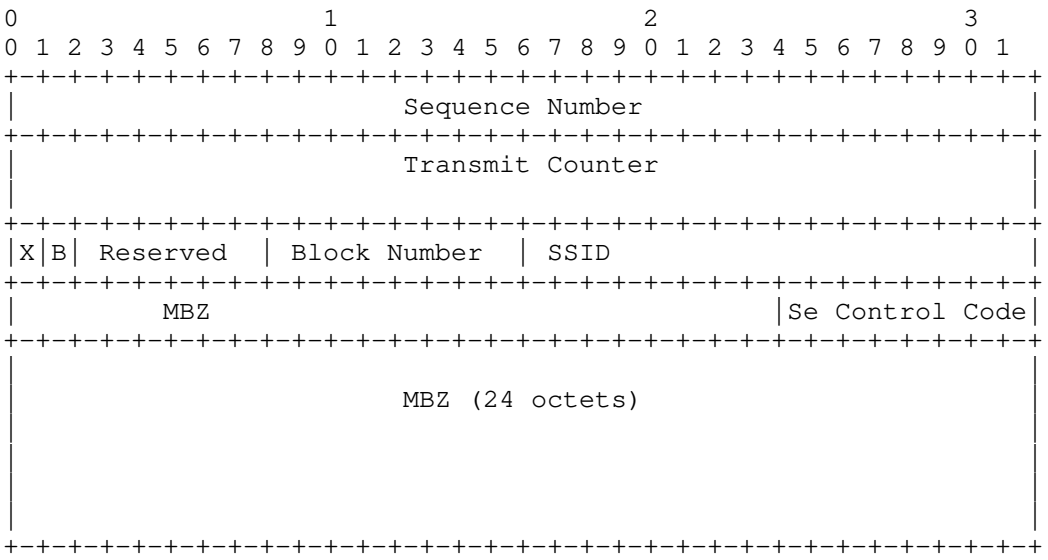


Figure 7: STAMP LM Probe Query Message - Unauthenticated Mode

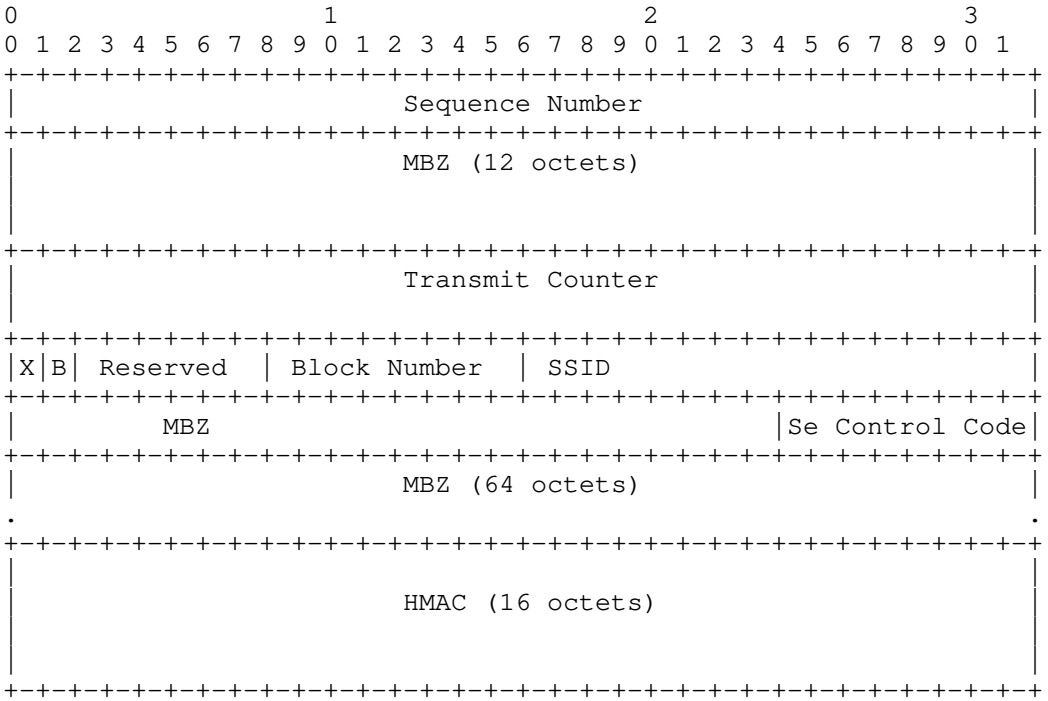


Figure 8: STAMP LM Probe Query Message - Authenticated Mode

Sequence Number (32-bit): As defined in [RFC8762].

Transmit Counter (64-bit): The number of packets or octets sent by the sender node in the query message and by the reflector node in the response message. The counter is always written at the well-known location in the probe query and response messages.

Receive Counter (64-bit): The number of packets or octets received at the reflector node. It is written by the reflector node in the probe response message.

Sender Counter (64-bit): This is the exact copy of the transmit counter from the received query message. It is written by the reflector node in the probe response message.

Sender Sequence Number (32-bit): As defined in [RFC8762].

Sender TTL: As defined in Section 7.1.

LM Flags: The meanings of the Flag bits are:

X: Extended counter format indicator. Indicates the use of extended (64-bit) counter values. Initialized to 1 upon creation (and prior to transmission) of an LM query and copied from an LM query to an LM response message. Set to 0 when the LM message is transmitted or received over an interface that writes 32-bit counter values.

B: Octet (byte) count. When set to 1, indicates that the Counter 1-4 fields represent octet counts. The octet count applies to all packets within the LM scope, and the octet count of a packet sent or received includes the total length of that packet (but excludes headers, labels, or framing of the channel itself). When set to 0, indicates that the Counter fields represent packet counts.

Block Number (8-bit): The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to color the data traffic. To be able to correlate the transmit and receive traffic counters of the matching color, the Block Number (or color) of the traffic counters is carried by the probe query and response messages for loss measurement. The Block Number can also be used to aggregate performance metrics collected.

HMAC: The probe message in authenticated mode includes a key Hashed Message Authentication Code (HMAC) [RFC2104] hash. Each probe query and response messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It can use HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPSec

defined in [RFC4868]); hence the length of the HMAC field is 16 octets.

HMAC uses its own key and the mechanism to distribute the HMAC key is outside the scope of this document.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe message MAY include Comp.MBZ (Must Be Zero) variable length field to align the packet on 16 octets boundary.

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 9.

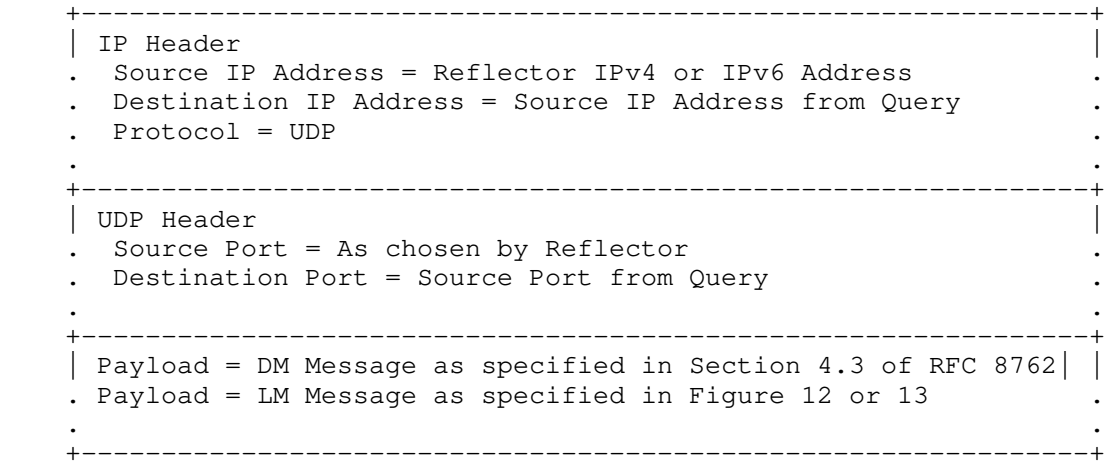


Figure 9: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way measurement mode, the probe response message as defined in Figure 9 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. However, only timestamps t1 and t2 are used to measure one-way delay as (t2 - t1).

4.2.2. Two-way Measurement Mode

In two-way measurement mode, when using a bidirectional path, the probe response message as defined in Figure 9 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. All four timestamps are used to measure two-way delay as ((t4 - t1) - (t3 - t2)).

Specifically, the probe response message is sent back on the incoming physical interface where the probe query message is received. This is required for example, in case of two-way measurement mode for Link delay.

4.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message for two-way performance measurement of an end-to-end SR-MPLS Policy is shown in Figure 10.

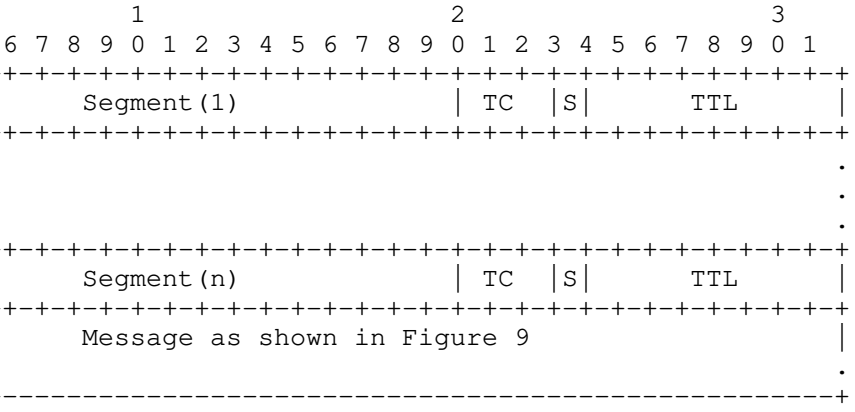


Figure 10: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR Policy in the probe query can be used to find the associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path] to send the probe response message for two-way measurement of SR Policy unless when using STAMP message with Return Path TLV.

4.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way performance measurement of an end-to-end SRv6 Policy with SRH is shown in Figure 11. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe response messages.

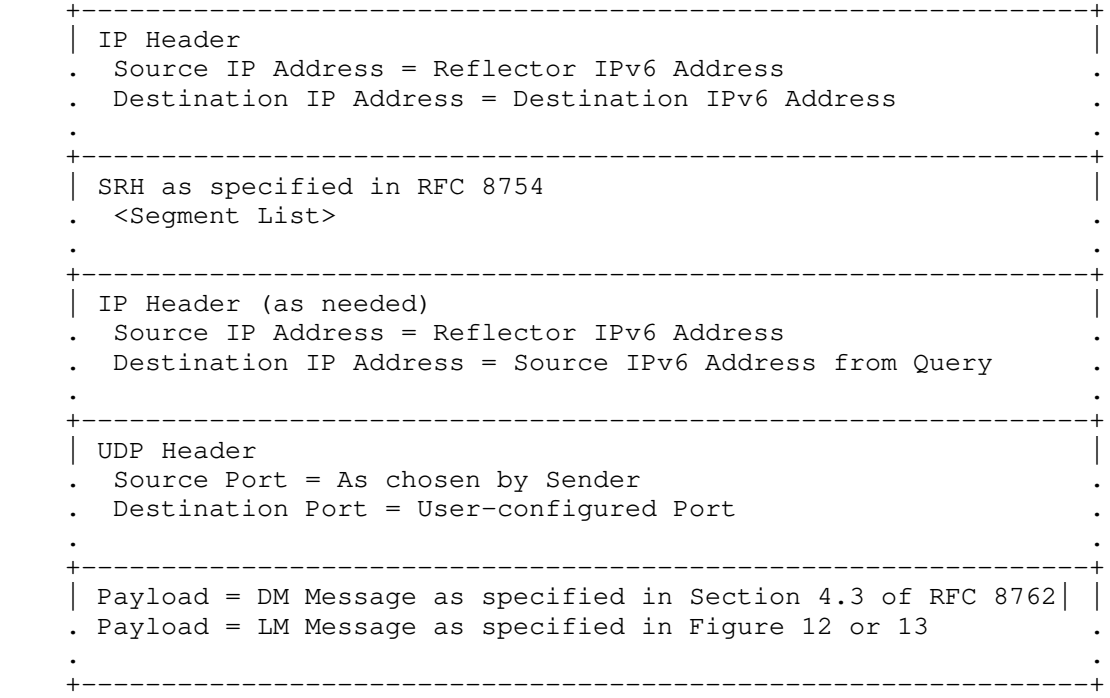


Figure 11: Example Probe Response Message for SRv6 Policy

4.2.3. Loss Measurement Response Message Extensions

In this document, STAMP probe response message formats are defined for loss measurement as shown in Figure 12 and Figure 13.

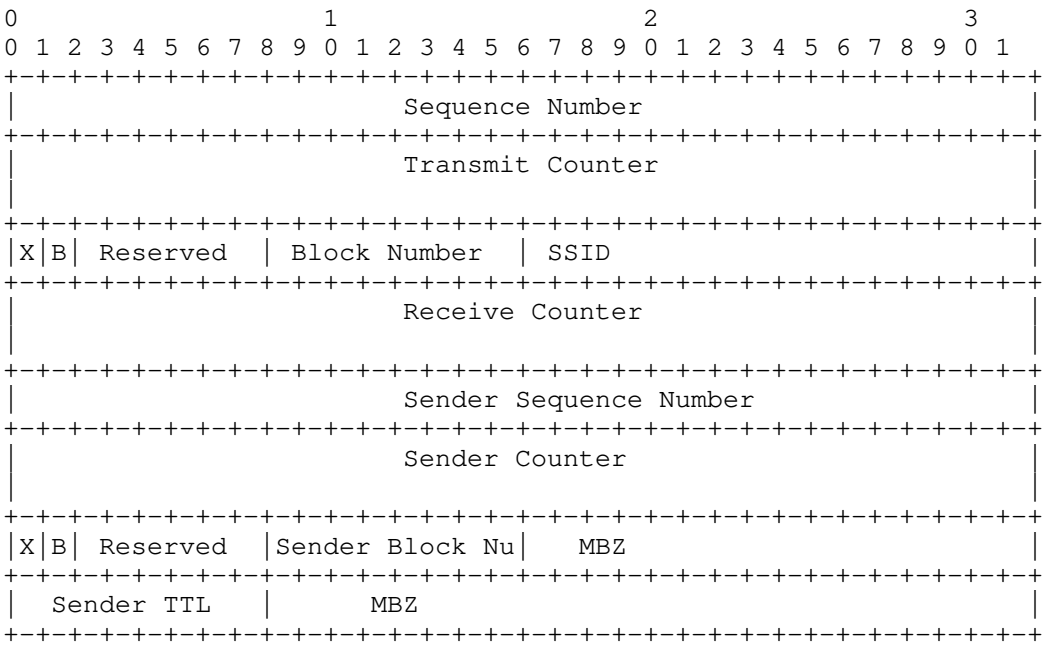


Figure 12: STAMP LM Probe Response Message - Unauthenticated Mode

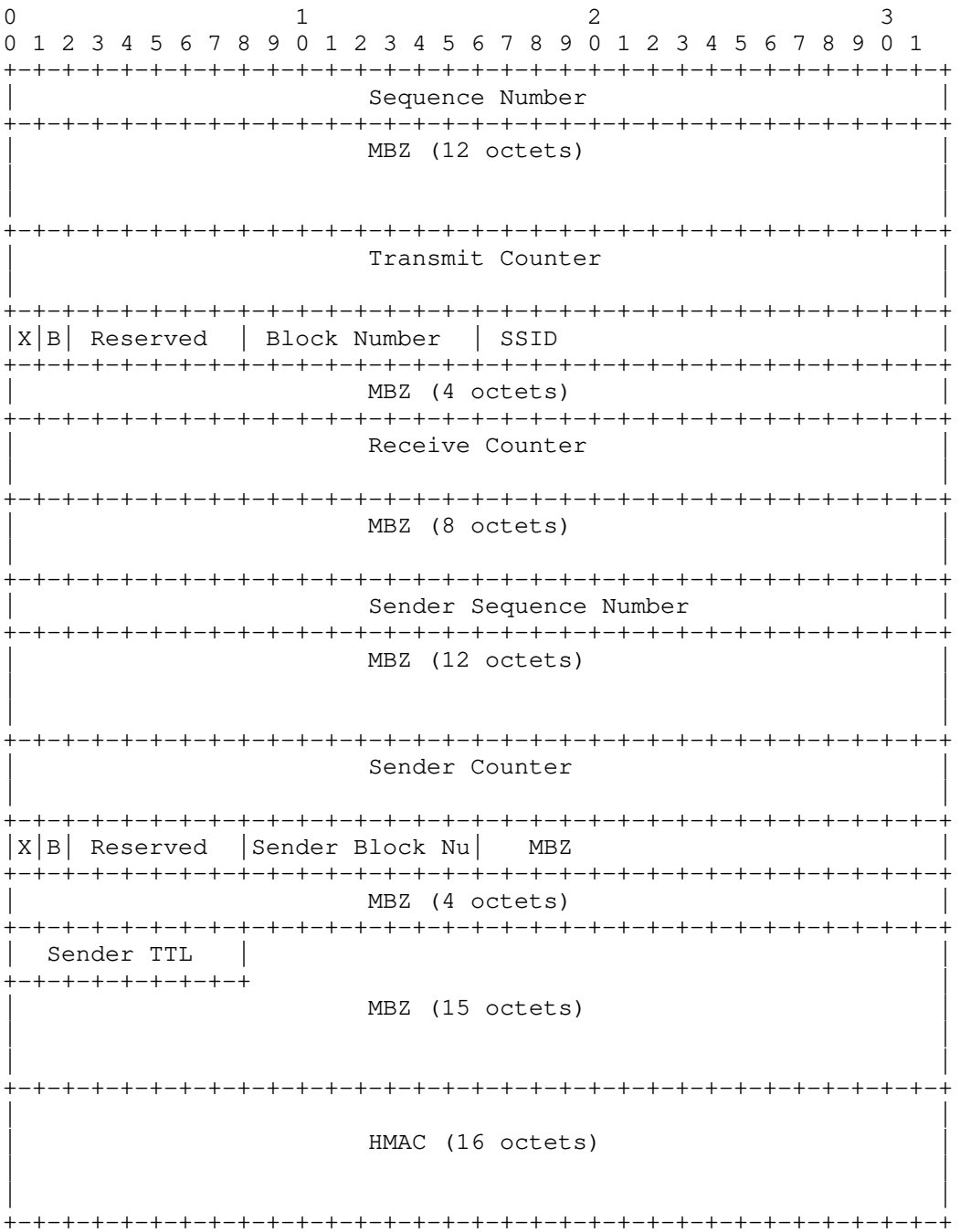


Figure 13: STAMP LM Probe Response Message - Authenticated Mode

4.3. Node Address TLV Extensions

In this document, Node Address TLV is defined for STAMP message [I-D.ietf-ippm-stamp-option-tlv] and has the following format shown in Figure 14:

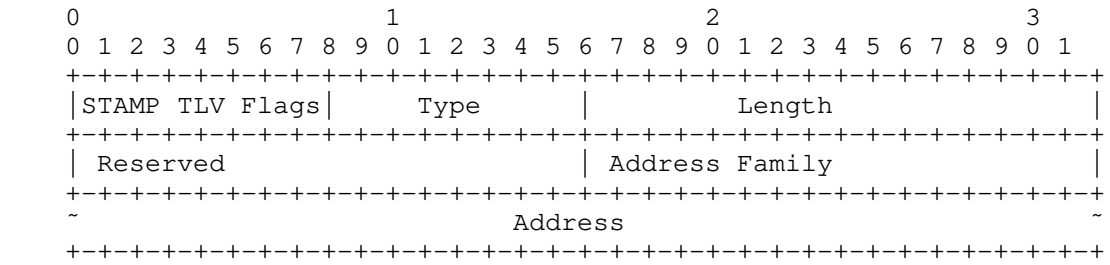


Figure 14: Node Address TLV Format

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

The STAMP TLV Flags are set using the procedures described in [I-D.ietf-ippm-stamp-option-tlv].

The following Type is defined and it contains Node Address TLV:

Destination Node Address (value TBA1):

The Destination Node Address TLV is optional. The Destination Node Address TLV indicates the address of the intended recipient node of the probe message. The reflector node MUST NOT send response message if it is not the intended destination node of the probe query message. This check is useful for example, for performance measurement of SR Policy when using the destination address in 127/8 range for IPv4 or in ::FFFF:127/104 range for IPv6.

4.4. Return Path TLV Extensions

For two-way performance measurement, the reflector node needs to send the probe response message on a specific reverse path. The sender node can request in the probe query message to the reflector node to send a response message back on a given reverse path (e.g. co-routed bidirectional path). This way the reflector node does not require any additional SR state for PM (recall that in SR networks, the state is in the probe packet and signaling of the parameters is avoided).

For one-way performance measurement, the sender node address may not be reachable via IP route from the reflector node. The sender node in this case needs to send its reachability path information to the reflector node.

[I-D.ietf-ippm-stamp-option-tlv] defines STAMP probe query messages that can include one or more optional TLVs. The TLV Type (value TBA2) is defined in this document for Return Path that carries reverse path for STAMP probe response messages (in the payload of the message). The format of the Return Path TLV is shown in Figure 15:

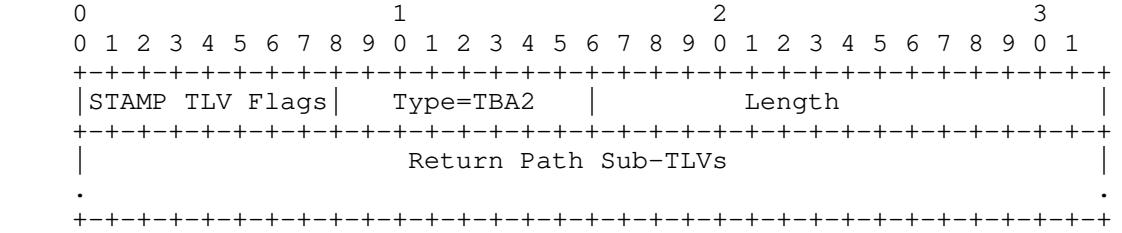


Figure 15: Return Path TLV

The STAMP TLV Flags are set using the procedures described in [I-D.ietf-ippm-stamp-option-tlv].

The following Type defined for the Return Path TLV contains the Node Address sub-TLV using the format shown in Figure 14:

- Type (value 0): Return Address. Target node address of the response message different than the Source Address in the query

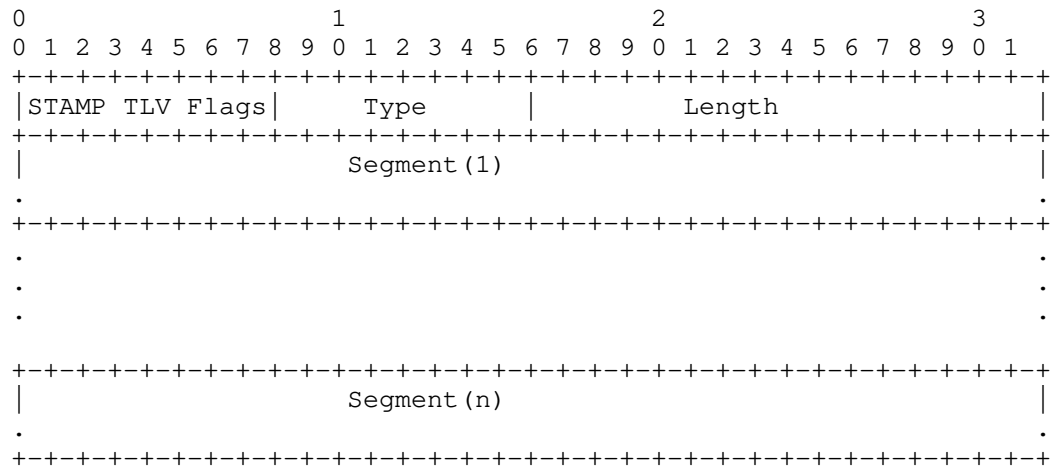


Figure 16: Segment List Sub-TLV in Return Path TLV

The Segment List Sub-TLV (shown in Figure 16) in the Return Path TLV can be one of the following Types:

- Type (value 1): SR-MPLS Label Stack of the Reverse Path
- Type (value 2): SR-MPLS Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy
- Type (value 3): SRv6 Segment List of the Reverse Path
- Type (value 4): SRv6 Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

The Return Path TLV is optional. The sender node MUST only insert one Return Path TLV in the probe query message and the reflector node MUST only process the first Return Path TLV in the probe query message and ignore other Return Path TLVs if present. The reflector node MUST send probe response message back on the reverse path specified in the Return Path TLV and MUST NOT add Return Path TLV in the probe response message.

4.5. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to the STAMP messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.5.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC8762]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC8762].

When using the Destination IPv4 Address from the 127/8 range, the TTL in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.5.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.5.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for PM delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR Path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy] or P2MP Transport [I-D.shen-spring-p2mp-transport-chain]).

The procedures for delay and loss measurement described in this document for P2P SR Policies are also equally applicable to the P2MP SR Policies. The procedure for one-way measurement is defined as following:

- The sender root node sends probe query messages using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 17.

- The probe query messages can contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].

- Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 17. This allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the delay and loss performance for each P2MP leaf node of the end-to-end P2MP SR Policy.

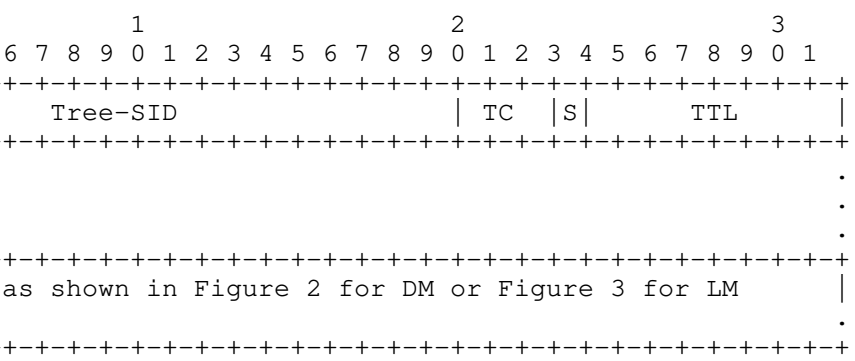


Figure 17: Example Probe Query with Tree-SID for SR-MPLS Policy

The probe query messages can also be sent using the scheme defined for P2MP Transport using Chain Replication that may contain Bud SID as defined in [I-D.shen-spring-p2mp-transport-chain].

The considerations for two-way mode for performance measurement for P2MP SR Policy (e.g. for bidirectional SR Path) are outside the scope of this document.

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the probe messages, sweeping of Destination Address in 127/8 range can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

Liveness monitoring is required for connectivity verification and continuity check in an SR network. The procedure defined in this document for delay measurement using the STAMP probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that for one-way and two-way modes, the failure detection interval and scale for number of probe messages need to account for the processing of the probe query messages which need to be punted from the forwarding fast path (to slow path or control plane) and response messages need to be injected on the reflector node. This is enhanced by using the probes in loopback mode as described in [I-D.gandhi-spring-sr-enhanced-plm].

8. Security Considerations

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

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, probe messages for SRv6 may not need authentication mode. Cryptographic measures may be

enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

IANA will create a "STAMP TLV Type" registry for [I-D.ietf-ippm-stamp-option-tlv]. IANA is requested to allocate a value for the following Destination Address TLV Type from the IETF Review TLV range of this registry. This TLV is to be carried in the probe messages.

- Type TBA1: Destination Node Address TLV

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

- Type TBA2: Return Path TLV

IANA is requested to create a sub-registry for "Return Path Sub-TLV Type". All code points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to Table 1:

Value	Description	Reference
0	Reserved	This document
1 - 175	Unassigned	This document
176 - 239	Unassigned	This document
240 - 251	Experimental	This document
252 - 254	Private Use	This document
255	Reserved	This document

Table 1: Return Path Sub-TLV Type Registry

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

- Type (value 1): Return Address

- Type (value 2): SR-MPLS Label Stack of the Reverse Path

- Type (value 3): SR-MPLS Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy
- Type (value 4): SRv6 Segment List of the Reverse Path
- Type (value 5): SRv6 Binding SID [I-D.ietf-pce-binding-label-sid] of the Reverse SR Policy

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>.

[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>.

[I-D.ietf-ippm-stamp-option-tlv] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-way Active Measurement Protocol Optional Extensions", draft-ietf-ippm-stamp-option-tlv-08 (work in progress), August 2020.

10.2. Informative References

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

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[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>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[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.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[I-D.ietf-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-00 (work in progress), July 2020.

[I-D.shen-spring-p2mp-transport-chain] Shen, Y., Zhang, Z., Parekh, R., Bidgoli, H., and Y. Kamite, "Point-to-Multipoint Transport Using Chain Replication in Segment Routing", draft-shen-spring-p2mp-transport-chain-02 (work in progress), April 2020.

[I-D.ietf-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.

[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.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-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.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-02 (work in progress), March 2020.

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.

[I-D.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.

[I-D.gandhi-spring-sr-enhanced-plm]
Gandhi, R., Filsfils, C., Vaghamshi, N., Nagarajah, M., and R. Foote, "Enhanced Performance Delay and Liveness Monitoring in Segment Routing Networks", draft-gandhi-spring-sr-enhanced-plm-02 (work in progress), July 2020.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document. The authors would like to acknowledge the earlier work on the loss measurement using TWAMP described in draft-xiao-ippm-twamp-ext-direct-loss. The authors would also like to thank Sam Aldrin for the discussions to check for broken path.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: December 7, 2020

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

June 5, 2020

Performance Measurement Using TWAMP Light for Segment Routing Networks
draft-gandhi-spring-twamp-srpm-09

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 describes procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the messages defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP) Light) for Delay Measurement, and uses the messages defined in this document for Loss Measurement. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 7, 2020.

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. 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 SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

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. These protocols rely on control-channel signaling to establish a test-channel over an UDP path. The TWAMP Light [Appendix I in RFC5357] [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer IP networks by provisioning UDP paths and eliminates the need for control-channel signaling. As described in Appendix A of [RFC8545], TWAMP Light mechanism is informative only. These protocols lack support for direct-mode Loss Measurement (LM) to detect actual Customer data traffic loss which is required in SR networks.

This document describes procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the messages defined in [RFC5357] (TWAMP Light) for Delay Measurement (DM), and uses the messages defined in this document for Loss Measurement. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies. This document also defines mechanisms for handling ECMPs of SR Paths for performance delay measurement. Unless otherwise described, the messages defined in [RFC5357] are not modified by this document.

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

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

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.

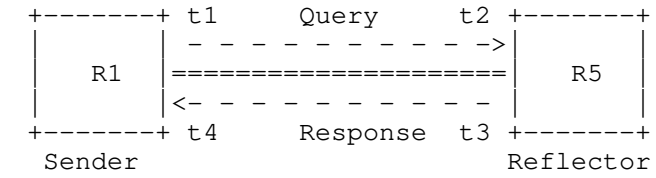
TC: Traffic Class.

TWAMP: Two-Way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a probe query for performance measurement and the reflector node R5 sends a probe response for the query message received. The probe response is sent to the sender node R1. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Paths e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on

node R1 with destination to node R5. In case of Point-to-Multipoint (P2MP), SR Policy originating from source node R1 may terminate on multiple destination leaf nodes [I-D.voyer-spring-sr-replication-segment].



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC5357] are used. For direct-mode and inferred-mode loss measurements in Segment Routing networks, the messages defined in this document are used. Separate UDP destination port numbers are user-configured for delay and loss measurements. As specified in [RFC8545], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages defined in this document. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. For both Links and end-to-end SR Paths including SR Policies, no PM session for delay or loss measurement is created on the reflector node R5 [RFC5357].

For Performance Measurement, probe query and response messages are sent as following:

- For Delay Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to measure the delay experienced by the actual data traffic flowing on the Links and SR Policies.

- For Loss Measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM session state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for

SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

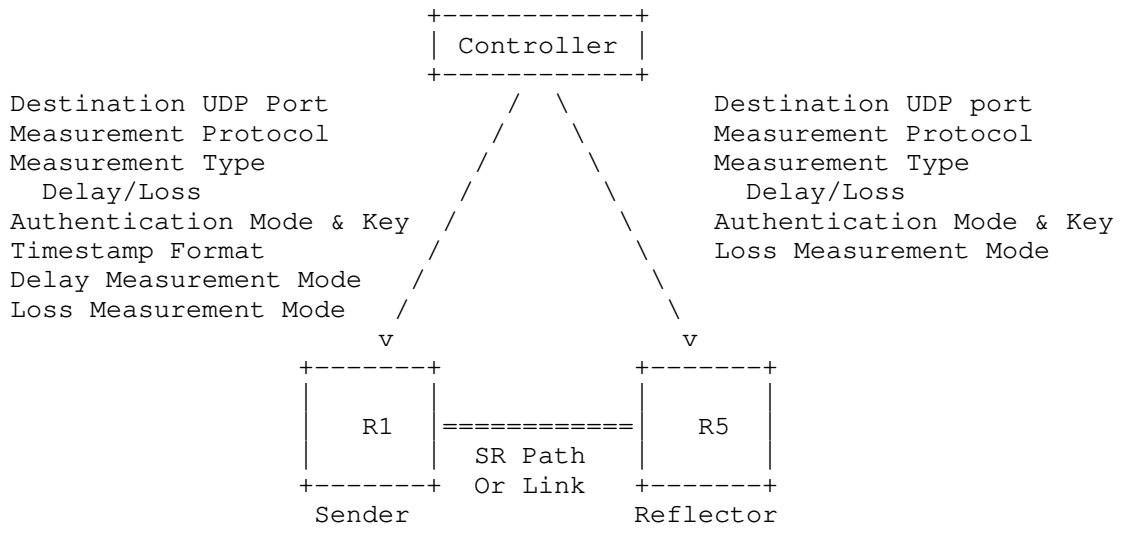


Figure 1: Example Provisioning Model

Example of Measurement Protocol is TWAMP Light, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

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

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC5357] are used for Delay Measurement for Links and end-to-end SR Paths including SR Policies. For Loss Measurement, the probe messages defined in this document are used.

The Sender IPv4 or IPv6 address is used as the source address. The reflector IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the address in the range of 127/8 for IPv4 or ::FFFF:127/104 for IPv6 is used as the destination address, respectively.

4.1.1. Delay Measurement Query Message

The message content for Delay Measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port for DM, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in Section 4.1.2 of [RFC5357]. For symmetrical size query and response messages as defined in [RFC6038], the DM probe query message contains the payload format defined in Section 4.2.1 of [RFC5357].

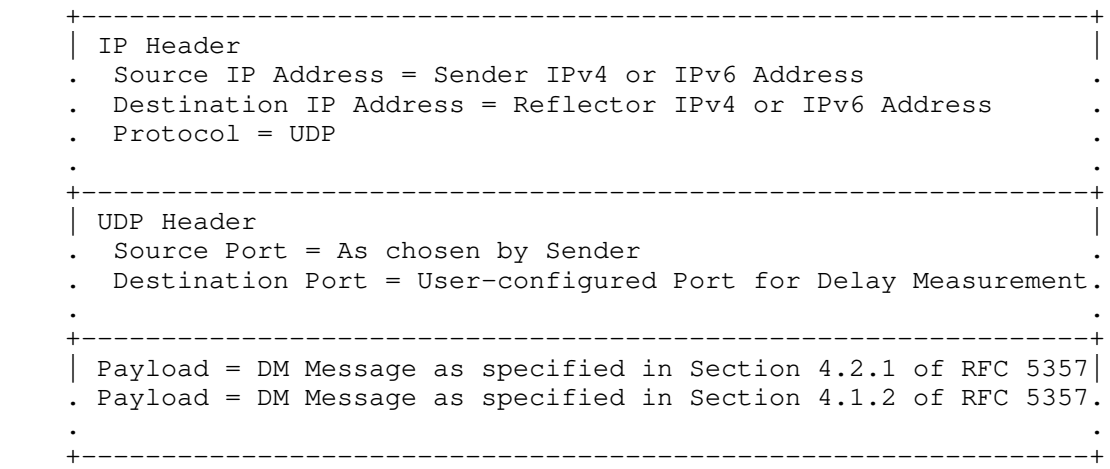


Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC5357]. It is recommended to use the IEEE 1588v2

Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for Loss Measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port for LM, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in Figure 7 and Figure 8.

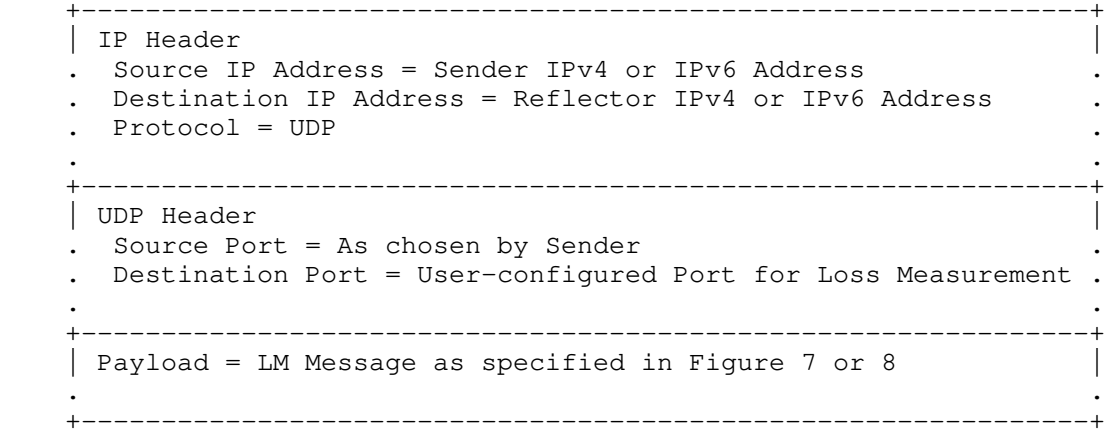


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured

destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement is sent on the congruent path of the data traffic. The probe messages are routed over the Link for both delay and loss measurement.

4.1.4. Probe Query for End-to-end Measurement for SR Policy

The performance delay and loss measurement for segment routing is applicable to both SR-MPLS and SRv6 Policies.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for end-to-end performance measurement of an SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

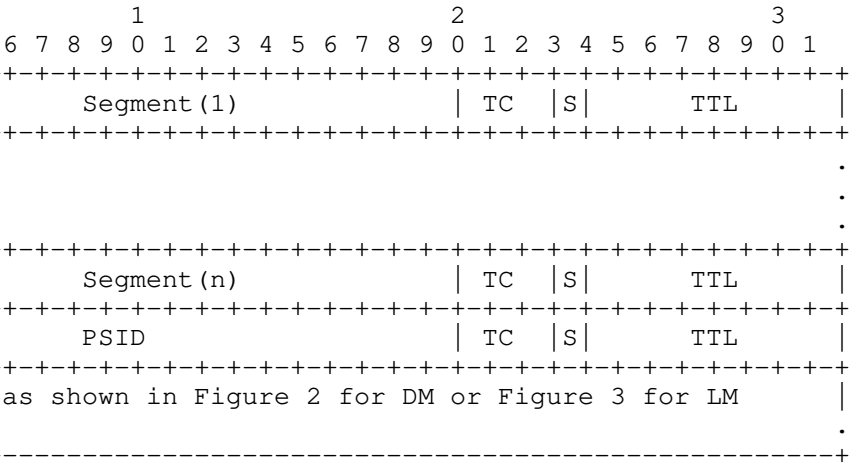


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. For SRv6, network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for end-to-end performance measurement of an SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5.

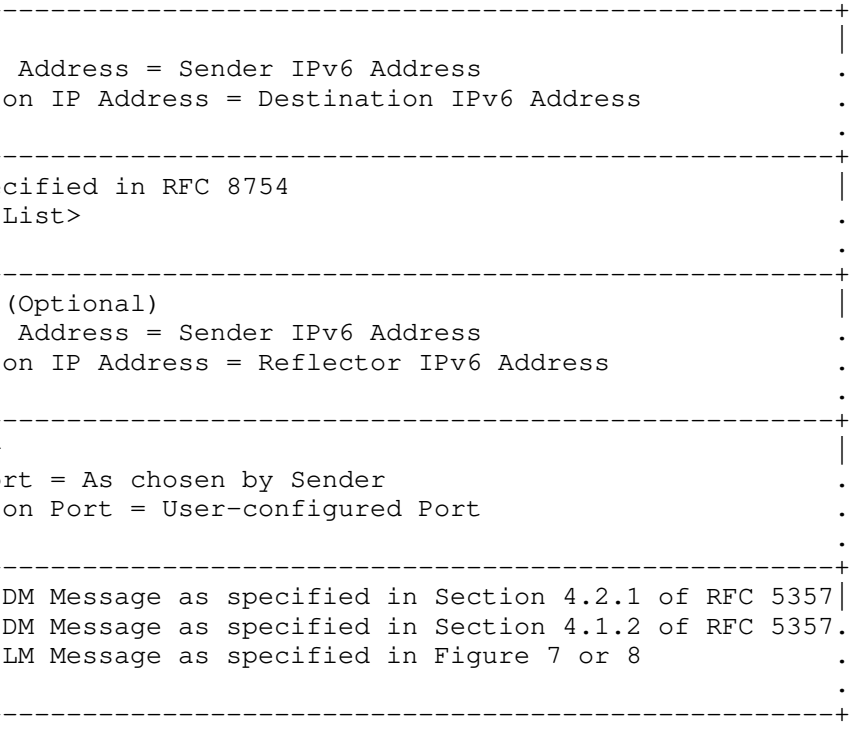


Figure 5: Example Probe Query Message for SRv6 Policy

4.1.5. Control Code Field for TWAMP Light Messages

The Control Code field is defined for delay and loss measurement probe query messages for TWAMP Light in unauthenticated and authenticated modes. The modified delay measurement probe query message format is shown in Figure 6. This message format is backwards compatible with the message format defined in [RFC5357] as its reflectors ignore the received field (previously identified as MBZ). The usage of the Control Code is not limited to the SR paths and can be used for non-SR paths in a network.

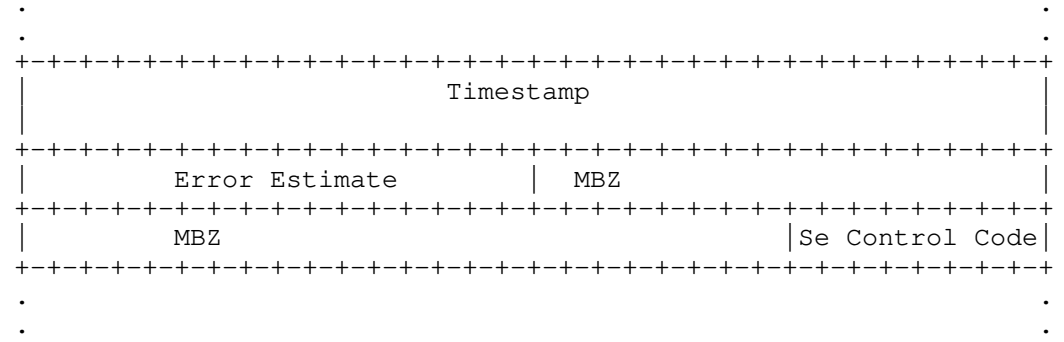


Figure 6: Sender Control Code in TWAMP Light DM Message

Sender Control Code: Set as follows in TWAMP Light probe query message.

In a Query:

0x0: Out-of-band Response Requested. Indicates that the probe response is not required over the same path in the reverse direction. This is also the default behavior.

0x1: In-band Response Requested. Indicates that this query has been sent over a bidirectional path and the probe response is required over the same path in the reverse direction.

0x2: No Response Requested.

4.1.6. Loss Measurement Query Message Formats

In this document, TWAMP Light probe query messages for loss measurement are defined as shown in Figure 7 and Figure 8. The message formats are hardware efficient due to well-known locations of the counters and payload small in size. They are stand-alone and similar to the delay measurement message formats (e.g. location of the Counter and Timestamp). They also do not require backwards compatibility and support for the existing DM message formats from [RFC5357] as different user-configured destination UDP port is used for loss measurement.

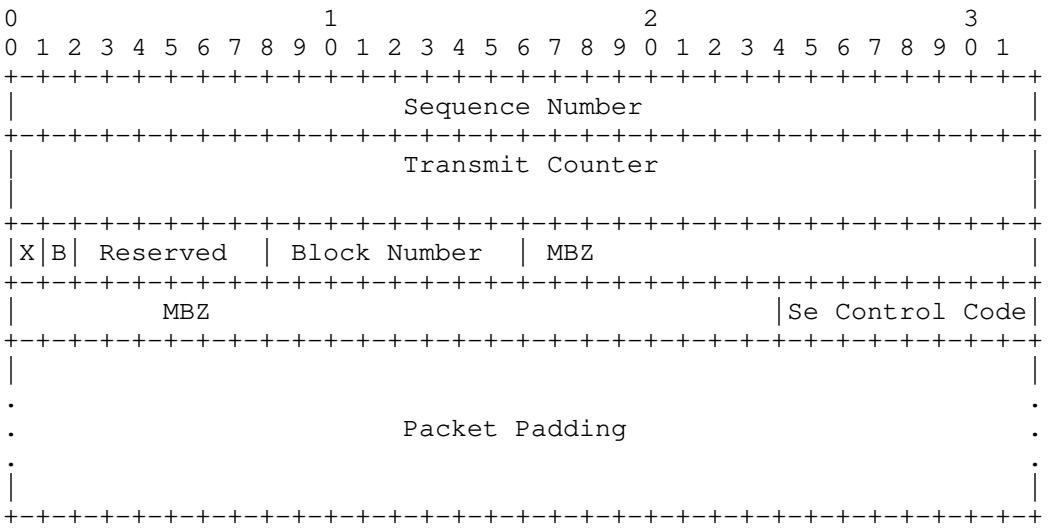


Figure 7: TWAMP Light LM Probe Query Message - Unauthenticated Mode

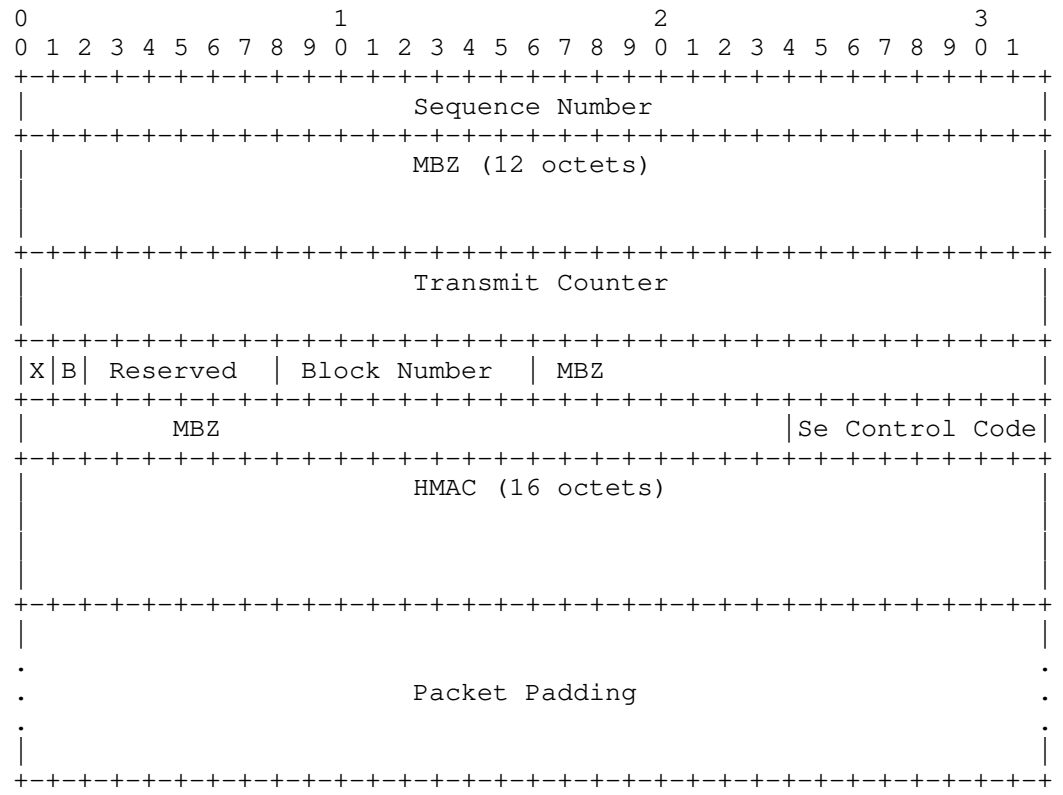


Figure 8: TWAMP Light LM Probe Query Message - Authenticated Mode

Sequence Number (32-bit): As defined in [RFC5357].

Transmit Counter (64-bit): The number of packets or octets sent by the sender node in the query message and by the reflector node in the response message. The counter is always written at the well-known location in the probe query and response messages.

Receive Counter (64-bit): The number of packets or octets received at the reflector node. It is written by the reflector node in the probe response message.

Sender Counter (64-bit): This is the exact copy of the transmit counter from the received query message. It is written by the reflector node in the probe response message.

Sender Sequence Number (32-bit): As defined in [RFC5357].

Sender TTL: As defined in Section 7.1.

LM Flags: The meanings of the Flag bits are:

X: Extended counter format indicator. Indicates the use of extended (64-bit) counter values. Initialized to 1 upon creation (and prior to transmission) of an LM Query and copied from an LM Query to an LM response. Set to 0 when the LM message is transmitted or received over an interface that writes 32-bit counter values.

B: Octet (byte) count. When set to 1, indicates that the Counter 1-4 fields represent octet counts. The octet count applies to all packets within the LM scope, and the octet count of a packet sent or received includes the total length of that packet (but excludes headers, labels, or framing of the channel itself). When set to 0, indicates that the Counter fields represent packet counts.

Block Number (8-bit): The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to color the data traffic. To be able to compare the transmit and receive traffic counters of the matching color, the Block Number (or color) of the traffic counters is carried by the probe query and response messages for loss measurement.

HMAC: The PM probe message in authenticated mode includes a key Hashed Message Authentication Code (HMAC) [RFC2104] hash. Each probe query and response messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It can use HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPSec defined in [RFC4868]); hence the length of the HMAC field is 16 octets.

HMAC uses its own key and the mechanism to distribute the HMAC key is outside the scope of this document.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe message may include Comp.MBZ (Must Be Zero) variable length field to align the packet on 16 octets boundary.

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 9.

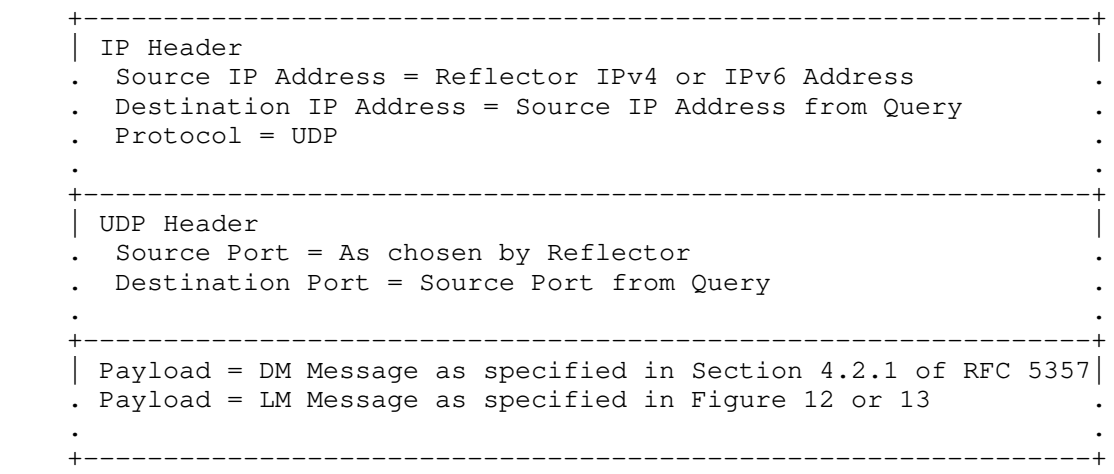


Figure 9: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way performance measurement mode, the probe response message as defined in Figure 9 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. However, only timestamps t1 and t2 are used to measure one-way delay.

4.2.2. Two-way Measurement Mode

In two-way performance measurement mode, when using a bidirectional path, the probe response message as defined in Figure 9 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. All four timestamps are used to measure two-way delay.

Specifically, the probe response message is sent back on the incoming physical interface where the probe query message is received. This is required for example, in case of two-way measurement mode for Link delay.

4.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 10.

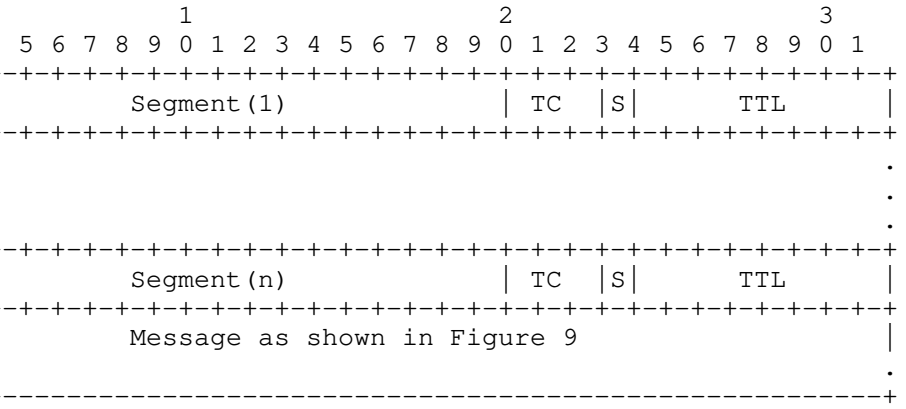


Figure 10: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR Policy in the probe query can be used to find the associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path] to send the probe response message for two-way measurement of SR Policy.

4.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way end-to-end performance measurement of an SRv6 Policy with SRH is shown in Figure 11.

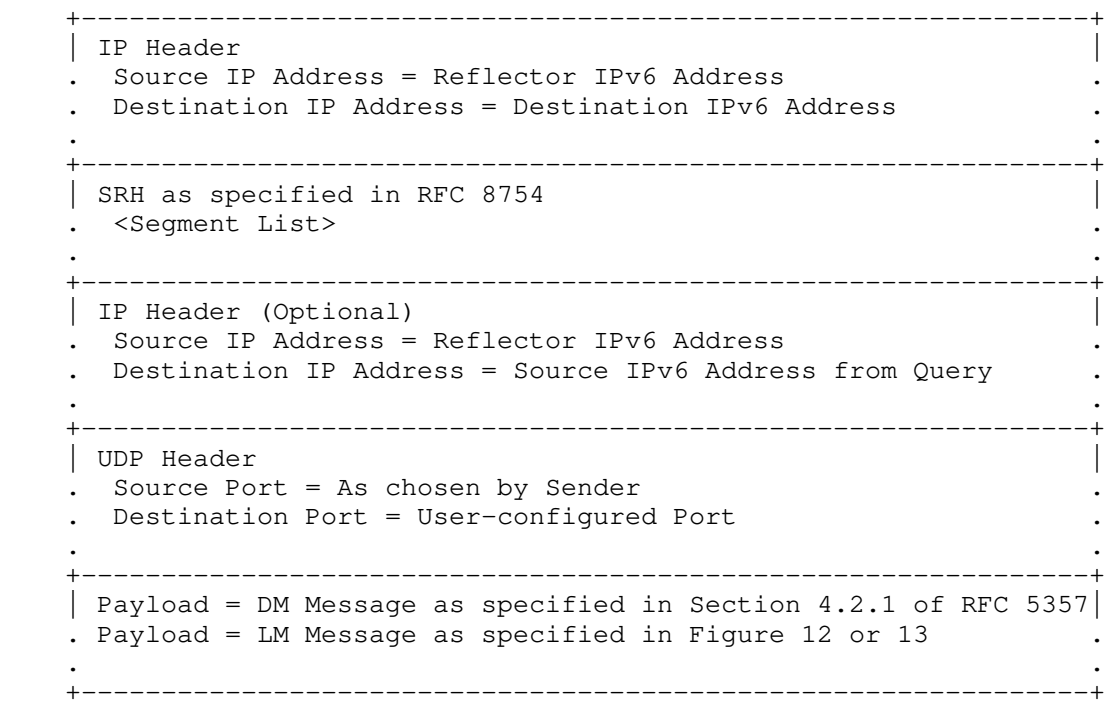


Figure 11: Example Probe Response Message for SRv6 Policy

4.2.3. Loss Measurement Response Message Formats

In this document, TWAMP Light probe response message formats are defined for loss measurement as shown in Figure 12 and Figure 13.

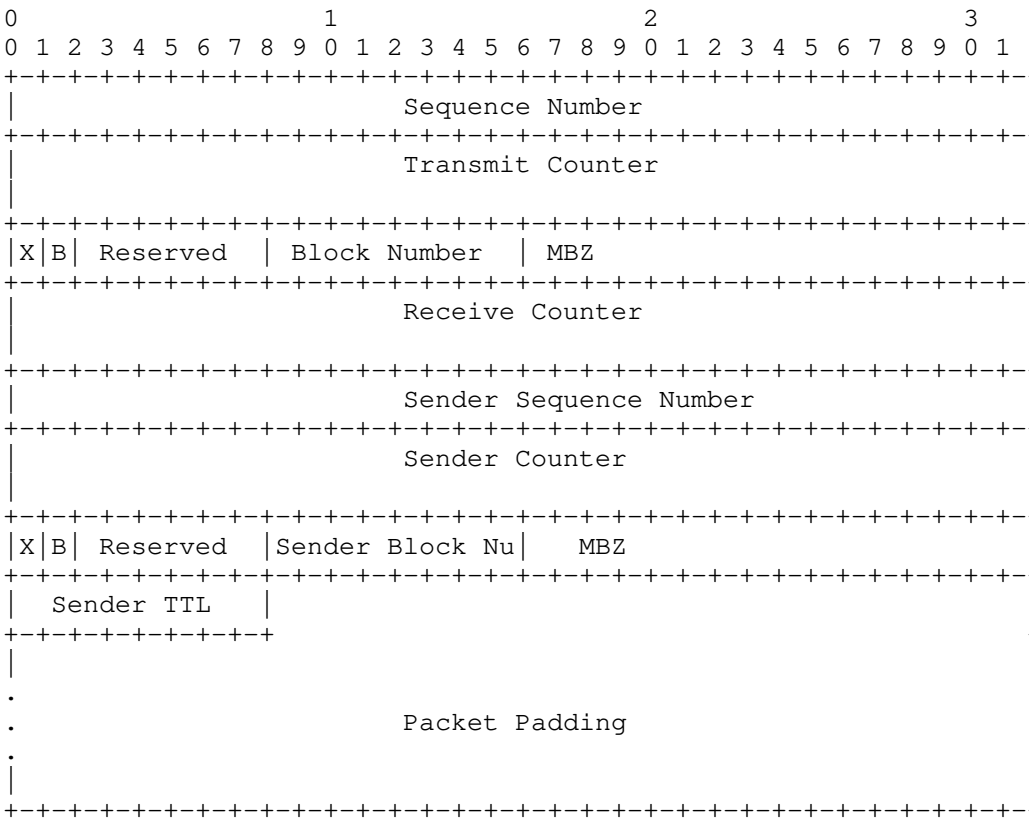
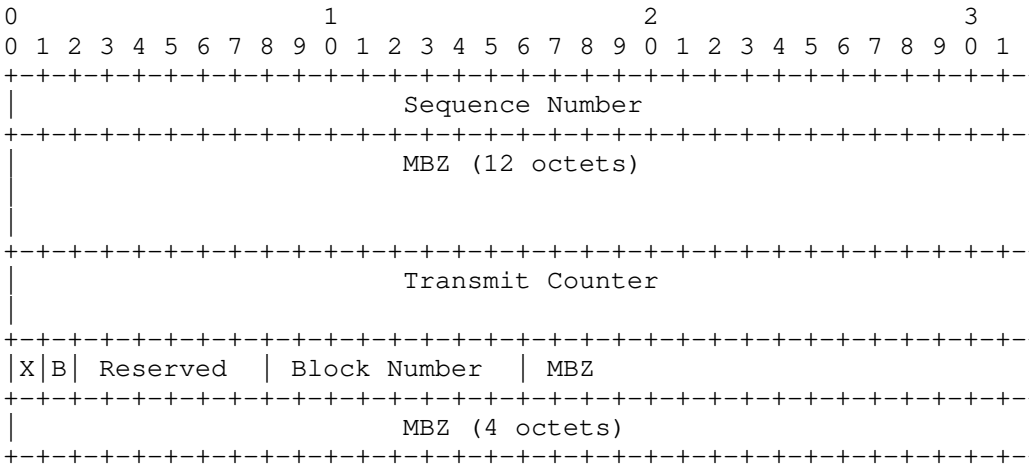


Figure 12: TWAMP Light LM Probe Response Message - Unauthenticated Mode



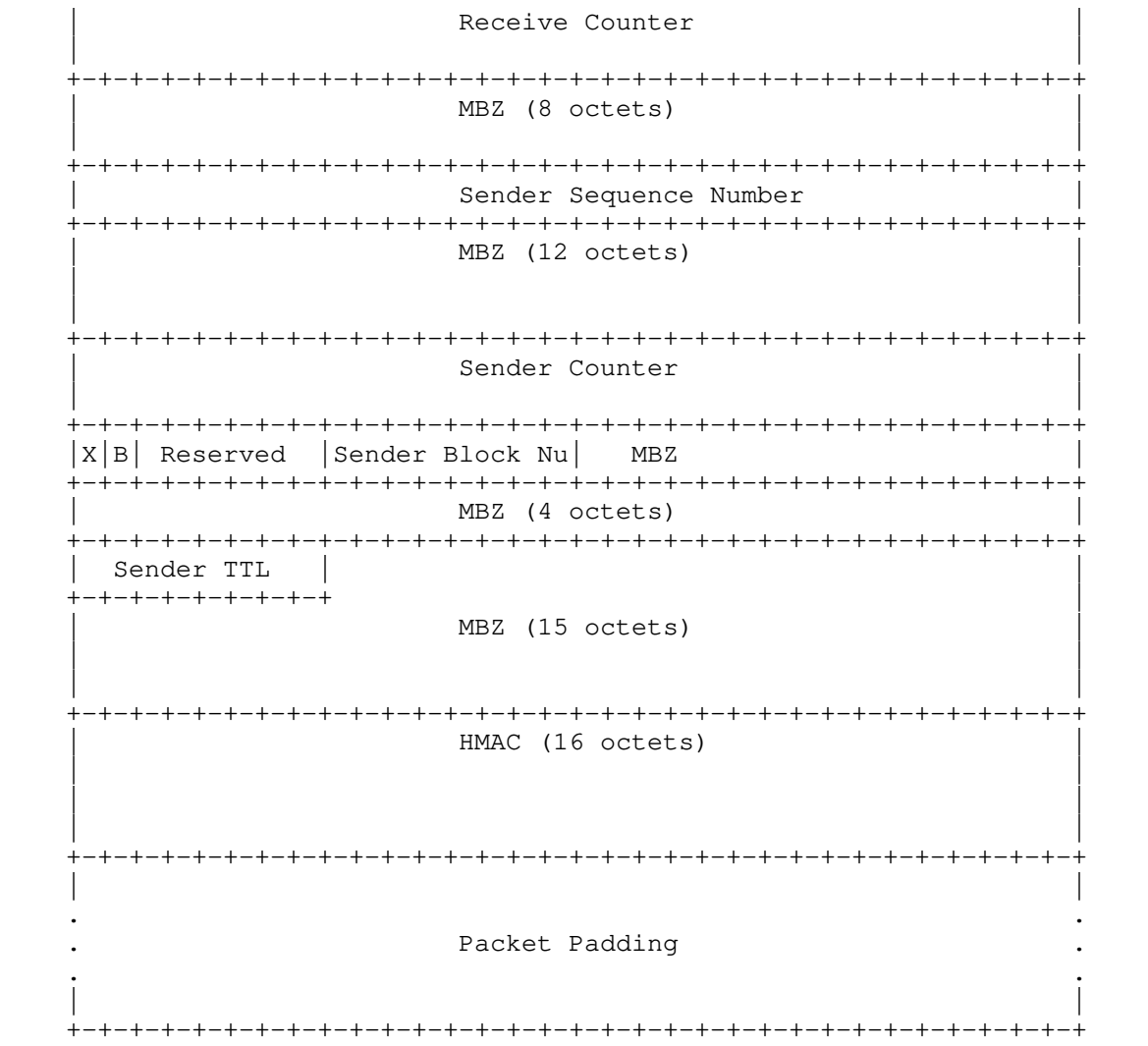


Figure 13: TWAMP Light LM Probe Response Message - Authenticated Mode

4.3. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to TWAMP Light messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.3.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC5357]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC5357].

When using the Destination IPv4 Address from the 127/8 range, the TTL field in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.3.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.3.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for PM delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR Policies [I-D.ietf-spring-segment-routing-policy] are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies as following:

- The sender root node sends probe query messages using the Replication Segment defined in [I-D.voyer-spring-sr-replication-segment] for the P2MP SR Policy as shown in Figure 14.

- Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 9. This

allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the end-to-end delay and loss performance for each P2MP leaf node of the P2MP SR Policy.

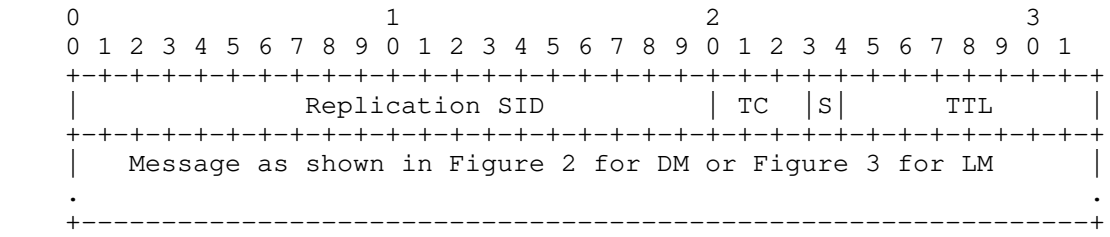


Figure 14: Example Query with Replication Segment for SR-MPLS Policy

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The PM probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the PM probe messages, sweeping of Destination Address in 127/8 range can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

The procedure defined in this document for delay measurement using the TWAMP Light probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that detection interval and scale for number of sessions need to account for the processing of the probe messages which are

punted out of fast path in forwarding (to slow path or control plane), and re-injected back on the reflector node.

8. Security Considerations

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

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, PM probe messages for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

This document does not require any IANA action.

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>.

10.2. Informative References

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

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

[RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <https://www.rfc-editor.org/info/rfc6038>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[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>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[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>.

[RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019, <https://www.rfc-editor.org/info/rfc8545>.

[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.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.voyer-spring-sr-replication-segment]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-voyer-spring-sr-replication-segment-03 (work in progress), June 2020.

[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.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-15 (work in progress), March 2020.

[BBF.TR-390]
"Performance Measurement from IP Edge to Customer Equipment using TWAMP Light", BBF TR-390, May 2017.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-02 (work in progress), March 2020.

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.

[I-D.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

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document. The authors would like to acknowledge the earlier work on the loss measurement using TWAMP described in draft-xiao-ippm-twamp-ext-direct-loss.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: February 7, 2021

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

August 6, 2020

Performance Measurement Using TWAMP Light for Segment Routing Networks
draft-gandhi-spring-twamp-srpm-10

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 describes procedure for sending and processing probe query and response messages for Performance Measurement (PM) in Segment Routing networks. The procedure uses the messages defined in RFC 5357 (Two-Way Active Measurement Protocol (TWAMP) Light) for Delay Measurement, and uses the messages defined in this document for Loss Measurement. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies.

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. 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 SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

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. These protocols rely on control-channel signaling to establish a test-channel over an UDP path. The TWAMP Light [Appendix I in RFC5357] [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer IP networks by provisioning UDP paths and eliminates the need for control-channel signaling. As described in Appendix A of [RFC8545], TWAMP Light mechanism is informative only. These protocols lack support for direct-mode Loss Measurement (LM) to detect actual Customer data traffic loss which is required in SR networks.

This document describes procedures for sending and processing probe query and response messages for Performance Measurement in SR networks. The procedure uses the messages defined in [RFC5357] (TWAMP Light) for Delay Measurement (DM), and uses the messages defined in this document for Loss Measurement. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths. This document also defines mechanisms for handling ECMPs of SR Paths for performance delay measurement. Unless otherwise described, the messages defined in [RFC5357] are not modified by this document.

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

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

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.

TC: Traffic Class.

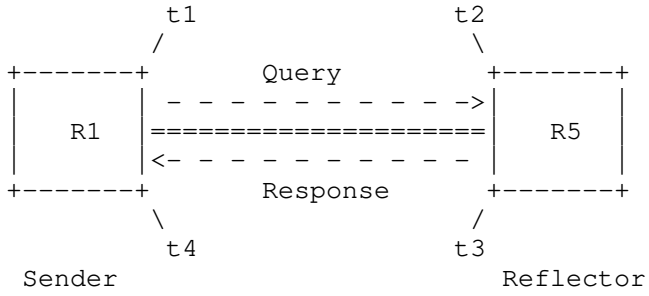
TWAMP: Two-Way Active Measurement Protocol.

2.3. Reference Topology

In the reference topology shown below, the sender node R1 initiates a performance measurement probe query message and the reflector node R5

sends a probe response message for the query message received. The probe response message is typically sent to the sender node R1.

SR is enabled on nodes R1 and R5. The nodes R1 and R5 may be directly connected via a Link or there exists a Point-to-Point (P2P) SR Path e.g. SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R5 (called tail-end).



Reference Topology

3. Overview

For one-way and two-way delay measurements in Segment Routing networks, the probe messages defined in [RFC5357] are used. For direct-mode and inferred-mode loss measurements, the messages defined in this document are used. For both Links and end-to-end SR Paths including SR Policies and Flex-Algo IGP Paths, no PM state for delay or loss measurement need to be created on the reflector node R5.

Separate UDP destination port numbers are user-configured for delay and loss measurements. As specified in [RFC8545], the reflector supports the destination UDP port 862 for delay measurement probe messages by default. This UDP port however, is not used for loss measurement probe messages defined in this document. The sender uses the UDP port number following the guidelines specified in Section 6 in [RFC6335]. The same destination UDP port is used for Links and SR Paths and the reflector is unaware if the query is for the Links or SR Paths. The number of UDP ports with PM functionality needs to be minimized due to limited hardware resoucres.

For Performance Measurement, probe query and response messages are sent as following:

- For delay measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are

used to measure the delay experienced by the actual data traffic flowing on the Links and SR Policies.

- For loss measurement, the probe messages are sent on the congruent path of the data traffic by the sender node, and are used to collect the receive traffic counters for the incoming link or incoming SID where the probe query messages are received at the reflector node (incoming link or incoming SID needed since the reflector node does not have PM state present).

The In-Situ Operations, Administration, and Maintenance (IOAM) mechanisms for SR-MPLS defined in [I-D.gandhi-mpls-ioam-sr] and for SRv6 defined in [I-D.ali-spring-ioam-srv6] are used to carry PM information such as timestamp in-band as part of the data packets, and are outside the scope of this document.

3.1. Example Provisioning Model

An example of a provisioning model and typical measurement parameters for each user-configured destination UDP port for performance delay and loss measurements is shown in the following Figure 1:

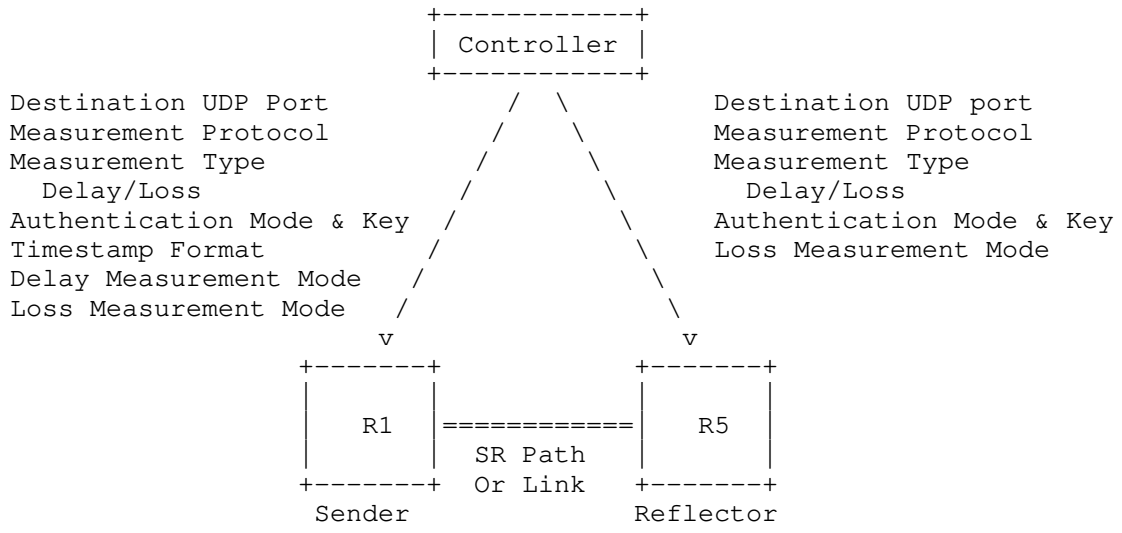


Figure 1: Example Provisioning Model

Example of Measurement Protocol is TWAMP Light, example of the Timestamp Format is PTPv2 [IEEE1588] or NTP and example of the Loss Measurement mode is inferred-mode or direct-mode.

The mechanisms to provision the sender and reflector nodes are outside the scope of this document. The provisioning model is not used for signaling the PM parameters between the reflector and sender nodes in SR networks.

The reflector node R5 uses the parameters for the timestamp format and delay measurement mode (i.e. one-way or two-way mode) from the received probe query message.

4. Probe Messages

4.1. Probe Query Message

The probe messages defined in [RFC5357] are used for delay measurement for Links and end-to-end SR Paths including SR Policies. For loss measurement, the probe messages defined in this document are used.

The sender IPv4 or IPv6 address is used as the source address. The reflector IPv4 or IPv6 address is used as the destination address. In the case of SR Policy with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the address in the range of 127/8 for IPv4 or ::FFFF:127/104 for IPv6 is used as the destination address, respectively.

4.1.1. Delay Measurement Query Message

The message content for delay measurement probe query message using UDP header [RFC0768] is shown in Figure 2. The DM probe query message is sent with user-configured Destination UDP port number for DM. The Destination UDP port cannot be used as Source port for DM, since the message does not have any indication to distinguish between the query and response message. The payload of the DM probe query message contains the delay measurement message defined in Section 4.1.2 of [RFC5357]. For symmetrical size query and response messages as defined in [RFC6038], the DM probe query message contains the payload format defined in Section 4.2.1 of [RFC5357].

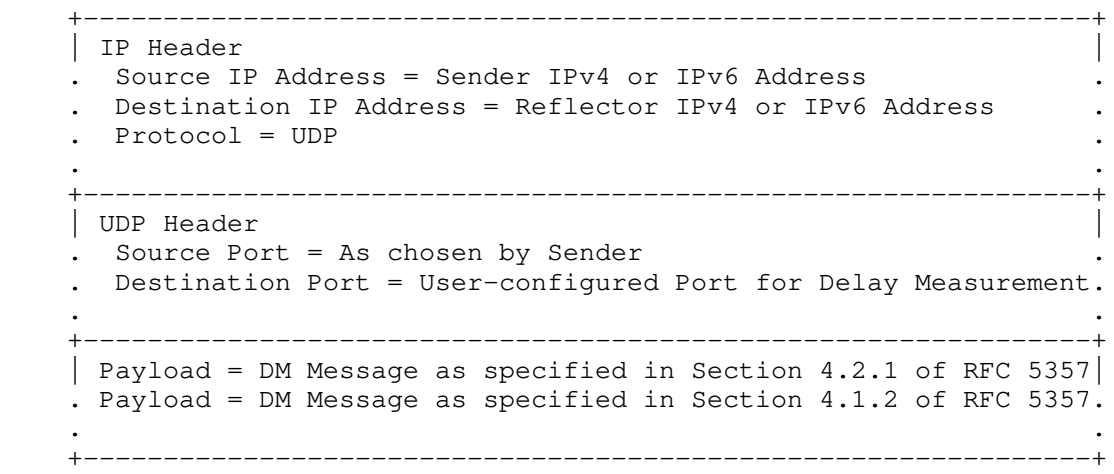


Figure 2: DM Probe Query Message

Timestamp field is eight bytes and use the format defined in Section 4.2.1 of [RFC5357]. It is recommended to use the IEEE 1588v2 Precision Time Protocol (PTP) truncated 64-bit timestamp format [IEEE1588] as specified in [RFC8186], with hardware support in Segment Routing networks.

4.1.1.1. Delay Measurement Authentication Mode

When using the authenticated mode for delay measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the delay measurement in authentication mode due to the different probe message format.

4.1.2. Loss Measurement Query Message

The message content for loss measurement probe query message using UDP header [RFC0768] is shown in Figure 3. The LM probe query message is sent with user-configured Destination UDP port number for LM, which is a different Destination UDP port number than DM. Separate Destination UDP ports are used for direct-mode and inferred-mode loss measurements. The Destination UDP port cannot be used as Source port for LM, since the message does not have any indication to distinguish between the query and response message. The LM probe query message contains the payload for loss measurement as defined in Figure 7 and Figure 8.

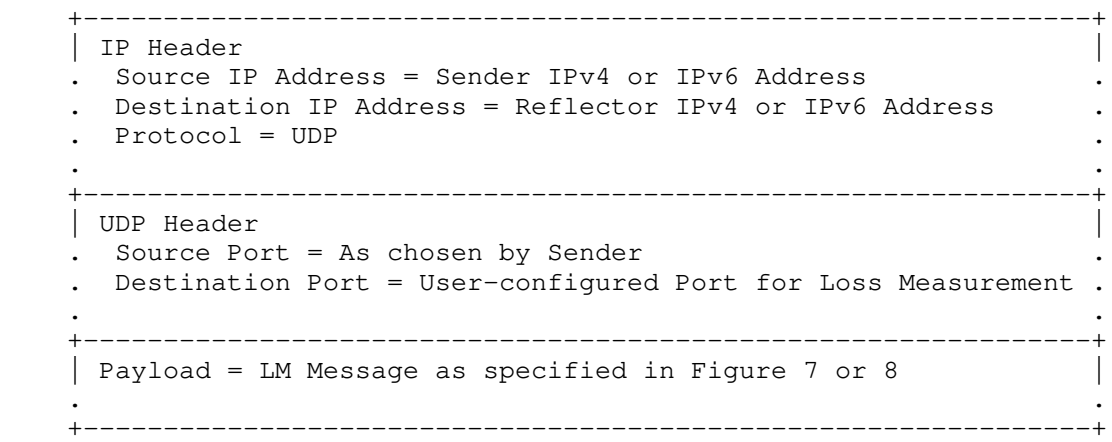


Figure 3: LM Probe Query Message

4.1.2.1. Loss Measurement Authentication Mode

When using the authenticated mode for loss measurement, the matching authentication type (e.g. HMAC-SHA-256) and key are user-configured on both the sender and reflector nodes. A separate user-configured destination UDP port is used for the loss measurement in authentication mode due to the different message format.

4.1.3. Probe Query for Links

The probe query message as defined in Figure 2 for delay measurement and Figure 3 for loss measurement are used for Links which may be physical, virtual or LAG (bundle), LAG (bundle) member, numbered/ unnumbered Links. The probe messages are pre-routed over the Link for both delay and loss measurement.

4.1.4. Probe Query for SR Policy

The performance delay and loss measurement for segment routing is applicable to both end-to-end SR-MPLS and SRv6 Policies.

4.1.4.1. Probe Query Message for SR-MPLS Policy

The probe query messages for performance measurement of an end-to-end SR-MPLS Policy is sent using its SR-MPLS header containing the MPLS segment list as shown in Figure 4.

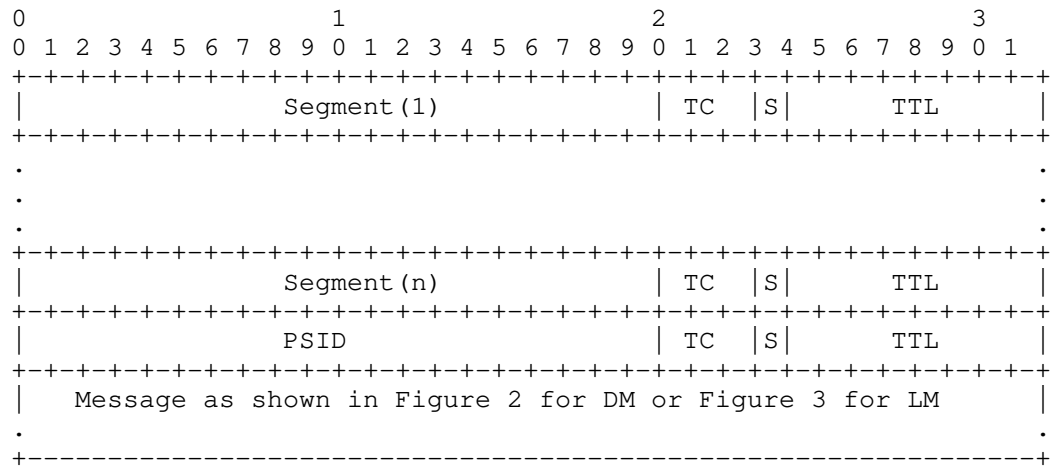


Figure 4: Example Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case for a single-hop SR Policy.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the SR-MPLS Policy is used for accounting received traffic on the egress node for loss measurement.

4.1.4.2. Probe Query Message for SRv6 Policy

An SRv6 Policy setup using the SRv6 Segment Routing Header (SRH) and a Segment List as defined in [RFC8754]. The SRv6 network programming is defined in [I-D.ietf-spring-srv6-network-programming]. The probe query messages for performance measurement of an end-to-end SRv6 Policy is sent using its SRH with Segment List as shown in Figure 5. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe query messages.

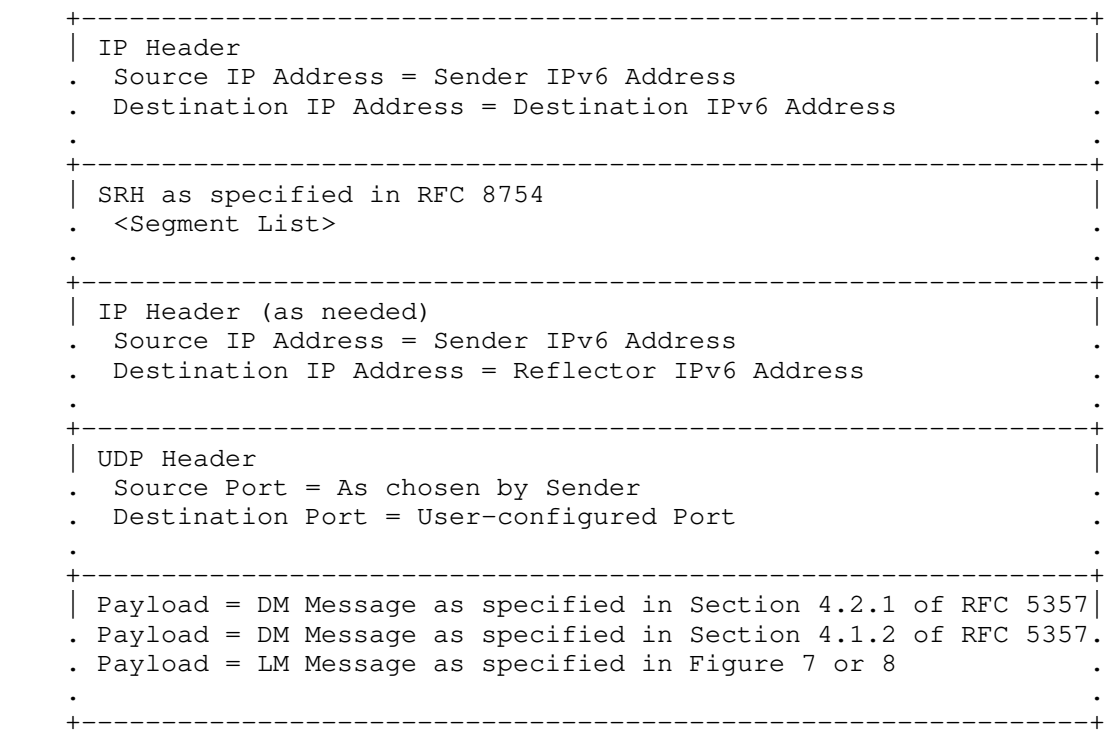


Figure 5: Example Probe Query Message for SRv6 Policy

4.1.5. Control Code Field Extension for TWAMP Light Messages

In this document, the Control Code field is defined for delay and loss measurement probe query messages for TWAMP Light in unauthenticated and authenticated modes. The modified delay measurement probe query message format is shown in Figure 6. This message format is backwards compatible with the message format defined in [RFC5357] as its reflectors ignore the received field (previously identified as MBZ). With this field, the reflector node does not require any additional SR state for PM (recall that in SR networks, the state is in the probe packet and signaling of the parameters is avoided). The usage of the Control Code is not limited to the SR paths and can be used for non-SR paths in a network.

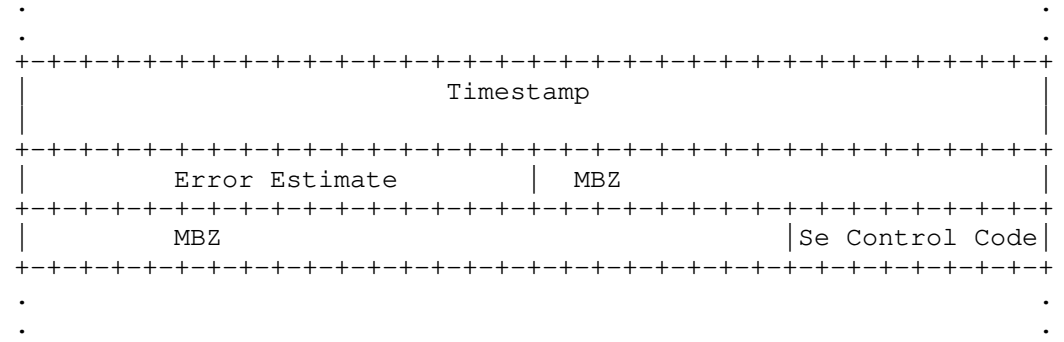


Figure 6: Sender Control Code in TWAMP Light DM Message

Sender Control Code: Set as follows in TWAMP Light probe query message.

In a Query:

0x0: Out-of-band Response Requested. Indicates that the probe response is not required over the same path in the reverse direction. This is also the default behavior.

0x1: In-band Response Requested. Indicates that this query has been sent over a bidirectional path and the probe response is required over the same path in the reverse direction.

0x2: No Response Requested.

4.1.6. Loss Measurement Query Message Extensions

In this document, TWAMP Light probe query messages for loss measurement are defined as shown in Figure 7 and Figure 8. The message formats are hardware efficient due to well-known locations of the counters and payload small in size. They are stand-alone and similar to the delay measurement message formats (e.g. location of the Counter and Timestamp). They also do not require backwards compatibility and support for the existing DM message formats from [RFC5357] as different user-configured destination UDP port is used for loss measurement.

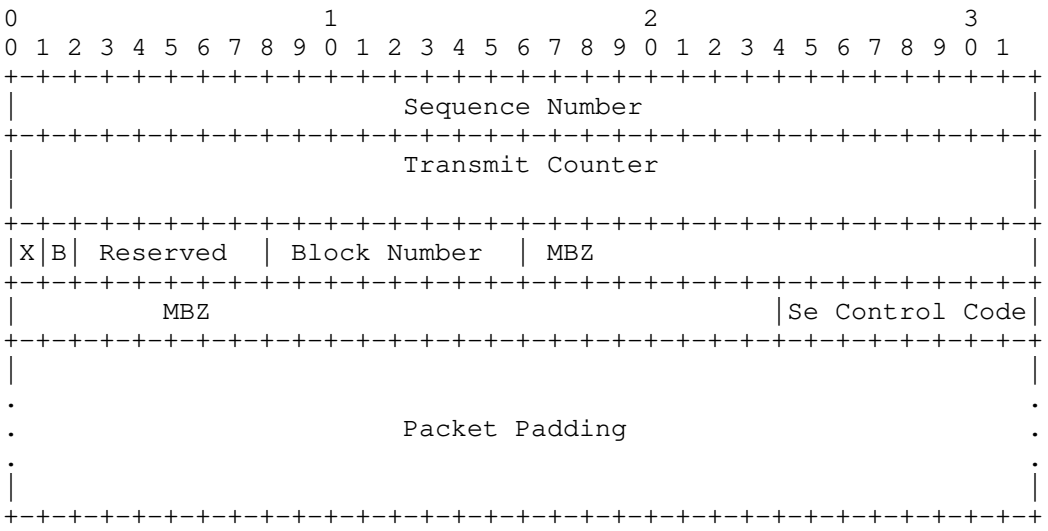


Figure 7: TWAMP Light LM Probe Query Message - Unauthenticated Mode

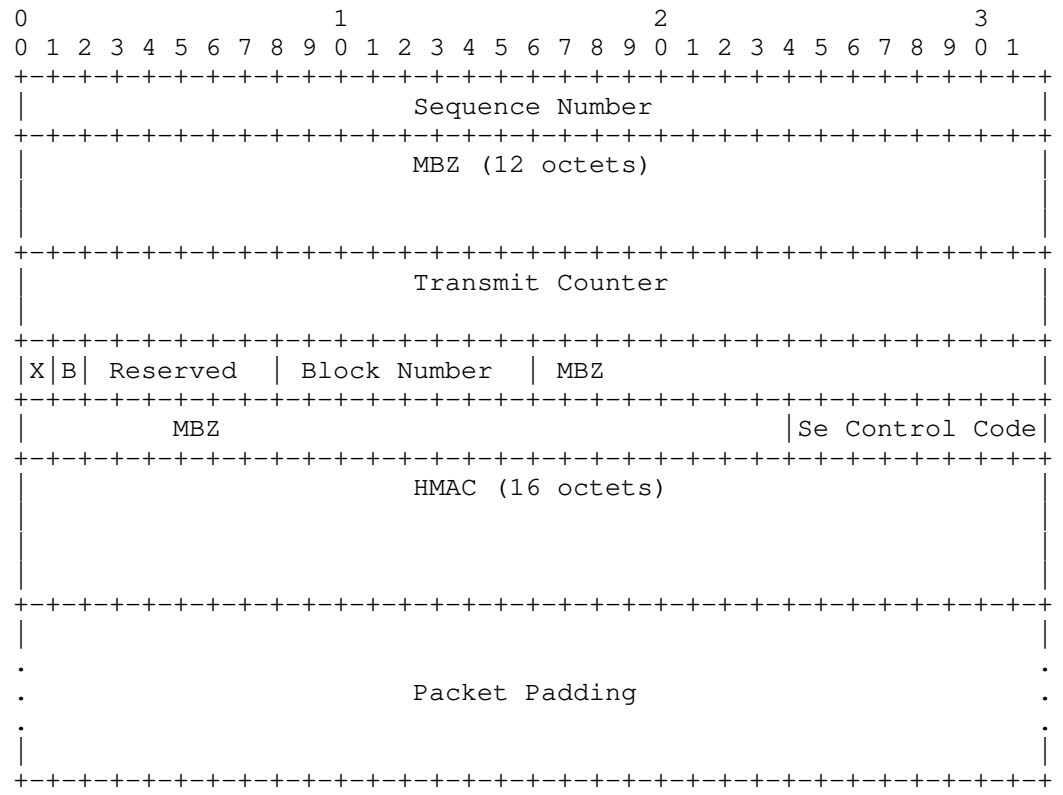


Figure 8: TWAMP Light LM Probe Query Message - Authenticated Mode

Sequence Number (32-bit): As defined in [RFC5357].

Transmit Counter (64-bit): The number of packets or octets sent by the sender node in the query message and by the reflector node in the response message. The counter is always written at the well-known location in the probe query and response messages.

Receive Counter (64-bit): The number of packets or octets received at the reflector node. It is written by the reflector node in the probe response message.

Sender Counter (64-bit): This is the exact copy of the transmit counter from the received query message. It is written by the reflector node in the probe response message.

Sender Sequence Number (32-bit): As defined in [RFC5357].

Sender TTL: As defined in Section 7.1.

LM Flags: The meanings of the Flag bits are:

X: Extended counter format indicator. Indicates the use of extended (64-bit) counter values. Initialized to 1 upon creation (and prior to transmission) of an LM query and copied from an LM Query to an LM response message. Set to 0 when the LM message is transmitted or received over an interface that writes 32-bit counter values.

B: Octet (byte) count. When set to 1, indicates that the Counter 1-4 fields represent octet counts. The octet count applies to all packets within the LM scope, and the octet count of a packet sent or received includes the total length of that packet (but excludes headers, labels, or framing of the channel itself). When set to 0, indicates that the Counter fields represent packet counts.

Block Number (8-bit): The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to color the data traffic. To be able to correlate the transmit and receive traffic counters of the matching color, the Block Number (or color) of the traffic counters is carried by the probe query and response messages for loss measurement. The Block Number can also be used to aggregate performance metrics collected.

HMAC: The probe message in authenticated mode includes a key Hashed Message Authentication Code (HMAC) [RFC2104] hash. Each probe query and response messages are authenticated by adding Sequence Number with Hashed Message Authentication Code (HMAC) TLV. It can use HMAC-SHA-256 truncated to 128 bits (similarly to the use of it in IPSec defined in [RFC4868]); hence the length of the HMAC field is 16 octets.

HMAC uses its own key and the mechanism to distribute the HMAC key is outside the scope of this document.

In authenticated mode, only the sequence number is encrypted, and the other payload fields are sent in clear text. The probe message may include Comp.MBZ (Must Be Zero) variable length field to align the packet on 16 octets boundary.

4.2. Probe Response Message

The probe response message is sent using the IP/UDP information from the received probe query message. The content of the probe response message is shown in Figure 9.

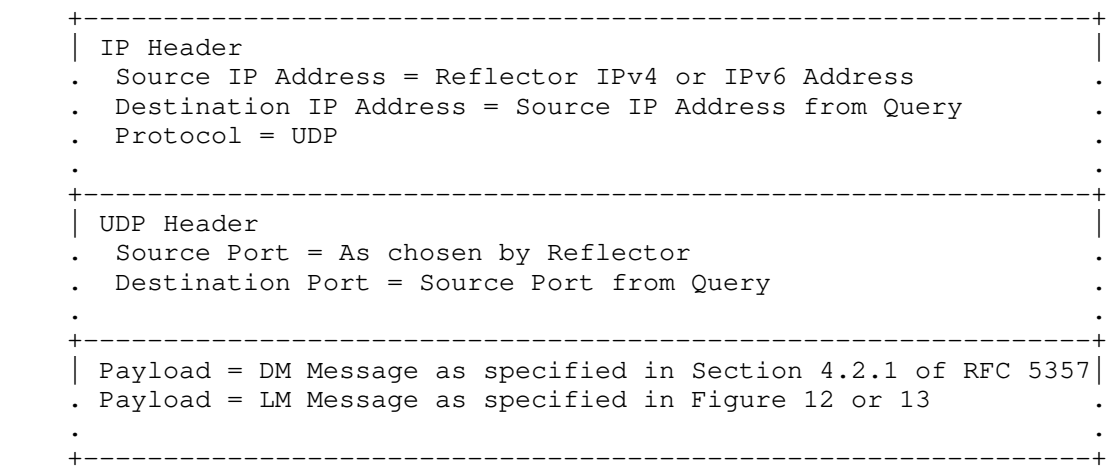


Figure 9: Probe Response Message

4.2.1. One-way Measurement Mode

In one-way measurement mode, the probe response message as defined in Figure 9 is sent back out-of-band to the sender node, for both Links and SR Policies. The Sender Control Code is set to "Out-of-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. However, only timestamps t1 and t2 are used to measure one-way delay as (t2 - t1).

4.2.2. Two-way Measurement Mode

In two-way measurement mode, when using a bidirectional path, the probe response message as defined in Figure 9 is sent back to the sender node on the congruent path of the data traffic on the same reverse direction Link or associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. The Sender Control Code is set to "In-band Response Requested". In this delay measurement mode, as per Reference Topology, all timestamps t1, t2, t3, and t4 are collected by the probes. All four timestamps are used to measure two-way delay as ((t4 - t1) - (t3 - t2)).

Specifically, the probe response message is sent back on the incoming physical interface where the probe query message is received. This is required for example, in case of two-way measurement mode for Link delay.

4.2.2.1. Probe Response Message for SR-MPLS Policy

The message content for sending probe response message for two-way performance measurement of an end-to-end SR-MPLS Policy is shown in Figure 10.

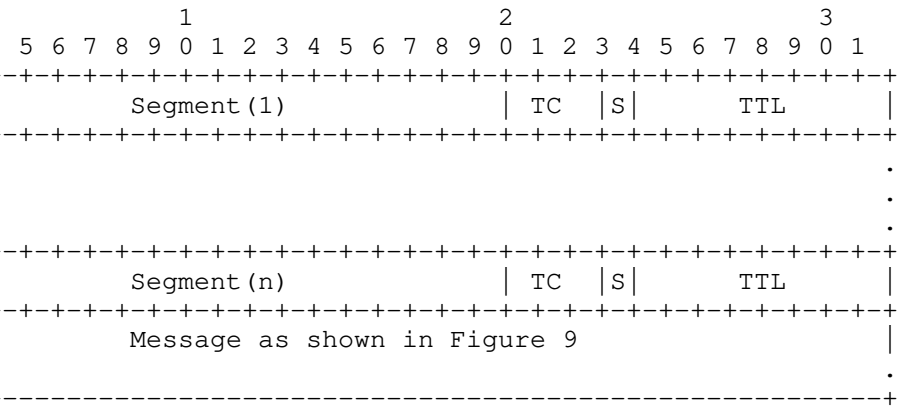


Figure 10: Example Probe Response Message for SR-MPLS Policy

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of the forward SR Policy in the probe query can be used to find the associated reverse SR Policy [I-D.ietf-pce-sr-bidir-path] to send the probe response message for two-way measurement of SR Policy.

4.2.2.2. Probe Response Message for SRv6 Policy

The message content for sending probe response message on the congruent path of the data traffic for two-way performance measurement of an end-to-end SRv6 Policy with SRH is shown in Figure 11. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received probe response messages.

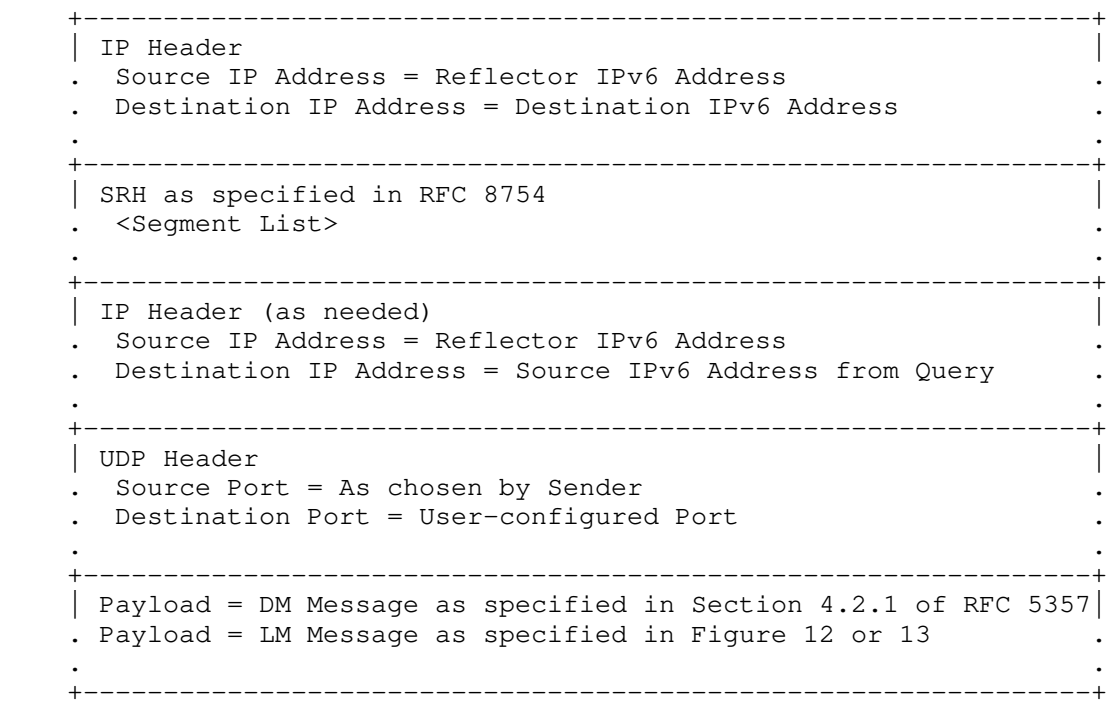


Figure 11: Example Probe Response Message for SRv6 Policy

4.2.3. Loss Measurement Response Message Extensions

In this document, TWAMP Light probe response message formats are defined for loss measurement as shown in Figure 12 and Figure 13.

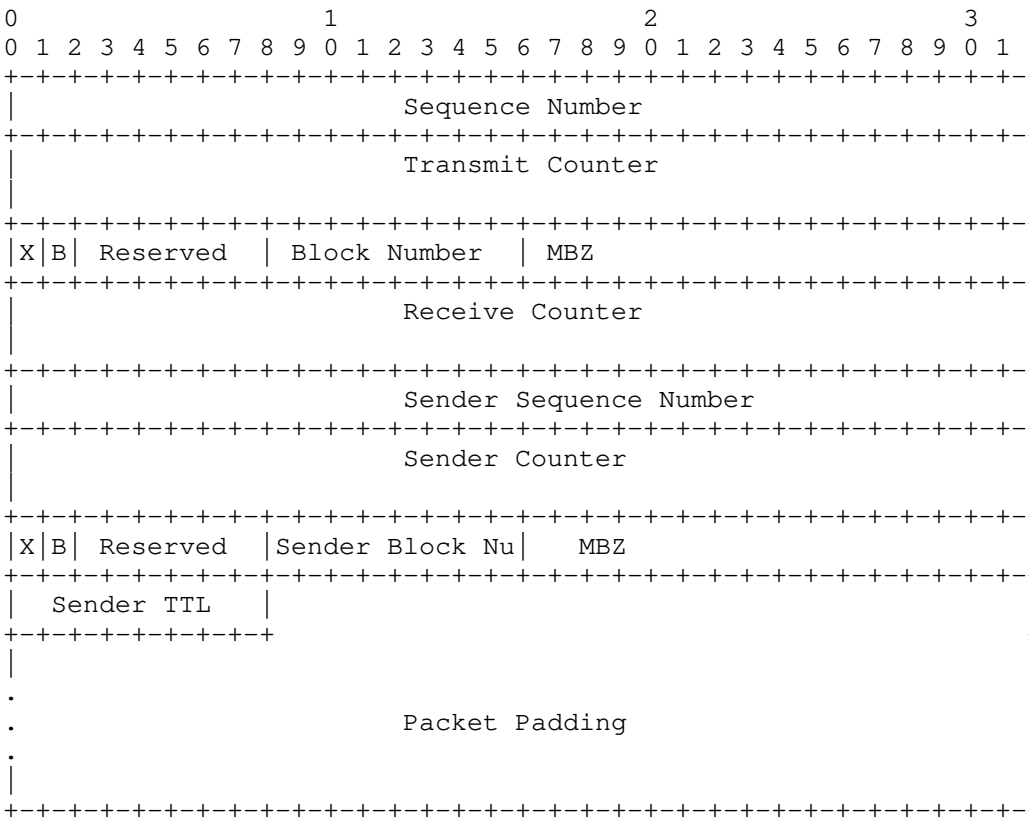
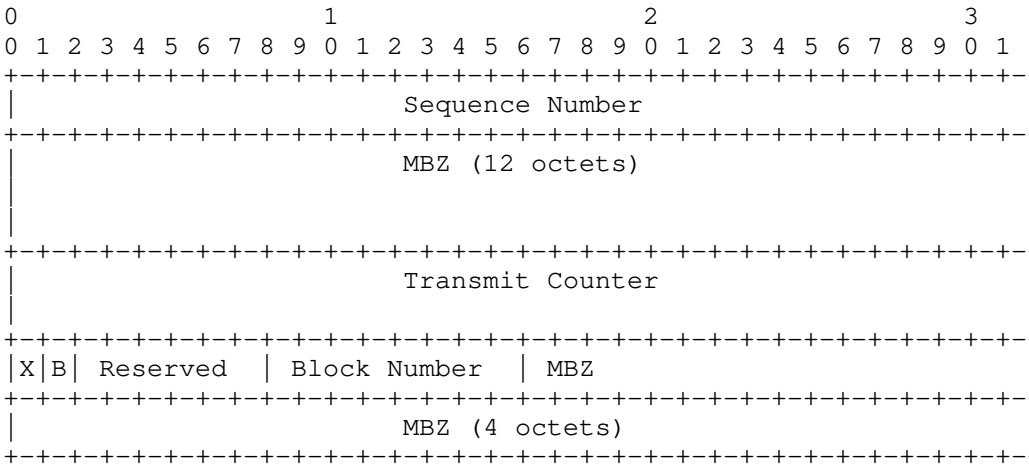


Figure 12: TWAMP Light LM Probe Response Message - Unauthenticated Mode



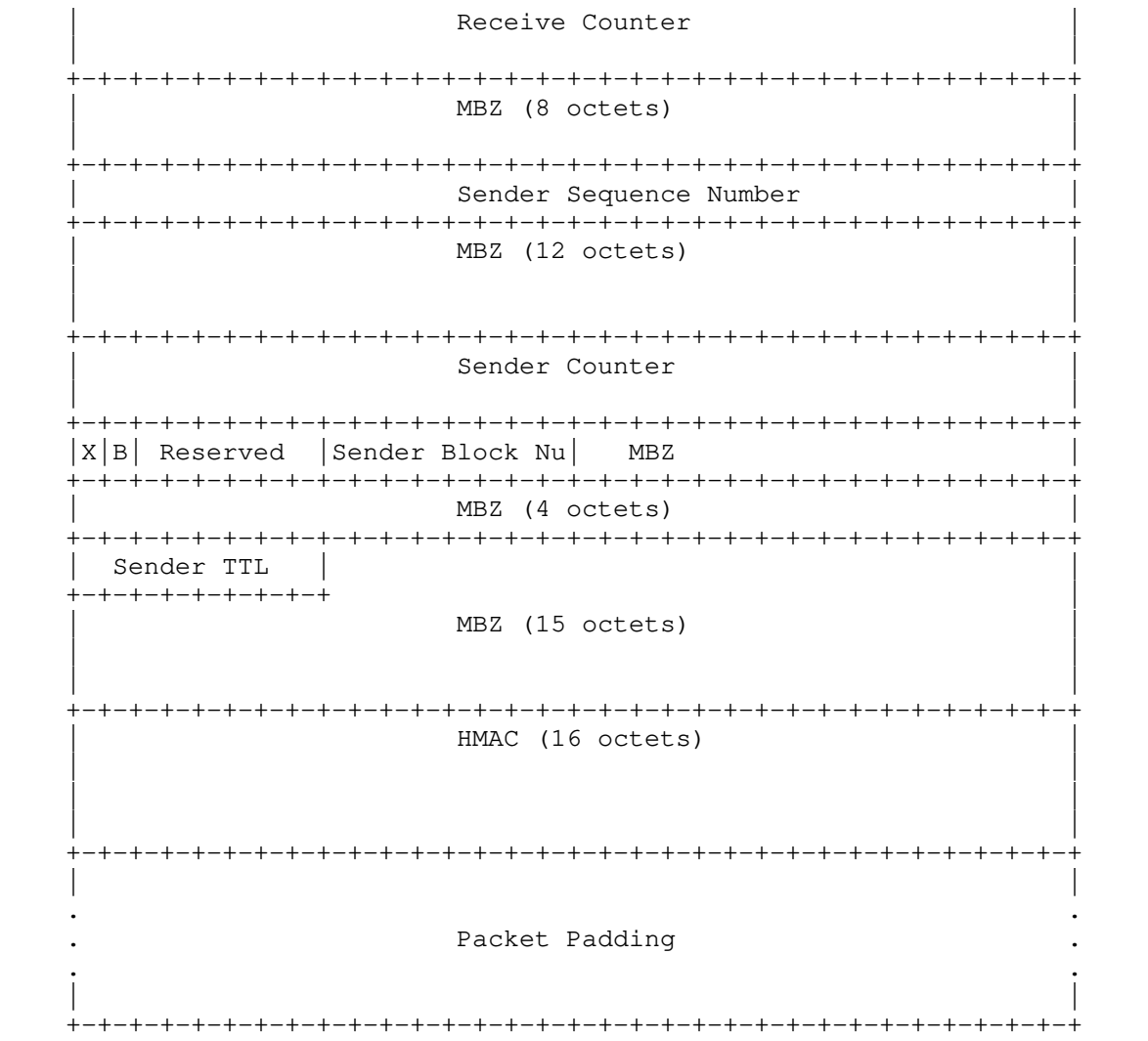


Figure 13: TWAMP Light LM Probe Response Message - Authenticated Mode

4.3. Additional Probe Message Processing Rules

The processing rules defined in this section are applicable to TWAMP Light messages for delay and loss measurement for Links and end-to-end SR Paths including SR Policies.

4.3.1. TTL and Hop Limit

The TTL field in the IPv4 and MPLS headers of the probe query messages is set to 255 [RFC5357]. Similarly, the Hop Limit field in the IPv6 and SRH headers of the probe query messages is set to 255 [RFC5357].

When using the Destination IPv4 Address from the 127/8 range, the TTL field in the IPv4 header is set to 1 [RFC8029]. Similarly, when using the Destination IPv6 Address from the ::FFFF:127/104 range, the Hop Limit field in the IPv6 header is set to 1.

For Link performance delay and loss measurements, the TTL or Hop Limit field in the probe message is set to 1 in both one-way and two-way measurement modes.

4.3.2. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the probe messages.

4.3.3. UDP Checksum

The UDP Checksum Complement for delay and loss measurement messages follows the procedure defined in [RFC7820] and can be optionally used with the procedures defined in this document.

For IPv4 and IPv6 probe messages, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the sender node sets the UDP checksum to 0 [RFC6936] [RFC8085]. The receiving node bypasses the checksum validation and accepts the packets with UDP checksum value 0 for the UDP port being used for delay and loss measurements.

5. Performance Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR Path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy] or P2MP Transport [I-D.shen-spring-p2mp-transport-chain]).

The procedures for delay and loss measurement described in this document for P2P SR Policies are also equally applicable to the P2MP SR Policies. The procedure for one-way measurement is defined as following:

- The sender root node sends probe query messages using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 14.

- The probe query messages can contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].

- Each reflector leaf node sends its IP address in the Source Address of the probe response messages as shown in Figure 14. This allows the sender root node to identify the reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the delay and loss performance for each P2MP leaf node of the end-to-end P2MP SR Policy.

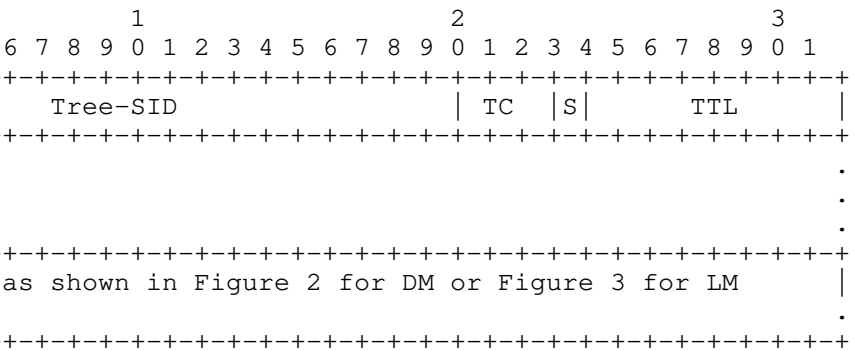


Figure 14: Example Probe Query with Tree-SID for SR-MPLS Policy

The probe query messages can also be sent using the scheme defined for P2MP Transport using Chain Replication that may contain Bud SID as defined in [I-D.shen-spring-p2mp-transport-chain].

The considerations for two-way mode for performance measurement for P2MP SR Policy (e.g. for bidirectional SR Path) are outside the scope of this document.

6. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The probe messages need to be sent to traverse different ECMP paths to measure performance delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the performance measurement. In IPv4 header of the probe messages, sweeping of Destination Address in 127/8 range can be used to exercise particular ECMP paths. As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping.

The considerations for performance loss measurement for different ECMP paths of an SR Policy are outside the scope of this document.

7. Performance Delay and Liveness Monitoring

Liveness monitoring is required for connectivity verification and continuity check in an SR network. The procedure defined in this document for delay measurement using the TWAMP Light probe messages can also be applied to liveness monitoring of Links and SR Paths. The one-way or two-way measurement mode can be used for liveness monitoring. Liveness failure is notified when consecutive N number of probe response messages are not received back at the sender node, where N is locally provisioned value. Note that failure detection interval and scale for number of probes need to account for the processing of the probe query messages which need to be punted from the forwarding fast path (to slow path or control plane) and response messages need to be injected on the reflector node. This is enhanced by using the probes in loopback mode as described in [I-D.gandhi-spring-sr-enhanced-plm].

8. Security Considerations

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

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the sender, of the counter or timestamp fields in received measurement response messages. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid message to a single query/response cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the probe messages. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, probe messages for SRv6 may not need authentication mode. Cryptographic measures may be

enhanced by the correct configuration of access-control lists and firewalls.

9. IANA Considerations

This document does not require any IANA action.

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>.

10.2. Informative References

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

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <https://www.rfc-editor.org/info/rfc4868>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

[RFC6038] Morton, A. and L. Ciavattone, "Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features", RFC 6038, DOI 10.17487/RFC6038, October 2010, <https://www.rfc-editor.org/info/rfc6038>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[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>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[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>.

[RFC8545] Morton, A., Ed. and G. Mirsky, Ed., "Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)", RFC 8545, DOI 10.17487/RFC8545, March 2019, <https://www.rfc-editor.org/info/rfc8545>.

[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.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[I-D.ietf-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-00 (work in progress), July 2020.

[I-D.shen-spring-p2mp-transport-chain] Shen, Y., Zhang, Z., Parekh, R., Bidgoli, H., and Y. Kamite, "Point-to-Multipoint Transport Using Chain Replication in Segment Routing", draft-shen-spring-p2mp-transport-chain-02 (work in progress), April 2020.

[I-D.ietf-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.

[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.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.

[BBF.TR-390]
"Performance Measurement from IP Edge to Customer Equipment using TWAMP Light", BBF TR-390, May 2017.

[I-D.gandhi-mpls-ioam-sr]
Gandhi, R., Ali, Z., Filsfils, C., Brockners, F., Wen, B., and V. Kozak, "MPLS Data Plane Encapsulation for In-situ OAM Data", draft-gandhi-mpls-ioam-sr-02 (work in progress), March 2020.

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Kumar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-02 (work in progress), November 2019.

[I-D.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.

[I-D.gandhi-spring-sr-enhanced-plm]
Gandhi, R., Filsfils, C., Vaghamshi, N., Nagarajah, M., and R. Foote, "Enhanced Performance Delay and Liveness Monitoring in Segment Routing Networks", draft-gandhi-spring-sr-enhanced-plm-02 (work in progress), July 2020.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in Segment Routing. The authors would also like to thank Greg Mirsky for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu, both from Cisco Systems have helped significantly improve the mechanisms defined in this document. The authors would like to acknowledge the earlier work on the loss measurement using TWAMP described in draft-xiao-ippm-twamp-ext-direct-loss.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2021

S. Hegde
C. Bowers
Juniper Networks Inc.
X. Xu
Alibaba Inc.
A. Gulko
Refinitiv

July 12, 2020

Seamless Segment Routing
draft-hegde-spring-mpls-seamless-sr-00

Abstract

In order to operate networks with large numbers of devices, network operators organize networks into multiple smaller network domains. Each network domain typically runs an IGP which has complete visibility within its own domain, but limited visibility outside of its domain. Seamless Segment Routing (Seamless SR) provides flexible, scalable and reliable end-to-end connectivity for services across independent network domains. Seamless SR accomodates domains using SR, LDP, and RSVP for MPLS label distribution as well as domains running IP without MPLS (IP-Fabric).

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 13, 2021.

Table of Contents

1. Introduction

The Seamless SR architecture builds upon the Seamless MPLS architecture, which has been widely deployed to provide end-to-end transport for service in 3G/4G networks. [I-D.ietf-mpls-seamless-mpls], contains a good description of the Seamless MPLS architecture. Although [I-D.ietf-mpls-seamless-mpls] has not been published as an RFC, it serves as a useful description of the Seamless MPLS architecture. [I-D.ietf-mpls-seamless-mpls] describes the Seamless MPLS architecture, which uses LDP and/or RSVP for intra-domain label distribution, and BGP-LU [RFC3107] for end-to-end label distribution. The Seamless SR architecture builds on the the Seamless MPLS architecture. Seamless SR focuses on using segment routing for intra-domain label distribution.

By using segment routing for intra-domain label distribution, Seamless SR is able to easily support both SR-MPLS on IPv4 and IPv6 networks. This overcomes a limitation of the classic Seamless MPLS architecture, which was limited to run MPLS on IPv4 networks in practice. Seamless SR (like Seamless MPLS) can use BGP-LU (RFC 3107) to stitch different domains. However, Seamless SR can also take advantage of BGP Prefix-SID [RFC8669] to provide predictable and deterministic labels for the inter-domain connectivity.

5G technology is expected to place new requirements on the packet transport networks that support it. To enable 5G technology, packet transport networks will need to be capable of handling much greater bandwidth than today's 3G/4G networks. 5G networks are expected to require up to 250Gbps in the fronthaul and up to 400Gbps in the backhaul. The number of transport network devices is also expected to grow significantly to cater to 5G needs. Overall service availabilty requirements for 5G will place significant requirements on the resiliency of packet transport networks.

There is a desire to allow many 5G network functions to be virtualized and cloud native. In order to support latency-sensitive cloud-native 5G network functions, packet transport networks should be capable of providing low-latency paths end-to-end. Some services will require low-latency paths while others may require different QoS properties. The network should be able to differentiate the services and provide corresponding SLA transport paths.

The basic functionality of the Seamless SR architecture does not require any enhancements to existing protocols. However, in order to support end-to-end service requirements across multiple domains, protocol extensions may be needed. This draft discusses usecases, requirements, and potential protocol enhancements.

2. Terminology

This document uses the following terminology

- Access Node (AN): An access node is a node which processes customers frames or packets at Layer 2 or above. This includes but is not limited to DSLAMs and Cell Site Routers in 5G networks. Access nodes have only limited MPLS functionalities in order to reduce complexity in the access network.

- Pre-Aggregation Node (P-AGG): A pre-aggregation node (P-AGG) is a node which aggregates several access nodes (ANs).

- Aggregation Node (AGG): A aggregation node (AGG) is a node which aggregates several pre-aggregation nodes (P-AGG).

- Area Border Router (ABR): Router between aggregation and core domain.

- Label Switch Router (LSR): Label Switch router are pure transit nodes. ideally have no customer or service state and are therefore decoupled from service creation.

- Use Case: Describes a typical network including service creation points and distribution of remote node loopback prefixes.

Figure 1: Terminology

3. Use Cases

3.1. Service provider network

Service provider transport networks use multiple domains to support scalability. For this analysis, we consider a representative network design with four level of hierarchy: access domains, pre-aggregation domains, aggregation domains and a core. (See Figure 2). The 5G transport networks in particular are expected to scale to very large number of access nodes due to the shorter range of the 5G radio technology. The networks are expected to scale up to one million nodes.

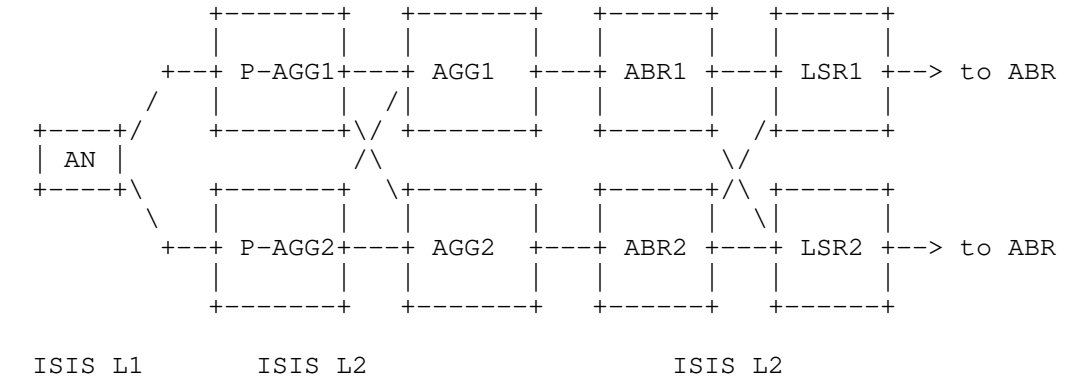


Figure 2: 5G network

Many network functions in a 5G network will be virtualized and distributed across multiple data centers. Virtualized network functions are instantiated dynamically across different compute resources. This requires that the underlying transport network supports the stringent SLA on end-to-end paths.

5G networks support variety of service use cases that require end-to-end slicing. In certain cases the end-to-end connectivity requires differentiated forwarding capabilities. Seamless SR architecture should provide ability to establish end-to-end paths that satisfy the required SLAs. For Example, End user requirement could be to establish low latency path end-to-end. The System Architecture for the 5G System [TS.23.501-3GPP] currently defines four standardized Slice/Service Types: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communication (URLLC), massive Internet of Things (mIoT), Vehicle to everything (V2X). The Seamless SR should support end-to-

end QoS mechansisms to allow the creation of network slices with these four Slice/Service Types.

Many deployments consist of ring topologies in the access and aggregation networks. In the ring topologies, there are atmost two forwarding paths for the traffic, where as the core networks consist of nodes with more denser connectivity compared to ring topologies. Thus core networks may have larger number of TE paths while access networks will have smaller number of TE paths. The Seamless SR architecture should support ability to have more TE paths in one domain and lesser number of TE paths in another domain and provide ability to effectively connect the domain end-to-end satisfying end-to-end constraints.

3.2. Large scale WAN networks

As WAN networks grow beyond several thousand nodes, it is often useful to divide the network into multiple IGP domains. The different IGP domains provide better fault isolation. Smaller IGP domains can also reduce FIB scale.

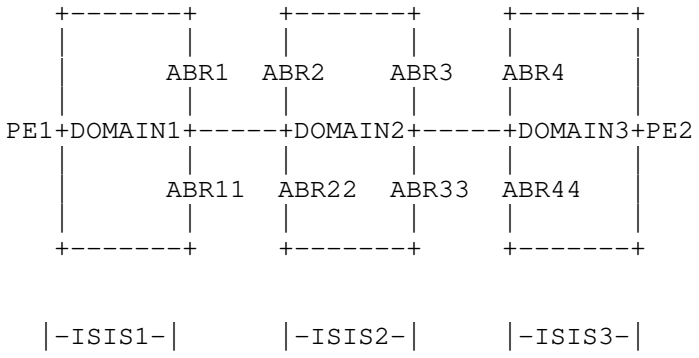


Figure 3: WAN Network

Large WAN networks often cross national boundaries. In order to meet data sovereignity requirements, operators need to maintain strict control over end-to-end traffic-engineered(TE) paths. Segment Routing provides two main solutions to implement highly constrained TE paths. Flex-algo (defined in [I-D.ietf-lsr-flex-algo]) uses prefix-SIDs computed by all nodes in the IGP domain using the same pruned topology. Highly constrained TE paths for the data soveriegnty use case can also be implemented using SR-TE policies ([I-D.ietf-spring-segment-routing-policy]) built using unprotected adjacency SIDs.

Both of these approaches work well for intra-domain TE paths. However, they both have limitations when one tries to extend them to the creation of highly constrained inter-domain TE paths. A goal of seamless SR is to be able to create highly constrained inter-domain TE paths in a scalable manner.

3.3. Data Center Interconnect (DCI) Networks

Data centers are playing an increasingly important role in providing access to information and applications. Geographically diverse data centers usually connect via a high speed, reliable and secure core network.

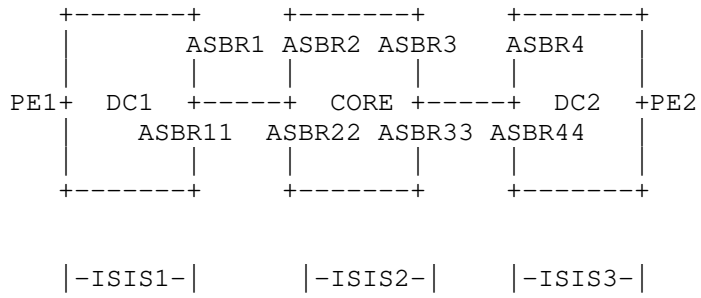


Figure 4: DCI Network

In many Data Center deployments, applications require end-to-end path diversity and/or end-to-end low latency paths. It is desirable to have a uniform technology deployed in the core as well as in the Data Centers to create these SLA paths. Such uniformity simplifies the network to a great extent. It is desirable for a solution to only require service-related configurations on the access end-points where services are attached, avoiding service-related configurations on the ABR/ASBR nodes.

3.4. Multicast Usecases

Multicast services such as IPTV and multicast also need to be support across a multi-domain service provider network. Multicast services such as IPTV, multicast VPN etc need to be supported in a service provider network.

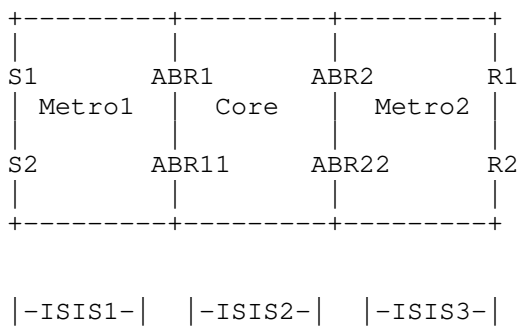


Figure 5: Multicast usecases

Figure 5 shows a simplified multi-domain network supporting multicast. Multicast sources S1 and S2 lie in a different domain from the receivers R1 and R2. Using multiple IGP domains presents a problem for the establishment of multicast replication trees. Typically, a multicast receiver does a reverse path forwarding (RPF) lookup for a multicast source. One solution is to leak the routes for multicast sources across the IGP domains. However, this can compromise the scaling properties of the multi-domain architecture. SR-P2MP [I-D.voyer-pim-sr-p2mp-policy] offers a solution for both intra-domain and inter-domain multicast. However, it does accomodate deployments using existing intra-domain multicast technology, such as mLDP [RFC6388] in some of the domains. A solution should accomodate a mixture of existing and newer technologies to better facilitate coexistence and migration.

4. Requirements

This section provides a summary of requirements derived from the use cases described in previous sections.

4.1. MPLS Transport

The architecture should provide MPLS transport between two service endpoints regardless of whether the two end-points are in the same IGP domain, different IGP domains, or in different autonomous systems.

The MPLS transport should be supported on IPv4, IPv6, and dual-stack networks.

4.2. SLA Guarantee

The architecture should allow the creation of paths that support end-to-end SLAs. The paths should for example obey constraints related to latency, diversity, and availability.

The architecture should support end-to-end network slicing as described by 5G transport requirements [TS.23.501-3GPP].

4.3. Scalability

The architecture should be able to support up to 1 million nodes.

The architecture should facilitate the use of access nodes with low RIB/FIB and low CPU capabilities.

The architecture should facilitate the use of access nodes with low label stacking capability.

The architecture should allow for a scalable response to network events. An individual node should only need to respond to a limited subset of network events.

Service routes on the border nodes should be minimized.

4.4. Availability

Traffic should be Fast Reroute (FRR) protected against link, node, and SRLG failures within a domain.

Traffic should be Fast Reroute (FRR) protected against border node failures.

Traffic should be Fast Reroute (FRR) protected against egress node and egress link failures.

4.5. Operations

Each domain should be independent and should not depend on the transport technology in another domain. This allows for more flexible evolution of the network.

Basic MPLS OAM mechanisms described in [RFC8029] should be supported.

End-to-end mpls ping and traceroute procedures should be supported.

Ability to validate the path inside each domain should be supported.

Statistics for inter-domain paths on the ingress and egress PE nodes as well as border nodes should be supported.

4.6. Service Mapping

The architecture should support the automated steering of traffic on to transport paths based on communities carried in the service prefix advertisements.

The architecture should support the steering of traffic on to transport paths based the DSCP value carried in IPv4/IPv6 packets.

Traffic steering based on EXP bits in the mpls header should be supported.

Traffic steering based on 5-tuple packet filter should be supported. Source address, destination address, source port, destination port and protocol fields should be allowed.

All traffic steering mechanims should be supported for all kinds of service traffic including VPN traffic as well as global internet traffic.

The core domain is expected to have more traffic enginnering constraints as compared to metros. The ability to map the services to appropriate transport tunnels at service attachment points should be supported.

5. Seamless Segment Routing architecture

5.1. Solution Concepts

The solution described below makes use of the following concepts.

- Transport Class (TC): A Transport Class is defined as a collection of end-to-end MPLS paths that satisfy a set of constraints or Service Level Aggreements.

- BGP-Classful Transport (BGP-CT): A new BGP family used to establish Transport Class paths across different domains.

- Route Distinguisher (RD): The Route Distinguisher is defined in RFC4364. In BGP-CT, the RD is used in BGP advertisements to differentiate multiple paths to the same loopback address. It may be useful to automatically generate RDs in order to simplify configuration.

- Route Target (RT): The Route Target extended community is carried in BGP-CT advertisements. The RT represents the Transport Class of an advertised path.

- Mapping Community (MC): The Mapping Community is the standard BGP community as defined in RFC1997. In the Seamless SR architecture, an MC is carried by a service route. The MC is used to identify the specific local policy used to map traffic for a service route to different Transport Class paths. The local policy can include additional traffic steering properties for placing traffic on different Transport Class paths. The values of the MCs and the corresponding local policies for service mapping are defined by the network operator.

Figure 6: Solution Concepts

5.2. BGP Classful Transport

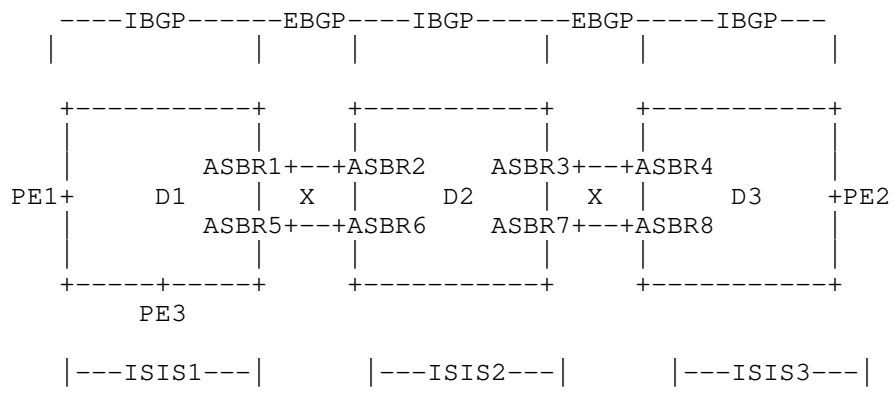


Figure 7: WAN Network

The above diagram shows a WAN network divided into 3 different domains. Within each domain, BGP sessions are established between the PE nodes and the border nodes as well as between border nodes. BGP sessions are also established between border nodes across domains. The goal is for PE1 to have MPLS connectivity to PE2, satisfying specific characteristics. Multiple MPLS paths from PE1 to PE2 are required in order to satisfy diffrent SLAs. [I-D.kaliraj-idr-bgp-classful-transport-planes] defines a new BGP family called BGP-Classful Transport. The NLRI for this new family consists of a prefix and a Route Distinguisher. The prefix corresponds to the loopback of the destination PE, and RD is used to distinguish multiple paths to the same PE loopback. The BGP-CT advertisement also carries a Route Target. The RT specifies the Transport Class to which the BGP-CT advertisement belongs.

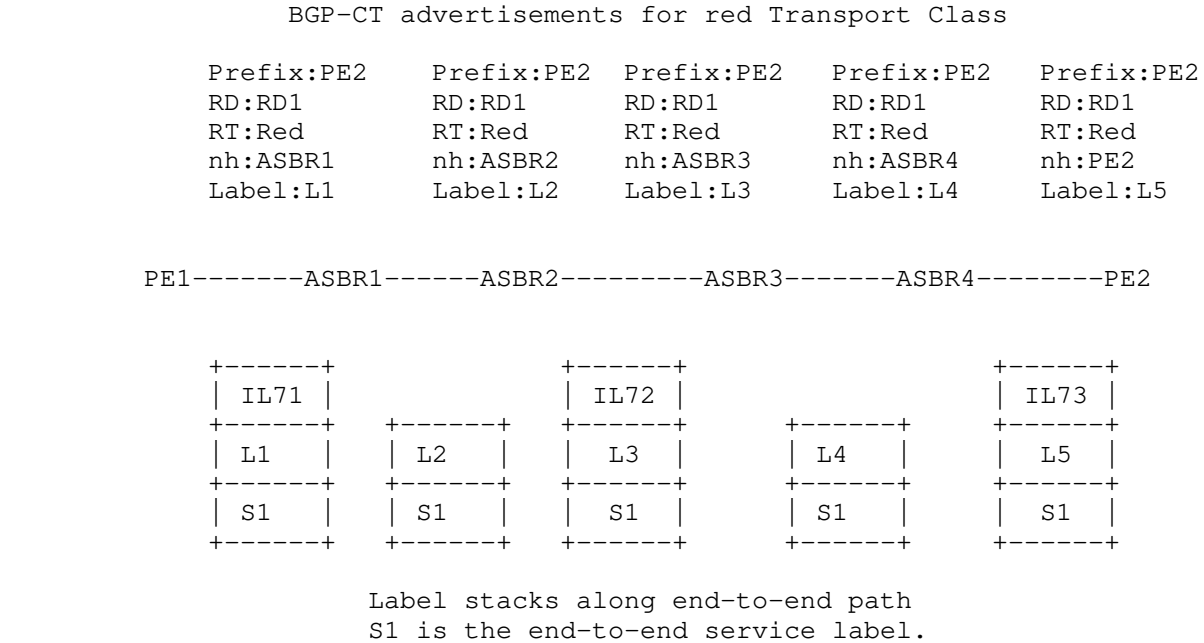


Figure 8: BGP-CT Advertisements and Label Stacks

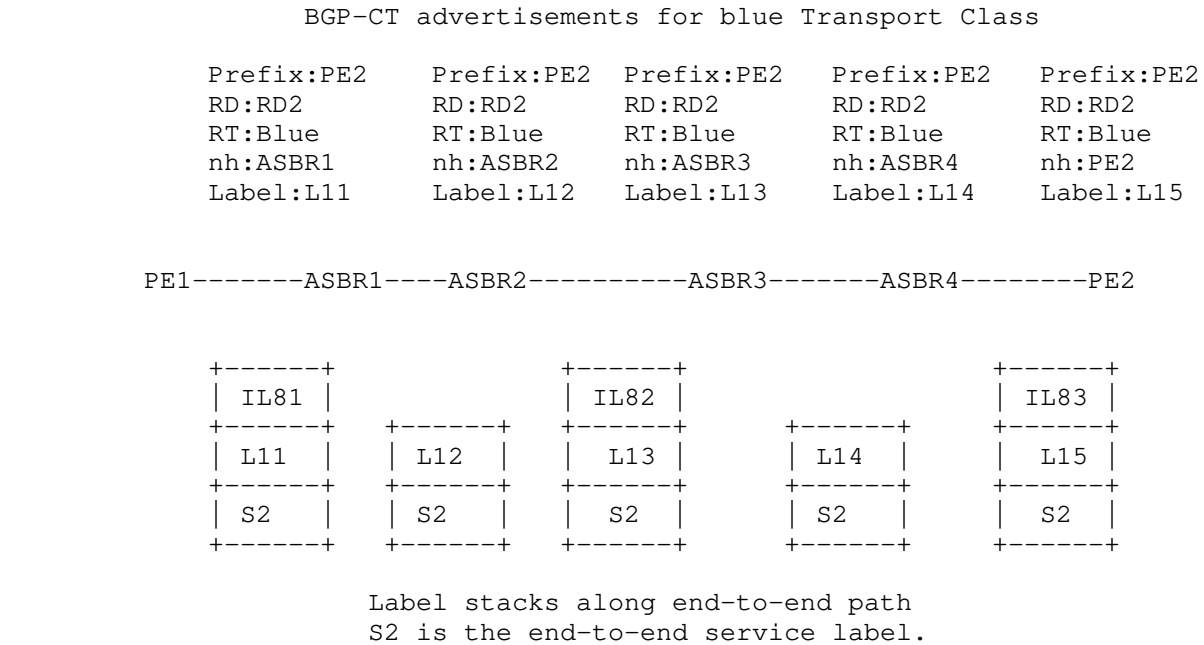


Figure 9: BGP-CT Advertisements and Label Stacks

For example, consider the diagram in Figure 8 and Figure 9 . The diagram shows the BGP-CT advertisements corresponding to two different end-to-end paths between PE1 and PE2. The two different paths belong to two different Transport Classes, red and blue. In order to create unique NLRIs for the two advertisements, PE2 uses two different RDs. In the example above, the red BGP-CT advertisement has an RD of RD1 and the blue BGP-CT advertisement has an RD of RD2. The advertisements will have RTs corresponding to the red and blue Transport Classes respectively. The RT MAY be directly mapped from the color extended community defined in [I-D.ietf-idr-tunnel-encaps]. In addition to the red and blue BGP-CT advertisments, the diagram shows the label stacks at different points along the end-to-end paths for the forwarding entries which are established by the two advertisements. Labels L1-L4 are red BGP-CT labels advertised by border nodes ASBR1,2,3,and 4, while label L5 is advertised by PE2 for the red Transport Class. Labels L11-L14 are blue BGP-CT labels advertised by border nodes ASBR1,2,3,and 4, while label L15 is advertised by PE2 for the blue Transport Class.

IL71, IL72, and IL73 represent tunnels internal to the domains 1, 2, and 3 which correspond to the red Transport Class. IL81, IL82, and IL83 represent tunnels internal to the domains 1, 2, and 3 which correspond to the blue Transport Class. In this example, we assume that the intra-domain tunnels correspond to SRTE policies having red SRTE-policy-color and blue SRTE-policy-color. Service labels are represented by S1 and S2. In this example, we assume that the service advertisement corresponding to S1 carries the red extended-color community, while the service advertisement corresponding to S2 carries the blue extended-color community. By default, the Transport Class carried in the BGP-CT route target maps to the extend-color community as well as the SRTE-policy-color. Therefore, based on the simple BGP-CT advertisment originated by PE2, PE1 is able to automatically steer traffic for service S1 over an end-to-end path made up of red SRTE policies in each domain.

Note that this example focuses on how signalling originated by PE2 results in forwarding state used by PE1 to reach PE2 on a specific Transport Class path. The solution supports the establishment of forwarding state for an arbitrary number of PEs to reach PE2. For example, PE3 in Figure 8 can reach PE2 on a red Transport Class path established using the same BGP-CT signalling. The signalling and forwarding state from ASBR1 all the way to PE2 is common to the paths used by both PE1 and PE3. This merging of signalling and forwarding state is essentially to the good scaling properties of the Seamless SR architecture. Millions of end-to-end Transport Class paths can be established in a scalable manner.

5.3. SLA Guarantee

5.3.1. Low latency

In a 5G network, many network functions are virtualized and distributed. Certain functions are time and latency sensitive. Latency is one of the main SLA parameter for 5G networks. In inter-domain networks, End-to-End latency measurement is required. Inside a domain, latency measurement mechanisms such as TWAMP [RFC5357] are used and link latency is advertised in IGP using extensions described in [RFC8570]and [RFC7471] .

[I-D.ietf-idr-performance-routing] extends the BGP AIGP attribute [RFC7311] by adding a sub TLV to carry an accumulated latency metric. The BGP best path selection algorithm used for a Transport Class requiring low latency will consider the accumulated latency metric to choose lowest latency path.

5.3.2. Traffic Engineering (TE) constraints

TE constraints generally include the ability to send traffic via certain nodes or links or avoid using certain nodes or links. In the Seamless SR architecture, the intra-domain transport technology is responsible for ensuring the TE constraints inside the domain, BGP-CT ensures that the end-to-end path is construct from intra-domain paths and inter-AS links that individually satisfy the TE constraints.

For example, in order to construct a pair of diverse paths, we can define a red and a blue Transport Class. Within each domain, the red and blue Transport Class path are realized using intra-domain path diversity mechansisms. For example, in a domain using flex-algo, red and blue Transport Classes are realized using red and blue flex-algo which don't share any links. To maintain path diversity on inter-AS links, BGP policies are used to associate two inter-AS peers with the red Transport Class and another two inter-AS peers with the blue Transport Class.

5.3.3. Bandwidth constraints

The Seamless SR architecture does not natively support end-to-end bandwidth reservations. In this architecture, the bandwidth utilization characteristics of each domain are managed independently. The intra-domain bandwidth management can make use of a variety of tools.

Link bandwidth extended community as defined in [I-D.ietf-idr-link-bandwidth] allows for efficient weighted load-balancing of traffic on multiple BGP-CT paths that belong to the same Transport Class. For optimized path placement, a seperate tool may be deployed and BGP policies/communites used for path placement.

5.4. Scalability

5.4.1. Access node scalability

The Seamless SR architecture needs to be able to accommodate very large numbers of access devices. These access devices are expected to be low-end devices with limited FIB capacity. The Seamless MPLS architecture, as described in [I-D.ietf-mpls-seamless-mpls], recommends the use of LDP DOD mode to limit the size of both the RIB and the FIB needed on the access devices. In the Seamless SR architecture, networks use IGP based label distribution and do not have this selective label request mechanism. However, RIB scalability of access nodes has not been a problem for real seamless MPLS deployments. In cases where access devices are low on CPU and memory and unable to support large a RIB, BGP filtering policies can

be applied at the ABR/ASBR routers to restrict the number of BGP-CT advertisements towards the access devices. The access devices will receive only the PE loopbacks that it needs to connect to.

5.4.2. Label stack depth

The ability for a device to push multiple MPLS labels on a packet depends on hardware capabilities. Access devices are expected to have limited label stack push capabilities. The Seamless SR architecture can provide cross-domain MPLS connectivity with a single label. The access devices push one service label, one BGP-CT label, and one intra-domain transport label. Assuming shortest path SR-MPLS in the access domain, the access domain transport will use a single label. Light weight traffic-engineering and slicing could also be achieved with a single label as described in [I-D.ietf-lsr-flex-algo]. The access nodes will need to be able to push a minimum of 3 labels.

5.4.3. Label Resources

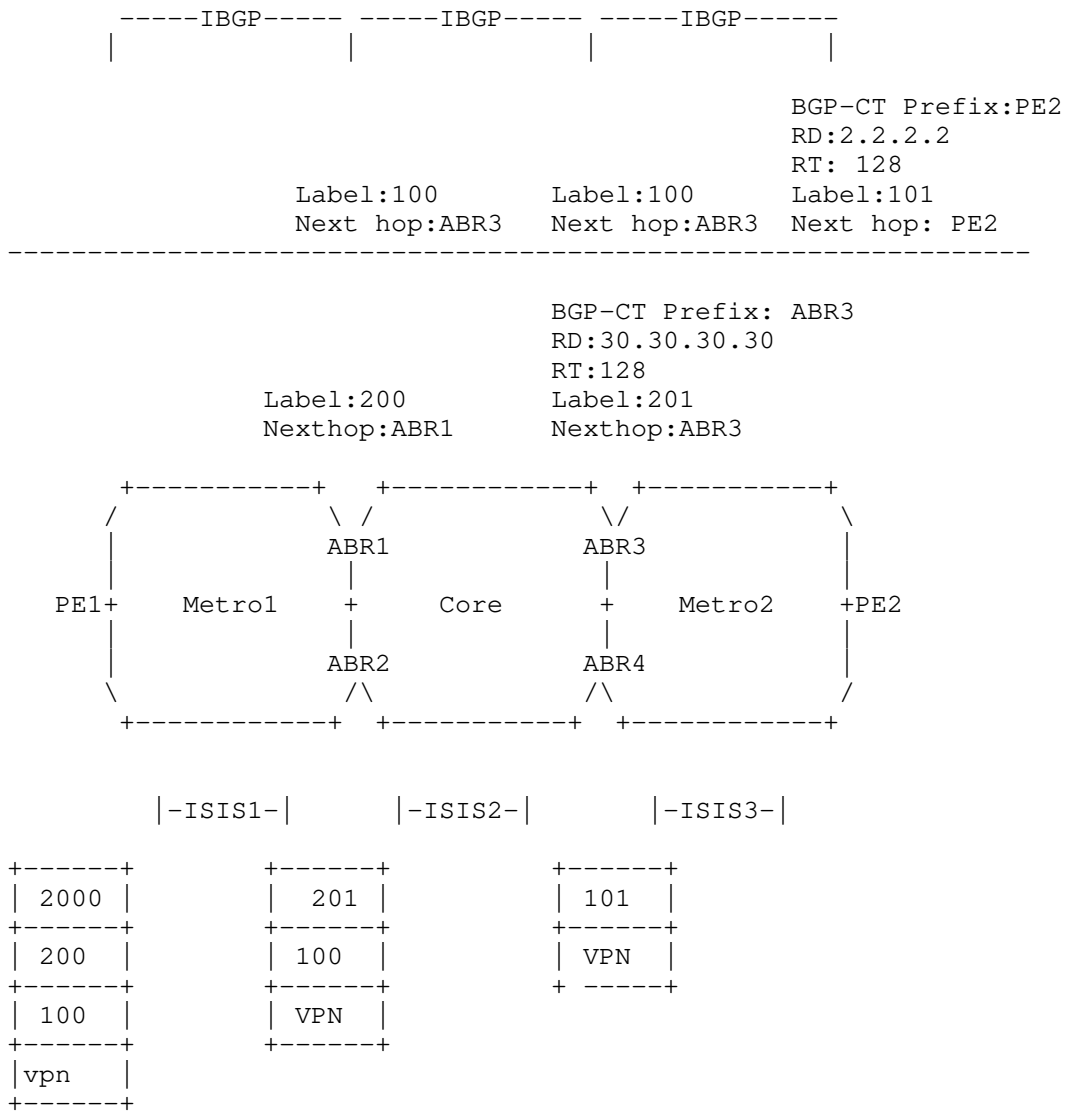


Figure 10: Recursive Route Resolution

The label resources are an important consideration in MPLS networks. On access devices, labels are consumed by services as well as for transport loopbacks inside IGP domain where the access device resides. For example, in the above diagram PE1 would have to allocate label resources equal to the number of customers connecting (i.e. the number of L2/L3 VPNs). Based on the size of the IGP domain

that PE1 resides in, it will also have to allocate labels for IGP loopbacks. This number is at most a few thousands. So overall a typical access device should have adequate label resources in Seamless SR architecture. The P routers need to allocate labels for IGP loopbacks. This number again is small. At most it will be a few thousand based on number of nodes in the largest IGP domains. The metro networks connect to the core network through ABRs. It is possible that a given ABR may end up having to maintain forwarding entries for a large subset of the transport loopback routes. There may be a large number of metro networks connecting to a given ABR, and in this case, the ABR will need forwarding entries for every access node in the directly connected metros. So, this ABR may have to maintain on the order of 100k routes. With BGP-CT each Transport Class will have to be separately allocated a label. So, in the above example, the ABR1 would have to use 300k labels if there were 3 Transport Classes. MPLS labels are 20 bit long and the label range of 16-1 million is available for general applications.This label space is shared between transport protocols and services. However, in a well-designed network, ABRs are not expected to host service routes. This leaves with 1 million labels completely available for transport infrastructure. This is sufficient in most cases.

In certain cases, it is desirable to reduce the forwarding state on the ABRs. This reduction can be achieved with label stacking as a result of recursive route resolution. In the Figure 10, PE2 advertises a BGP-CT prefix with nexthop being PE2 and 101 label. ABR3 advertises a label 100 for this BGP-CT prefix and changes the nexthop to self. When ABR1 receives this BGP-CT advertisement for PE2, it does not change the nexthop and advertises same label advertised by ABR3. When PE1 receives the BGP-CT advetisement for PE2 with a nexthop of ABR3, it resolves on another BGP-CT prefix for ABR3. As shown in the diagram, ABR3 advertises BGP-CT prefix with 201 and ABR1 advertises label 200 and sets nexthop to self. On PE1, the data packet consists of a VPN label at the bottom followed by 2 BGP-CT labels 100 and 200. The top most label 2000 is the transport label for the metro1 domain. There is 1 additional BGP-CT label on the datapacket.

Recursive route resolution provides significant forwarding state reduction on the ABRs. ABRs have to allocate label resources for the PE loopback that they directly connect to. This number is significantly lower as compared to the total number of PEs in the network.

5.5. Reliability

Transport layer redundancy is very important in 5G networks. Any link or node failure must be repaired with 50ms conevergence time. 50 ms convergence time can be achieved with Fast ReRoute (FRR) mechanisms. Seamless SR architecture supports Intra-domain link/node failures, Border node failures and the egress node and link failures for 50 ms convergence. Details of the FRR techniques are described in below sections.

5.5.1. Intra domain link and node protection

In the seamless SR architecture, protection against node and link failure is achieved with the relevant FRR techniques for the corresponding transport mechanism used inside the domain. In the case of an IP fabric, ECMP FRR or LFA can be used. In SR networks, TI-LFA [I-D.ietf-rtgwg-segment-routing-ti-lfa] provides link and node protection. For SR-TE [I-D.ietf-spring-segment-routing-policy] transport, link and node protection can be achieved using TI-LFA, combined with mechanisms described in [I-D.hegde-spring-node-protection-for-sr-te-paths].

5.5.2. Egress Link and node protection

[RFC8679] describes the mechanisms for providing protection for border nodes and PE devices where services are hosted. The mechanism can be further simplified operationally with anycast SIDs and anycast service labels, as described in [I-D.hegde-rtgwg-egress-protection-sr-networks].

5.5.3. Border Node protection

Border node protection is very important in a network consisting of multiple domains. Seamless SR architecture proposes to achieve 50ms FRR protection in the event of node failure with anycast address for the ABR/ASBRs and allocates same label for the BGP-CT Prefix.The detailed mechanism is described in [I-D.hegde-rtgwg-egress-protection-sr-networks].

5.6. Operations

5.6.1. MPLS ping and Traceroute

Seamless SR Architecture is based on hierarchical network modeling. The End-to-end BGP-CT connectivity can be verified. A new FEC is defined for BGP-CT as defined in draft [I-D.kaliraj-idr-bgp-classful-transport-planes] that describes End-to-End connectivity verification as well as fault isolation. The

BGP-CT verification happens only on the BGP nodes. The intra-domain connectivity verification and fault isolation will be based on the technology deployed in that domain as defined in [RFC8029] and [RFC8287].

5.6.2. Counters and Statistics

Traffic accounting and ability to build demand matrix for PE to PE traffic is very important. With BGP-CT, per-label transit counters should be supported on every transit router. per-label transit counters provide details of total traffic towards a remote PE measured at every BGP transit router. per-label egress counter should be supported on ingress PE router. per-label egress counter provides total traffic from ingress PE to the specific remote PE.

5.7. Service Mapping

Service mapping is an imprtant aspect of any architecture. It provides means to translate end users SLA requirements into operator's network configurations. Seamless SR architecture supports automatic steering with extended color community. The Transport Class and the route target carried by the BGP-CT advertisement directly map to the extended color community. Services that require specific SLA carry the extended color community which maps to the Transport Class to which the BGP-CT advertisement belongs.

Other types of traffic steering such as DSCP based forwarding is expressed with mapping-community. Mapping community is a standard BGP community and is completely generic and user defined. The mapping community will have a specific service mapping feature associated with it along with required fallback behaviour when the primary transport goes down. The below list provides a general guideline into the different service mapping features and fallback options an implementation should provide.

DSCP based mapping with each DSCP mapping to a Transport Class.

DSCP based mapping with default mapping to a best-effort transport

DSCP based mapping with fallback to best-effort when primary transport tunnel goes down.

Extended color community based mapping with fallback to best effort

Fallback options with specific protocol during migrations

Falback options to a different Transport Class.

No Fallback permitted.

5.8. Migrations

Networks that migrate from Seamless MPLS architecture to Seamless SR architecture, require that all the border nodes and PE devices be upgraded and enable new family on the BGP session. In cases where legacy nodes that cannot be upgraded exporting from BGP-LU into BGP-CT and vice versa SHOULD be supported.

5.9. Interworking with v6 transport technologies

A later version of this document will address interworking with other v6 technologies, including SRv6, SRm6, and MPLS over GRE6.

5.10. BGP based Multicast

BGP based multicast as described in draft [I-D.zzhang-bess-bgp-multicast] serves two main purposes. It can replace PIM/ mLDP inside a domain to natively do a BGP based multicast. It can also serve as an overlay stitching protocol to stitch multiple P2MP LSPs across the domain. This gives the ability to easily transition each domain independently from one technology to the other. BGP based multicast defines a new SAFI for carrying the MULTICAST TREE SAFI. Different route types are defined to support the various usecases.

6. Backward Compatibility

7. Security Considerations

TBD

8. IANA Considerations

9. Acknowledgements

Many thanks to Kireeti Kompella, Ron Bonica, Krzysztof Szarcowitz, Srihari Salngi,Julian Lucek for discussions and inputs.

10. Contributors

1.Kaliraj Vairavakkalai

Juniper Networks

kaliraj@juniper.net

2. Jeffrey Zhang

Juniper Networks

zzhang@juniper.net

11. References

11.1. Normative References

[I-D.hegde-rtgwg-egress-protection-sr-networks]
Hegde, S. and W. Lin, "Egress Protection for Segment Routing (SR) networks", draft-hegde-rtgwg-egress-protection-sr-networks-00 (work in progress), March 2020.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-02 (work in progress), October 2019.

[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.

[I-D.zzhang-bess-bgp-multicast]
Zhang, Z., Giuliano, L., Patel, K., Wijnands, I., mishra, m., and A. Gulko, "BGP Based Multicast", draft-zzhang-bess-bgp-multicast-03 (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>.

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

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

11.2. Informative References

[I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Node Protection for SR-TE Paths", draft-hegde-spring-node-protection-for-sr-te-paths-05 (work in progress), July 2019.

[I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-07 (work in progress), March 2018.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., and S. Ramachandra, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-15 (work in progress), December 2019.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-08 (work in progress), July 2020.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-03 (work in progress), March 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.voyer-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (work in progress), July 2020.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

[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>.

[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>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

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

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

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

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8679] Shen, Y., Jeganathan, M., Decraene, B., Gredler, H., Michel, C., and H. Chen, "MPLS Egress Protection Framework", RFC 8679, DOI 10.17487/RFC8679, December 2019, <https://www.rfc-editor.org/info/rfc8679>.

[TS.23.501-3GPP]
3rd Generation Partnership Project (3GPP), "System Architecture for 5G System; Stage 2, 3GPP TS 23.501 v16.4.0", March 2020.

Authors' Addresses

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

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks Inc.

Email: cbowers@juniper.net

Xiaohu Xu
Alibaba Inc.
Beijing
China

Email: xiaohu.xxh@alibaba-inc.com

Arkadiy Gulko
Refinitiv

Email: arkadiy.gulko@refinitiv.com

SPRING
Internet-Draft
Intended status: Standards Track
Expires: March 25, 2021

S. Hegde
C. Bowers
Juniper Networks Inc.
X. Xu
Alibaba Inc.
A. Gulko
Refinitiv
A. Bogdanov
Google Inc.
J. Uttaro
ATT
L. Jalil
Verizon

September 21, 2020

Seamless Segment Routing
draft-hegde-spring-mpls-seamless-sr-02

Abstract

In order to operate networks with large numbers of devices, network operators organize networks into multiple smaller network domains. Each network domain typically runs an IGP which has complete visibility within its own domain, but limited visibility outside of its domain. Seamless Segment Routing (Seamless SR) provides flexible, scalable and reliable end-to-end connectivity for services across independent network domains. Seamless SR accommodates domains using SR, LDP, and RSVP for MPLS label distribution as well as domains running IP without MPLS (IP-Fabric).

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/.

Table of Contents

1. Introduction

Evolving wireless access technology and cloud applications are expected to place new requirements on the packet transport networks. These services are contributing to significantly higher bandwidth throughput which in turn leads to a growing number of transport network devices. As an example, 5G networks are expected to require up to 250Gbps in the fronthaul and up to 400Gbps in the backhaul. There is a desire to allow many network functions to be virtualized and cloud native. In order to support latency-sensitive cloud-native network functions, packet transport networks should be capable of providing low-latency paths end-to-end. Some services will require low-latency paths while others may require different QoS properties. The network should be able to differentiate between the services and provide corresponding SLA transport paths. In addition, as these

applications become more sensitive and less loss tolerant, more and more emphasis is placed on overall service availability and reliability.

The Seamless SR architecture builds upon the Seamless MPLS architecture and caters to new requirements imposed by the 5G transport networks and the cloud applications. [I-D.ietf-mpls-seamless-mpls], contains a good description of the Seamless MPLS architecture. Although [I-D.ietf-mpls-seamless-mpls] has not been published as an RFC, it serves as a useful description of the Seamless MPLS architecture. [I-D.ietf-mpls-seamless-mpls] describes the Seamless MPLS architecture, which uses LDP and/or RSVP for intra-domain label distribution, and BGP-LU [RFC3107] for end-to-end label distribution. Seamless SR focuses on using segment routing for intra-domain label distribution. The mechansims described in this document are equally applicable to intra-domain tunneling mechanisms deployed using RSVP and/or LDP.

By using segment routing for intra-domain label distribution, Seamless SR is able to easily support both SR-MPLS on IPv4 and IPv6 networks. This overcomes a limitation of the classic Seamless MPLS architecture, which was limited to run MPLS on IPv4 networks in practice. Seamless SR (like Seamless MPLS) can use BGP-LU (RFC 3107) to stitch different domains. However, Seamless SR can also take advantage of BGP Prefix-SID [RFC8669] to provide predictable and deterministic labels for the inter-domain connectivity.

The basic functionality of the Seamless SR architecture does not require any enhancements to existing protocols. However, in order to support end-to-end service requirements across multiple domains, protocol extensions may be needed. This draft discusses use cases, requirements, and potential protocol enhancements.

2. Terminology

This document uses the following terminology

- Access Node (AN): An access node is a node which processes customers frames or packets at Layer 2 or above. This includes but is not limited to DSLAMs and Cell Site Routers in 5G networks. Access nodes have only limited MPLS functionalities in order to reduce complexity in the access network.

- Pre-Aggregation Node (P-AGG): A pre-aggregation node (P-AGG) is a node which aggregates several access nodes (ANs).

- Aggregation Node (AGG): A aggregation node (AGG) is a node which aggregates several pre-aggregation nodes (P-AGG).

- Area Border Router (ABR): Router between aggregation and core domain.

- Label Switch Router (LSR): Label Switch router are pure transit nodes. ideally have no customer or service state and are therefore decoupled from service creation.

- Use Case: Describes a typical network including service creation points and distribution of remote node loopback prefixes.

Figure 1: Terminology

3. Use Cases

3.1. Service provider network

Service provider transport networks use multiple domains to support scalability. For this analysis, we consider a representative network design with four level of hierarchy: access domains, pre-aggregation domains, aggregation domains and a core. (See Figure 2). The 5G transport networks in particular are expected to scale to very large number of access nodes due to the shorter range of the 5G radio technology. The networks are expected to scale up to one million nodes.

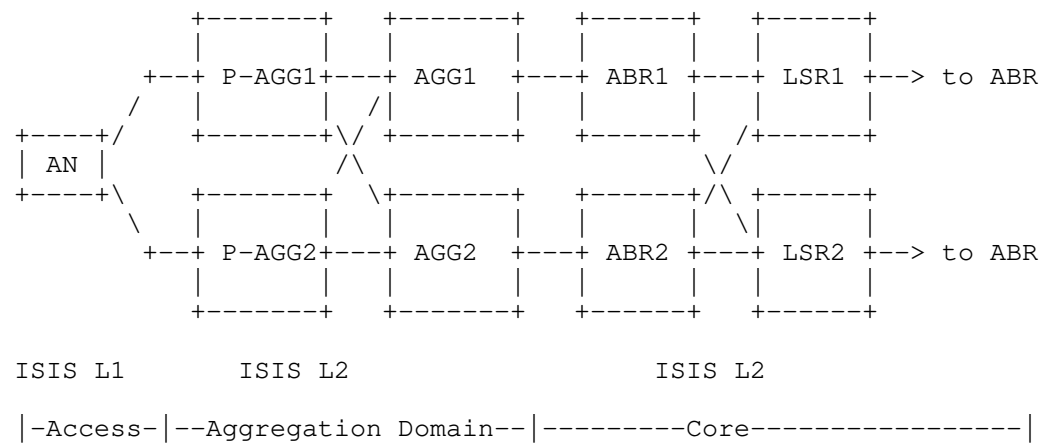


Figure 2: 5G network

Many network functions in a 5G network will be virtualized/ containerized and distributed across multiple data centers. Virtualized network functions are instantiated dynamically across different compute resources. This requires that the underlying transport network supports the stringent SLA on end-to-end paths.

5G networks support variety of service use cases that require end-to-end slicing. In certain cases the end-to-end connectivity requires differentiated forwarding capabilities. Seamless SR architecture should provide the ability to establish end-to-end paths that satisfy the required SLAs. For example, end user requirement could be to establish a low latency path end-to-end. The System Architecture for the 5G System [TS.23.501-3GPP] currently defines four standardized Slice/Service Types: Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communication (URLLC), massive Internet of Things (mIoT), Vehicle to everything (V2X). The Seamless SR should support end-to-end Service Level Objectives(SLO) to allow the creation of network slices with these four Slice/Service Types.

Many deployments consist of ring topologies in the access and aggregation networks. In the ring topologies, there are at most two forwarding paths for the traffic, whereas the core networks consist of nodes with more denser connectivity compared to ring topologies. Thus core networks may have a larger number of TE paths while access networks will have a smaller number of TE paths. The Seamless SR architecture should support the ability to have more TE paths in one domain and lesser number of TE paths in another domain and provide the ability to effectively connect the domains end-to-end while satisfying end-to-end constraints.

3.2. Large scale WAN networks

As WAN networks grow beyond several thousand nodes, it is often useful to divide the network into multiple IGP domains, as illustrated in Section 3.2. Separate IGP domains increase service availability by establishing a constrained failure domain. Smaller IGP domains may also improve network performance and health by reducing the device scale profile (including protocol and FIB scale).

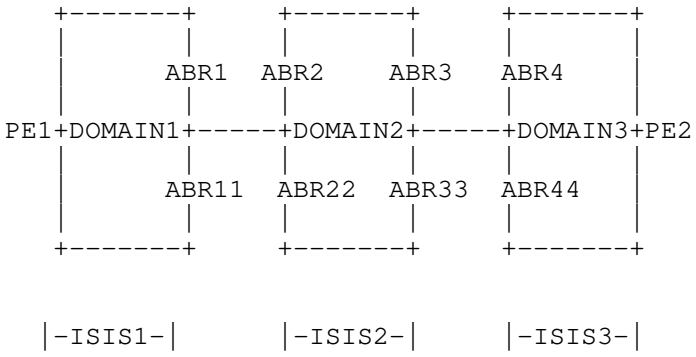


Figure 3: WAN Network

These Large WAN networks often cross national boundaries. In order to meet data sovereignty requirements, operators need to maintain strict control over end-to-end traffic-engineered(TE) paths. Segment Routing provides two main solutions to implement highly constrained TE paths. Flex-algo (defined in [I-D.ietf-lsr-flex-algo]) uses prefix-SIDs computed by all nodes in the IGP domain using the same pruned topology. Highly constrained TE paths for the data sovereignty use case can also be implemented using SR-TE policies ([I-D.ietf-spring-segment-routing-policy]) built using unprotected adjacency SIDs.

Both of these approaches work well for intra-domain TE paths. However, they both have limitations when one tries to extend them to the creation of highly constrained inter-domain TE paths. A goal of seamless SR is to be able to create highly constrained inter-domain TE paths in a scalable manner.

Some deployments may use a centralized controller to acquire the topologies of multiple domains and build end-to-end constrained paths. This can be scaled with hierarchical controllers. However, there is still significant risk of a loss of network connectivity to one or more controllers, which can result in a failure to satisfy the

strict requirements of data sovereignty. The network should have pre-established TE paths end-to-end that don't rely on controllers in order to address these failure scenarios.

3.3. Data Center Interconnect (DCI) Networks

Data centers are playing an increasingly important role in providing access to information and applications. Geographically diverse data centers usually connect via a high speed, reliable and secure core network.

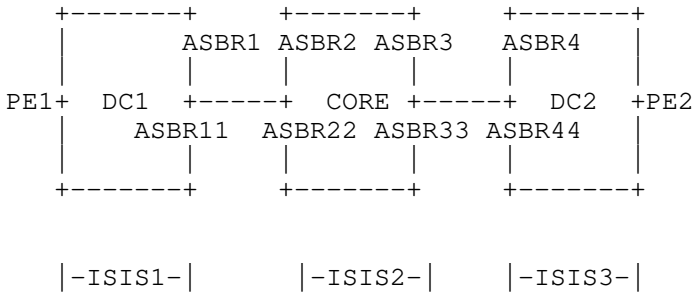


Figure 4: DCI Network

In many Data Center deployments, applications require end-to-end path diversity and/or end-to-end low latency paths. It is desirable to have a uniform technology deployed in the core as well as in the Data Centers to create these SLA paths. Such uniformity simplifies the network to a great extent. It is desirable for a solution to only require service-related configurations on the access end-points where services are attached, avoiding service-related configurations on the ABR/ASBR nodes.

3.4. Multicast Use cases

Multicast services such as IPTV and multicast also need to be support across a multi-domain service provider network. Multicast services such as IPTV, multicast VPN etc need to be supported in a service provider network.

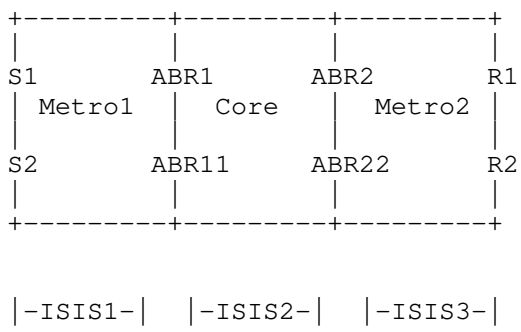


Figure 5: Multicast usecases

Figure 5 shows a simplified multi-domain network supporting multicast. Multicast sources S1 and S2 lie in a different domain from the receivers R1 and R2. Using multiple IGP domains presents a problem for the establishment of multicast replication trees. Typically, a multicast receiver does a reverse path forwarding (RPF) lookup for a multicast source. One solution is to leak the routes for multicast sources across the IGP domains. However, this can compromise the scaling properties of the multi-domain architecture. SR-P2MP [I-D.voyer-pim-sr-p2mp-policy] offers a solution for both intra-domain and inter-domain multicast. However, it does not accommodate deployments using existing intra-domain multicast technology, such as mLDP [RFC6388] in some of the domains. A solution should accommodate a mixture of existing and newer technologies to better facilitate coexistence and migration.

4. Requirements

This section provides a summary of requirements derived from the use cases described in previous sections.

4.1. MPLS Transport

The architecture SHOULD provide MPLS transport between two service endpoints regardless of whether the two end-points are in the same IGP domain, different IGP domains, or in different autonomous systems.

The MPLS transport SHOULD be supported on IPv4, IPv6, and dual-stack networks.

4.2. SLA Guarantee

The architecture SHOULD allow the creation of paths that support end-to-end SLAs. The paths should for example obey constraints related to latency, diversity, bandwidth and availability.

The architecture SHOULD support end-to-end network slicing as described by 5G transport requirements [TS.23.501-3GPP].

4.3. Scalability

The architecture SHOULD be able to support up to 1 million nodes.

The architecture SHOULD facilitate the use of access nodes with low RIB/FIB and low CPU capabilities.

The architecture SHOULD facilitate the use of access nodes with low label stacking capability.

The architecture SHOULD allow for a scalable response to network events. An individual node SHOULD only need to respond to a limited subset of network events.

Service routes on the border nodes SHOULD be minimized.

4.4. Availability

Traffic SHOULD be Fast Reroute (FRR) protected against link, node, and SRLG failures within a domain.

Traffic SHOULD be Fast Reroute (FRR) protected against border node failures.

Traffic SHOULD be Fast Reroute (FRR) protected against egress node and egress link failures.

4.5. Operations

Each domain SHOULD be independent and SHOULD not depend on the transport technology in another domain. This allows for more flexible evolution of the network.

Basic MPLS OAM mechanisms described in [RFC8029] SHOULD be supported.

End-to-end mpls ping and traceroute procedures SHOULD be supported.

Ability to validate the path inside each domain SHOULD be supported.

Statistics for inter-domain paths on the ingress and egress PE nodes as well as border nodes SHOULD be supported.

4.6. Service Mapping

The architecture SHOULD support the automated steering of traffic on to transport paths based on communities carried in the service prefix advertisements.

The architecture SHOULD support the steering of traffic on to transport paths based on the DSCP value carried in IPv4/IPv6 packets.

Traffic steering based on EXP bits in the mpls header SHOULD be supported.

Traffic steering based on 5-tuple packet filter SHOULD be supported. Source address, destination address, source port, destination port and protocol fields should be allowed.

All traffic steering mechanims SHOULD be supported for all kinds of service traffic including VPN traffic as well as global internet traffic.

The core domain is expected to have more traffic engineering constraints as compared to metros. The ability to map the services to appropriate transport tunnels at service attachment points SHOULD be supported.

5. Seamless Segment Routing architecture

5.1. Solution Concepts

The solution described below makes use of the following concepts.

- Transport Class (TC): A Transport Class is defined as a collection of end-to-end MPLS paths that satisfy a set of constraints or Service Level Agreements.

- BGP-Classful Transport (BGP-CT): A new BGP family used to establish Transport Class paths across different domains.

- Route Distinguisher (RD): The Route Distinguisher is defined in RFC4364. In BGP-CT, the RD is used in BGP advertisements to differentiate multiple paths to the same loopback address. It may be useful to automatically generate RDs in order to simplify configuration.

- Route Target (RT): The Route Target extended community is carried in BGP-CT advertisements. The RT represents the Transport Class of an advertised path. Note that the RT is only carried in the BGP-CT advertisements. No BGP-VPN related configuration or VPN family advertisements are needed when BGP-CT transport paths are used to carry non-VPN traffic.

- Mapping Community (MC): The Mapping Community is the BGP extended community as defined in RFC4360. In the Seamless SR architecture, an MC is carried by a BGP-CT route and/or a service route. The MC is used to identify the specific local policy used to map traffic for a service route to different Transport Class paths. When a mapping community is advertised in a BGP-CT route it identifies the specific local policy used to map the BGP-CT route to the intra-domain tunnels.The local policy can include additional traffic steering properties for placing traffic on different Transport Class paths. The values of the MCs and the corresponding local policies for service mapping are defined by the network operator.

Figure 6: Solution Concepts

5.2. BGP Classful Transport

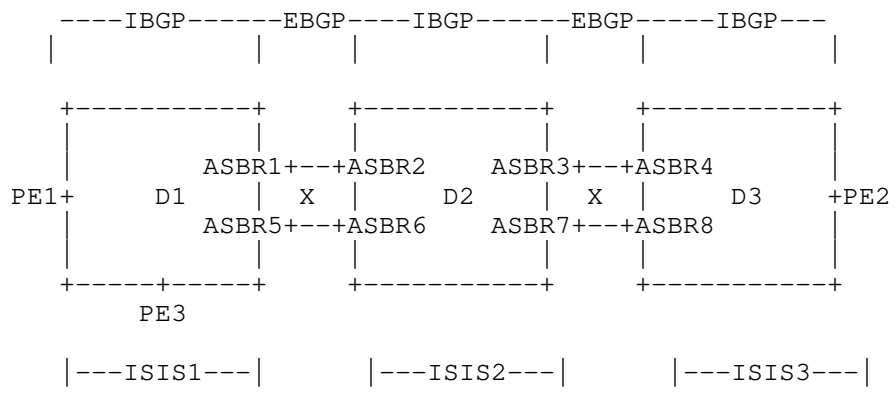


Figure 7: WAN Network

The above diagram shows a WAN network divided into 3 different domains. Within each domain, BGP sessions are established between the PE nodes and the border nodes as well as between border nodes. BGP sessions are also established between border nodes across domains. The goal is for PE1 to have MPLS connectivity to PE2, satisfying specific characteristics. Multiple MPLS paths from PE1 to PE2 are required in order to satisfy different SLAs. [I-D.kaliraj-idr-bgp-classful-transport-planes] defines a new BGP family called BGP-Classful Transport. The NLRI for this new family consists of a prefix and a Route Distinguisher. The prefix corresponds to the loopback of the destination PE, and RD is used to distinguish different paths to the same PE loopback. The BGP-CT advertisement also carries a Route Target. The RT specifies the Transport Class to which the BGP-CT advertisement belongs. BGP-CT mechanisms are applicable to single ownership networks that are organized into multiple domains. It is also applicable to multiple ASes with different ownership but closely co-operating administration. BGP-CT mechansims are not expected to be applied on the internet peering or between domains that have completely independent administrations.

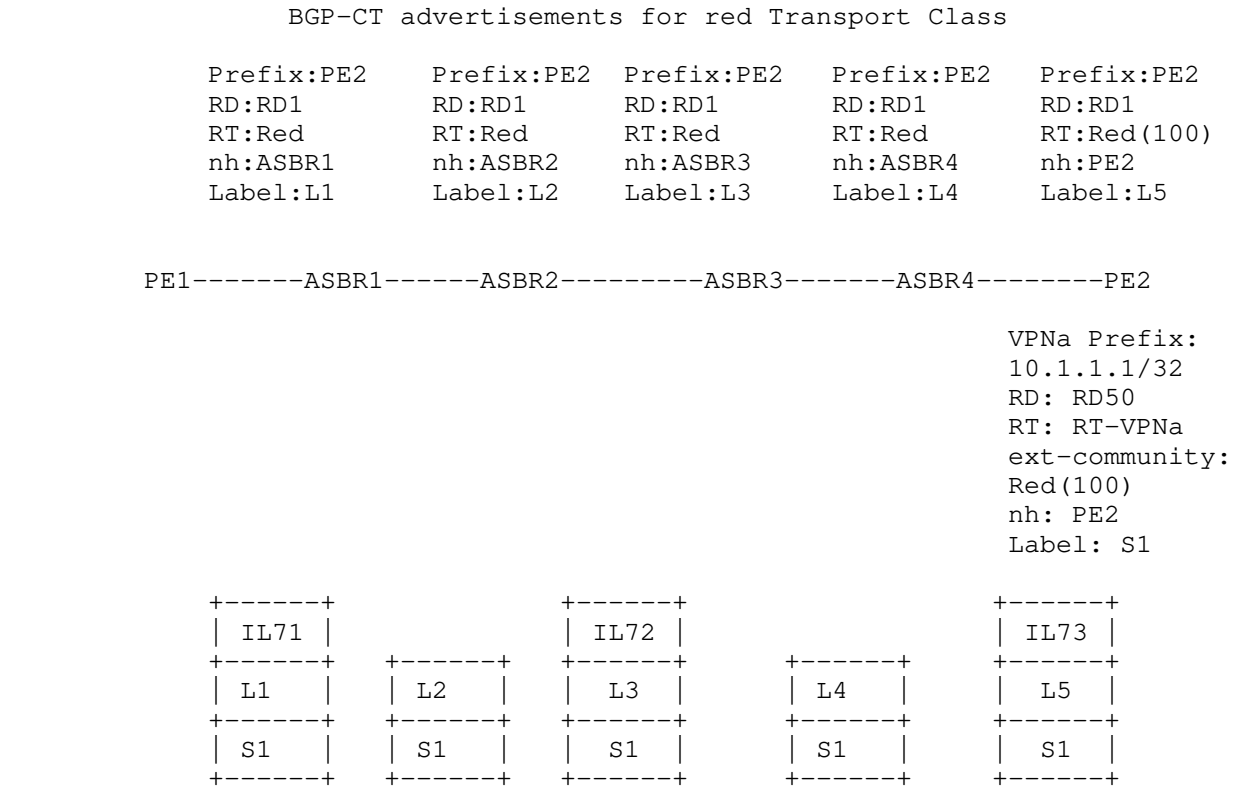


Figure 8: BGP-CT Advertisements and Label Stacks

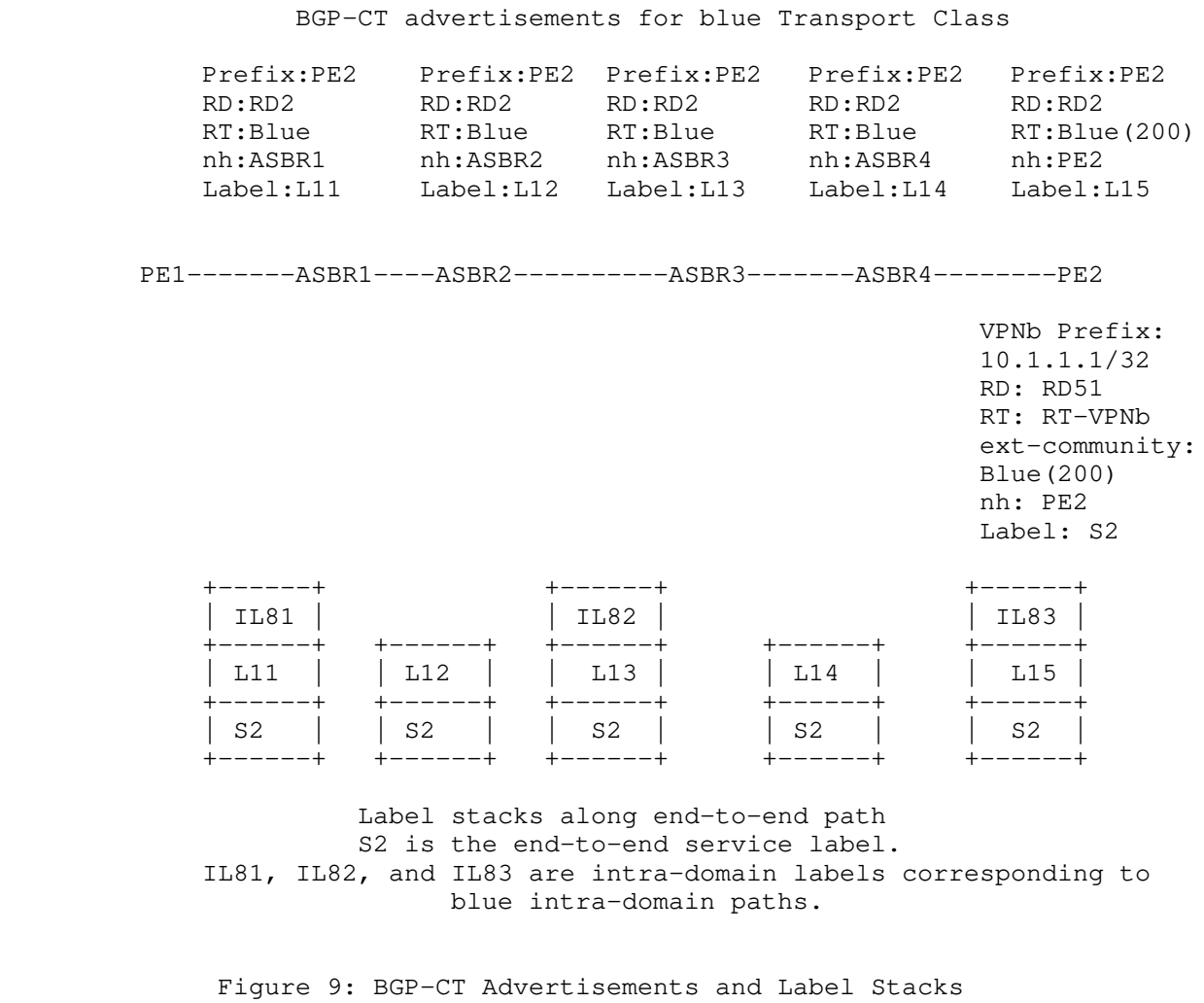


Figure 9: BGP-CT Advertisements and Label Stacks

For example, consider the diagram in Figure 8 and Figure 9 . The diagram shows the BGP-CT advertisements corresponding to two different end-to-end paths between PE1 and PE2. The two different paths belong to two different Transport Classes, red and blue.

The inter-domain paths created by BGP-CT Transport Classes can be used by any traffic that can be steered using BGP next-hop resolution, including vanilla IPv4 and IPv6, L2VPN, L3VPN, and eVPN. In the example above, we show how traffic from two different L3VPNs (VPNa and VPNb) is mapped onto two different BGP-CT Transport Classes (Red and Blue). The L3VPN advertisements for VPNa and VPNb are originated by PE2 as usual. PE1 receives these L3VPN advertisements

and uses the next-hop in the L3VPN advertisements to determine the path to use. In the absence of any BGP-CT Transport Classes in the network, PE1 would likely resolve the L3VPN next-hop over BGP-LU routes corresponding to the BGP best path. However, when BGP-CT Transport Classes are used, PE1 will resolve the L3VPN next-hop over a BGP-CT route.

In the example above, PE2 originates BGP-CT advertisements for the Red and Blue Transport Classes. These BGP-CT advertisements propagate across the multiple domains, causing forwarding state for the two Transport Classes to be installed at ABRs along the way. In order to create unique NLRIs for the two advertisements, PE2 uses two different RDs. In the example above, the red BGP-CT advertisement has an RD of RD1 and the blue BGP-CT advertisement has an RD of RD2. Note that the RD values used in the BGP-CT advertisement are completely independent of the RD values used in the L3VPN advertisements. In both cases, the RD values are simply a mechanism to guarantee uniqueness of a prefix/RD pair.

The RT values used in the BGP-CT advertisements are unrelated to the RT values used on the L3VPN advertisements. The L3VPN RT values identify VPN membership, as usual. The BGP-CT RT values identify Transport Class membership. In order to be able to easily map VPN traffic into BGP-CT Transport classes, it can be useful however to make an association between BGP-CT RT values and color extended community values in the L3VPN advertisements. In the example above,the RT value carried in the BGP-CT advertisement originated from PE2 for the red Transport Class is configured to correspond to the color extended community advertised in the VPN advertisement for VPNa. Similarly, the RT value for the blue Transport Class corresponds to the color extended community for VPNb. In this way, traffic on PE1 for each VPN can be mapped to a tranport class path by associating the value of the color extended community carried in the VPN advertisement with an RT value carried in a BGP-CT advertisement.

The example above also shows the label stacks at different points along the end-to-end paths for the forwarding entries which are established by the two advertisements. Labels L1-L4 are red BGP-CT labels advertised by border nodes ASBR1,2,3,and 4, while label L5 is advertised by PE2 for the red Transport Class. Labels L11-L14 are blue BGP-CT labels advertised by border nodes ASBR1,2,3,and 4, while label L15 is advertised by PE2 for the blue Transport Class.

IL71, IL72, and IL73 represent tunnels internal to the domains 1, 2, and 3 which correspond to the red Transport Class. IL81, IL82, and IL83 represent tunnels internal to the domains 1, 2, and 3 which correspond to the blue Transport Class. In this example, we assume that the intra-domain tunnels correspond to SRTE policies having red

SRTE-policy-color and blue SRTE-policy-color. Service labels are represented by S1 and S2.

Note that this example focuses on how signalling originated by PE2 results in forwarding state used by PE1 to reach PE2 on a specific Transport Class path. The solution supports the establishment of forwarding state for an arbitrary number of PEs to reach PE2. For example, PE3 in Figure 8 can reach PE2 on a red Transport Class path established using the same BGP-CT signalling. The signalling and forwarding state from ASBR1 all the way to PE2 is common to the paths used by both PE1 and PE3. This merging of signalling and forwarding state is essentially to the good scaling properties of the Seamless SR architecture. Millions of end-to-end Transport Class paths can be established in a scalable manner.

5.3. Automatically Creating Transport Classes

In order to simplify the creation of inter-domain paths, it may be desirable to automatically advertise a BGP-CT Transport Class based on the existence of an intra-domain tunnel. The RT value used on the BGP-CT advertisement is automatically derived from a property of the intra-domain tunnel that triggered its creation. How the Transpor Class RT value is derived for different types of intra-domain tunnels is discussed below.

5.3.1. Automatically Creating Transport Classes for BGP-SR-TE Intra-domain Tunnels

When the intra-domain tunnel is a BGP-SR-TE policy [I-D.ietf-idr-segment-routing-te-policy], the value of the Transport Class RT in the corresponding BGP-CT advertisement is derived from the Policy Color contained in SR Policy NLRI. The 32-bit Policy Color is directly converted to a 32-bit Transport Class RT.

5.3.2. Automatically Creating Transport Classes for Flex-Algo Tunnels

When the intra-domain tunnel is created using Flex-Algo [I-D.ietf-lsr-flex-algo], the value of the Transport Class RT in the corresponding BGP-CT advertisement is derived from the 8-bit Algorithm value carried in SR-Algorithm sub-TLV (RFC8667). The conversion from 8-bit Algorithm value to 32-bit Transport Class RT is done by treating both as unsigned integers. Note that this definition allows for intra-domain tunnels created via standardized algorithm (0-127) as well as flex-algo (128-255).

5.3.3. Auto-deriving Transport Classes for PCEP

When the intra-domain tunnel is created using PCEP, the value of the Transport Class RT in the corresponding BGP-CT advertisement is derived from the Color of the SR Policy Identifiers TLV defined in [I-D.ietf-pce-segment-routing-policy-cp]. The 32-bit Color is directly converted to a 32-bit Transport Class RT.

5.4. Inter-domain flex-algo with BGP-CT

Flex-algo (defined in [I-D.ietf-lsr-flex-algo]) provides a mechanism to separate routing planes. Multiple algorithms are defined and prefix-SIDs are advertised for each algorithm. BGP-CT can be used to advertise these flex-algo SIDs in BGP-CT. BGP Prefix-SID (RFC 8669) is an attribute and can be carried in the BGP-CT NLRI. Multiple transport classes that correspond to each of the flex-algo in IGP domain are defined. These Transport Classes advertise the IGP flex-algo SIDs in the prefix-SIDs attribute in the BGP-CT NLRI.

5.5. Applicability to color-only policies

Color-only policies consist of (nullEndpont, color) as specified in [I-D.ietf-spring-segment-routing-policy]. Special steering mechanisms are defined with "CO" flags defined in the color extended community [I-D.ietf-idr-segment-routing-te-policy]. Color-only policies can be advertised in BGP-CT with the prefix being NULL (0.0.0.0/32 or 0::0/128). Seperate RD will be advertised for each NULL advertisement with different color. The Route target carries the Policy Color contained in SR Policy NLRI. The steering mechanisms defined in [I-D.ietf-spring-segment-routing-policy] MUST be honoured while resolving services prefixes on the BGP-CT advertisements.

5.6. Data sovereignty

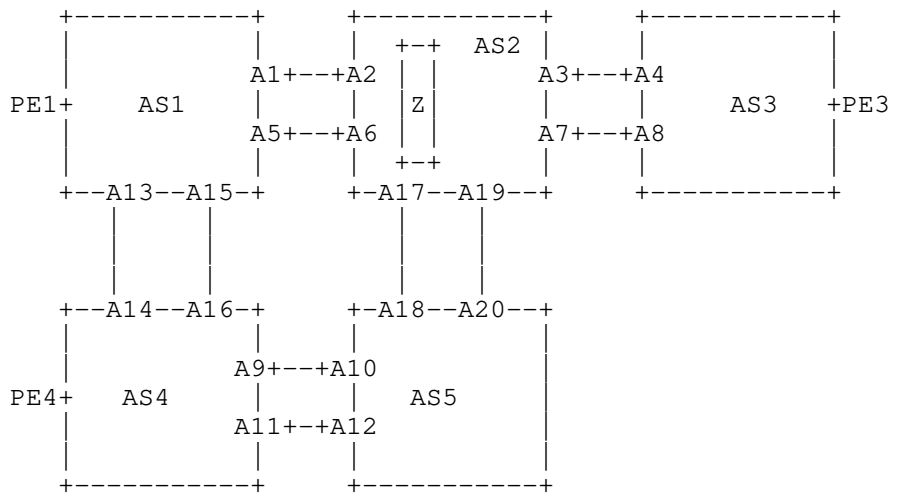


Figure 10: Multi domain Network

Consider a WAN network with multiple ASes as shown in the diagram Figure 10. The ASes roughly correspond to the geographical location of the nodes. In this example, we assume that each AS corresponds to a continent. The data sovereignty requirement in this example is that certain traffic from PE1(in AS1) to PE3(in AS3) must not cross through country Z in AS2. As indicate by the location of country Z in the diagram, all paths that go directly from AS1 to AS3 through AS2 necessarily passes through country Z. Using BGP-LU to provide connectivity from PE1 to PE3 would generally result in a path that goes from AS1 to AS2 to AS3, which does not satisfy the data sovereignty requirement in this example. Instead, the solution using BGP-CT will go from AS1 to AS4 to AS5 to AS2 to AS3. BGP-CT will ensure that when the traffic passes through AS2, only intra-domain paths satisfying the data sovereignty requirement will be used.

Within AS2, there are several different intra-domain TE mechanisms that can be used to exclude links that pass through country Z. For example, RSVP-TE or flex-algo can be used to create intra-domain paths that satisfy the data sovereignty requirement. BGP-CT allows the constrained intra-domain paths to satisfy requirements for end-to-end inter-domain paths. LSPs created by RSVP-TE or Flex-algo that satisfy the "exclude country Z" constraint are associated with a color Green. A Green Transport Class is defined on border nodes in all ASes. This Green Transport Class is associated with a mapping community called Not-Z.

In AS2, the ASBRs are configured such that the presence of the mapping community Not-Z in BGP-CT routes results in a strict route resolution mechanism for those routes. A BGP-CT route carrying the color extended community Not-Z will only resolve on the Green Tranport Class. So it will only use Green intra-domain tunnels.

In AS1, AS3, AS4, and AS5, no links pass through country Z, so all intra-domain paths automatically satisfy the data sovereignty requirement. So there is no need for the creation of Green intra-domain tunnels. In these ASes, the presence of the mapping community Not-Z in BGP-CT routes results in resolution on best-effort paths. Even though the ASBRs in these ASes do not need to create Green intra-domain tunnels, they still need to allocate labels to identify traffic using the Green Transport Class. These labels will be used by the ASBRs in AS2 to put traffic on the Green intra-domain tunnels in AS2.

The requirement is that only a subset of traffic honor the data sovereignty requirement. The service prefixes from PE1 to PE2 that need to honor the data sovereignty requirement will be associated with Green extended color community in the service advertisements. This will result in PE1 using the BGP-CT labels corresponding to {PE2, Green} to forward the traffic. BGP-CT labels corresponding to {PE2, Green} will exist at every ASBR along the path. The traffic originating on PE1, will be associated with Green color community. The bottom-most label in the packet consists of a VPN label. Above the VPN label, BGP-CT label is imposed. Above BGP-CT label, the intra-domain transport label is imposed. Let us assume the traffic from PE1 needs to go to PE2 through AS1, AS4, AS5, AS2, and AS3. The BGP-CT label for {PE2, Green} will be swapped at the border nodes.

Note that end-to-end inter-domain data sovereignty can in principle be accomplished using BGP-LU with multiple loopbacks and associating those loopbacks to appropriate transport tunnels at every border node in every domain. This is very configuration intensive and require multiple loopbacks. BGP-CT builds on the basic mechanisms of BGP-LU while greatly simplifying such use cases.

5.7. Interconnecting IP Fabric Data Centers

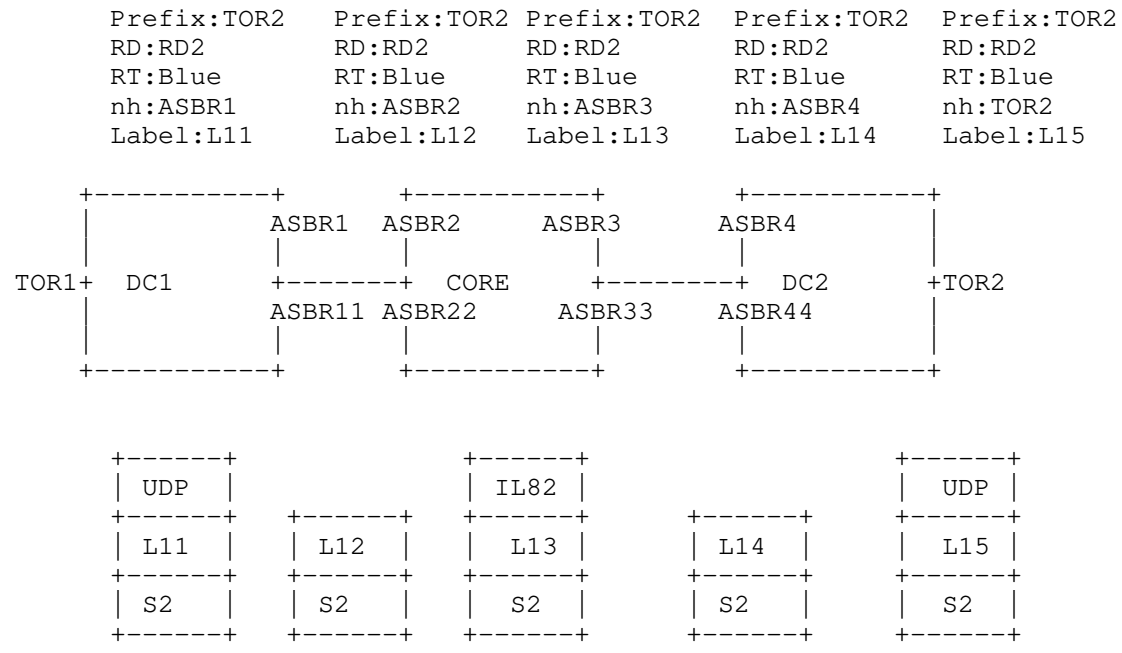


Figure 11: Operation in IP fabric

Many data center networks consist of IP fabrics which do not have MPLS packet processing capability. A common requirement is that traffic originated from an IP Fabric data center needs to satisfy certain constraints in the MPLS-enable core, for example, only using a subset of links (blue links). It is useful for the traffic originating in an IP Fabric DC to carry information that allows the MPLS-enable core to treat it accordingly. MPLSoUDP, as defined in [RFC7510], is a mechanism where a UDP header is imposed on an MPLS packets on the border nodes. In Figure 11 above, the traffic needs to take blue paths in the core. The Blue Transport Class is defined on the ASBRs. In the core, Blue intra-domain tunnels are created. The BGP-CT advertisements for the Blue Transport Class are as shown in the diagram. The BGP-CT advertisements originate at TOR2 and propagate through all the ASBRs, until finally reaching TOR1. Within DC1, traffic is encapsulated with a UDP header. Traffic with the UDP header gets decapsulated at ASBR1. The traffic follows Blue paths in

the core. At ASBR4, the MPLS packet gets encapsulated with a UDP header. The UDP header is removed at TOR2, and the lookup will be done for the service label.

5.8. Translating Transport Classes across Domains

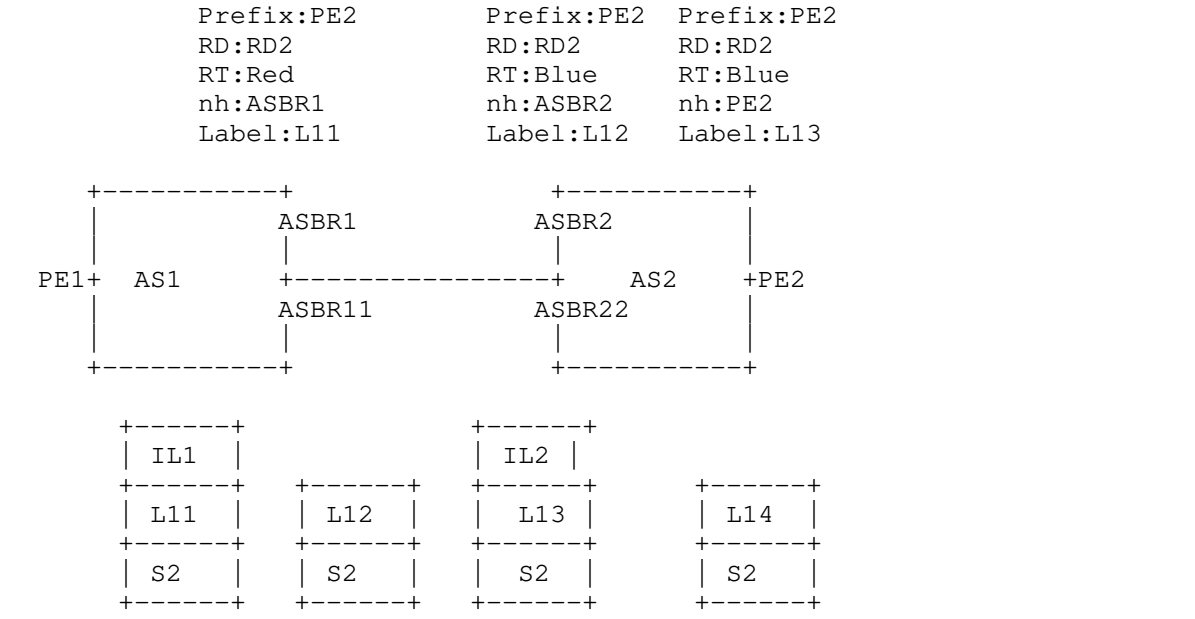


Figure 12: Translating Transport Classes across Domains

In certain scenarios, the TE intent represented by Transport Classes may differ from one domain to another. This could be the result of two independent organizations merging into one. It could also occur when two ASes are under different administration, but use BGP-CT to provide an end-to-end service. In both scenarios, the same color may represent different intent in each domain. When the traffic needs to satisfy certain TE characteristic, the colors need to be mapped correctly at the border. In the example in Figure 12, there are two ASes. The low latency TE intent is represented with the Red

Transport Class in AS1 and with the Blue Transport Class in AS2. PE2 advertises a BGP-CT prefix with RT of Blue. ASBR2 sets the nexthop to self and advertises a new label L12. On ASBR1, the Blue BGP-CT advertisement is imported into the Red Transport RIB and the advertisement from ASBR1 will carry a Red RT. This ensures that the BGP-CT prefix for PE2 resolves on a Red intra-domain path in AS1.

5.9. SLA Guarantee

5.9.1. Low latency

Many network functions are virtualized and distributed. Certain functions are time and latency sensitive. In inter-domain networks, End-to-End latency measurement is required. Inside a domain, latency measurement mechanisms such as TWAMP [RFC5357] are used and link latency is advertised in IGP using extensions described in [RFC8570]and [RFC7471] .

[I-D.ietf-idr-performance-routing] extends the BGP AIGP attribute [RFC7311] by adding a sub TLV to carry an accumulated latency metric. The BGP best path selection algorithm used for a Transport Class requiring low latency will consider the accumulated latency metric to choose the lowest latency path.

5.9.2. Traffic Engineering (TE) constraints

TE constraints generally include the ability to send traffic via certain nodes or links or avoid using certain nodes or links. In the Seamless SR architecture, the intra-domain transport technology is responsible for ensuring the TE constraints inside the domain, BGP-CT ensures that the end-to-end path is constructed from intra-domain paths and inter-AS links that individually satisfy the TE constraints.

For example, in order to construct a pair of diverse paths, we can define a red and a blue Transport Class. Within each domain, the red and blue Transport Class path are realized using intra-domain path diversity mechanisms. For example, in a domain using flex-algo, red and blue Transport Classes are realized using red and blue flex-algo definitions (FAD) which don’t share any links. To maintain path diversity on inter-AS links, BGP policies are used to associate two inter-AS peers with the red Transport Class and another two inter-AS peers with the blue Transport Class.

5.9.3. Bandwidth constraints

The Seamless SR architecture does not natively support end-to-end bandwidth reservations. In this architecture, the bandwidth utilization characteristics of each domain are managed independently. The intra-domain bandwidth management can make use of a variety of tools.

Link bandwidth extended community as defined in [I-D.ietf-idr-link-bandwidth] allows for efficient weighted load-balancing of traffic on multiple BGP-CT paths that belong to the same Transport Class. For optimized path placement, a centralized TE system may be deployed with BGP policies/communities used for path placement.

5.10. Scalability

5.10.1. Access node scalability

The Seamless SR architecture needs to be able to accommodate very large numbers of access devices. These access devices are expected to be low-end devices with limited FIB capacity. The Seamless MPLS architecture, as described in [I-D.ietf-mpls-seamless-mpls], recommends the use of LDP DOD mode to limit the size of both the RIB and the FIB needed on the access devices. In the Seamless SR architecture, networks use IGP-based label distribution and do not have this selective label request mechanism. However, RIB scalability of access nodes has not been a problem for real seamless MPLS deployments. In cases where access devices are low on CPU and memory and unable to support large a RIB, BGP filtering policies can be applied at the ABR/ASBR routers to restrict the number of BGP-CT advertisements towards the access devices. The access devices will receive only the PE loopbacks that it needs to connect to.

5.10.2. Label stack depth

The ability for a device to push multiple MPLS labels on a packet depends on hardware capabilities. Access devices are expected to have limited label stack push capabilities. Assuming shortest path SR-MPLS in the access domain, the access domain transport will use a single label. Lightweight traffic-engineering and slicing could also be achieved with a single label as described in [I-D.ietf-lsr-flex-algo]. The Seamless SR architecture can provide cross-domain MPLS connectivity with a single label. Assuming the use of a service label, end-to-end connectivity is provided by pushing one service label, one BGP-CT label, and one intra-domain transport label (which could also be a Binding-SID). Therefore, access nodes will only need to be able to push 3 labels for most applications.

5.10.3. Label Resources

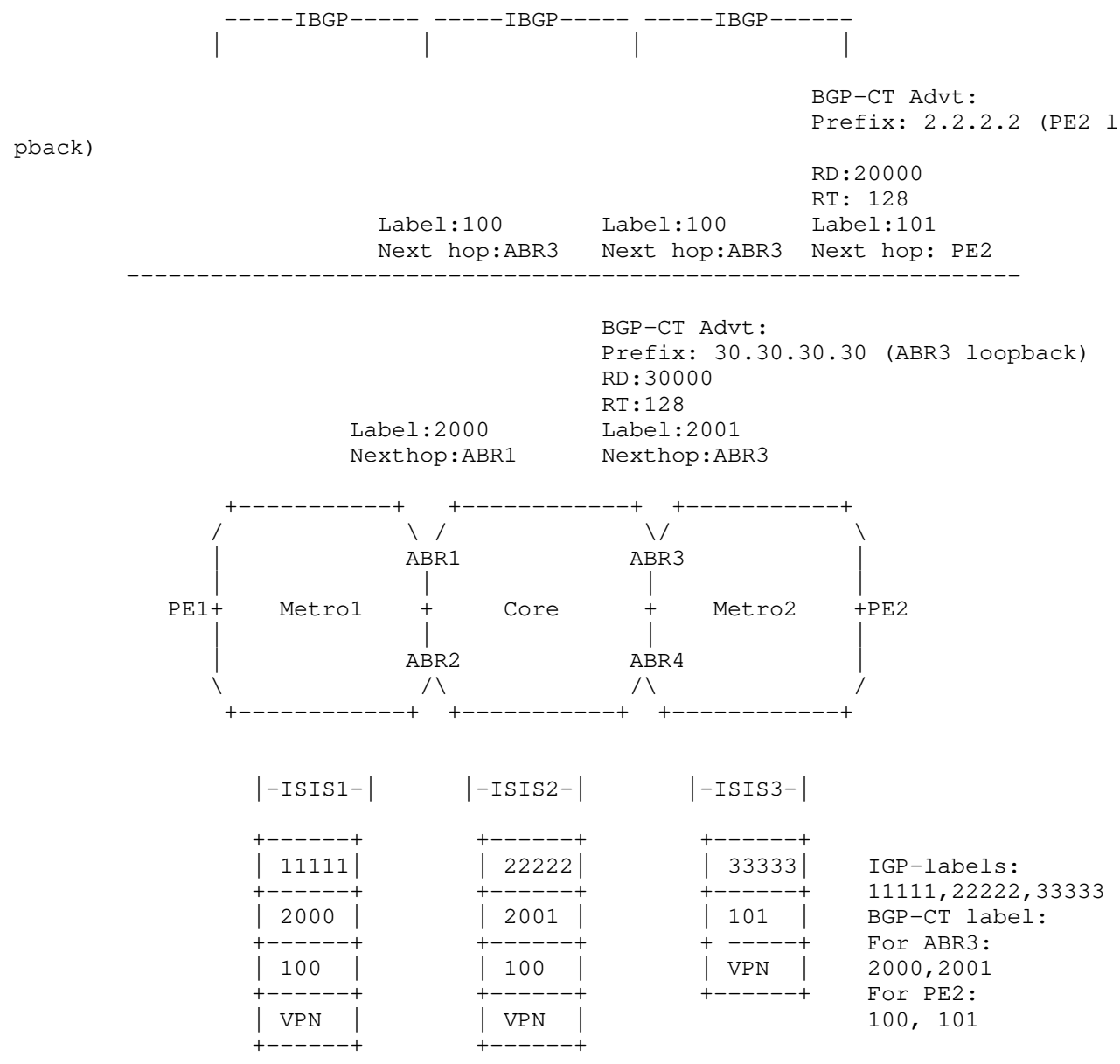


Figure 13: Recursive Route Resolution

The label resources are an important consideration in MPLS networks. On access devices, labels are consumed by services as well as for transport loopbacks inside IGP domain where the access device resides. For example, in the above diagram PE1 would have to allocate label resources equal to the number of customers connecting (i.e. the number of L2/L3 VPNs). Based on the size of the IGP domain that PE1 resides in, it will also have to allocate labels for IGP loopbacks. This number is at most a few thousands. So overall a typical access device should have adequate label resources in Seamless SR architecture. The P routers need to allocate labels for IGP loopbacks. This number again is small. At most it will be a few thousand based on number of nodes in the largest IGP domains. The metro networks connect to the core network through ABRs. It is possible that a given ABR may end up having to maintain forwarding entries for a large subset of the transport loopback routes. There may be a large number of metro networks connecting to a given ABR, and in this case, the ABR will need forwarding entries for every access node in the directly connected metros. So, this ABR may have to maintain on the order of 100k routes. With BGP-CT each Transport Class will have to be separately allocated a label. So, in the above example, the ABR1 would have to use 300k labels if there were 3 Transport Classes. This large number of label forwarding entries could be problematic.

In highly scaled scenarios, it is therefore desirable to reduce the forwarding state on the ABRs. This reduction can be achieved with label stacking as a result of recursive route resolution. Figure 13 illustrates how the forwarding state on ABRs can be greatly reduced by removing forward state for PEs in remote domains from the ABRs. In this example, we assume that we are setting up end-to-end paths for a single Transport Class, for example red. PE2 advertises a BGP-CT prefix of 2.2.2.2 with nexthop of 2.2.2.2 and label 101. 2.2.2.2 is PE2's loopback. ABR3 advertises label 100 for BGP-CT prefix 2.2.2.2 and changes the nexthop to self. When ABR1 receives the BGP-CT advertisement for 2.2.2.2, it does not change the nexthop and advertises same label advertised by ABR3. When PE1 receives the BGP-CT advertisement for 2.2.2.2 with a nexthop of ABR3, it resolves the route using reachability to ABR3.

The reachability of ABR3 has been learned by PE1 as the result of a BGP-CT advertisement originated by ABR3. As shown in Figure 13, ABR3 advertises BGP-CT prefix 30.30.30.30 with label 2001. ABR1 advertises label 2000 for BGP-CT prefix 30.30.30.30 and sets nexthop to self. PE1 constructs the service data packet with a VPN label at the bottom followed by 2 BGP-CT labels 100 and 2000. The top most label 2000 is the transport label for the metro1 domain. Removing the forwarding state for PEs in remote domains on the ABRs comes at the expense of one additional BGP-CT label on the data packet.

Recursive route resolution provides significant forwarding state reduction on the ABRs. ABRs have to allocate label resources only for the PEs in their local domain. The number of PEs in the same domain as a given ABR is much lower than the total number of PEs in the network.

The examples in this draft generally show VPN routes resolving on BGP-CT prefixes. However, the mechanisms are equally applicable to non-VPN routes.

5.11. Availability

Transport layer availability is very important in latency and loss sensitive networks. Any link or node failure must be repaired with 50ms convergence time. 50 ms convergence time can be achieved with Fast ReRoute (FRR) mechanisms. The seamless SR architecture provides protection against intra-domain link and node failures, Protection against border node failures and the egress link and node failures are also provided. Details of the FRR techniques are described in the sections below.

5.11.1. Intra domain link and node protection

In the seamless SR architecture, protection against node and link failure is achieved with the relevant FRR techniques for the corresponding transport mechanism used inside the domain. In the case of an IP fabric, ECMP FRR or LFA can be used. In SR networks, TI-LFA [I-D.ietf-rtgwg-segment-routing-ti-lfa] provides link and node protection. For SR-TE transport ([I-D.ietf-spring-segment-routing-policy]), link and node protection can be achieved using TI-LFA, combined with mechanisms described in [I-D.hegde-spring-node-protection-for-sr-te-paths].

5.11.2. Egress link and node protection

[RFC8679] describes the mechanisms for providing protection for border nodes and PE devices where services are hosted. The mechanism can be further simplified operationally with anycast SIDs and anycast service labels, as described in [I-D.hegde-rtgwg-egress-protection-sr-networks].

5.11.3. Border Node protection

Border node protection is very important in a network consisting of multiple domains. Seamless SR architecture can achieve 50ms FRR protection in the event of node failure using anycast addresses for the ABR/ASBRs. The requires that a set of ABRs advertise the same

label for a given BGP-CT Prefix. The detailed mechanism is described in [I-D.hegde-rtgwg-egress-protection-sr-networks].

5.12. Operations

5.12.1. MPLS ping and Traceroute

The Seamless SR Architecture consists of 3 layers: the service layer, intra-domain transport, and BGP-CT transport. Within each layer, connectivity can be verified independently. Within the BGP-CT transport layer, end-to-end connectivity can be verified using a new OAM FEC for BGP-CT defined in draft [I-D.kaliraj-idr-bgp-classful-transport-planes]. The draft describes end-to-end connectivity verification as well as fault isolation. BGP-CT verification happens only on the BGP nodes. The intra-domain connectivity verification and fault isolation will be based on the technology deployed in that domain as defined in [RFC8029] and [RFC8287].

5.12.2. Counters and Statistics

Traffic accounting and the ability to build demand matrix for PE to PE traffic is very important. With BGP-CT, per-label transit counters should be supported on every transit router. Per-label transit counters provide details of total traffic towards a remote PE measured at every BGP transit router. Per-label egress counters should be supported on ingress PE router. Per-label egress counters provide total traffic from ingress PE to the specific remote PE.

5.13. Service Mapping

Service mapping is an important aspect of any architecture. It provides means to translate end users SLA requirements into operator's network configurations. Seamless SR architecture supports automatic steering with extended color community. The Transport Class and the route target carried by the BGP-CT advertisement directly map to the extended color community. Services that require specific SLA carry the extended color community which maps to the Transport Class to which the BGP-CT advertisement belongs.

Other types of traffic steering such as DSCP based forwarding is expressed with mapping-community. Mapping community is a standard BGP community and is completely generic and user defined. The mapping community will have a specific service mapping feature associated with it along with required fallback behaviour when the primary transport goes down. The below list provides a general guideline into the different service mapping features and fallback options an implementation should provide.

DSCP based mapping with each DSCP mapping to a Transport Class.

DSCP based mapping with default mapping to a best-effort transport

DSCP based mapping with fallback to best-effort when primary transport tunnel goes down.

Extended color community based mapping with fallback to best effort

Fallback options with specific protocol during migrations

Fallback options to a different Transport Class.

No Fallback permitted.

5.14. Migrations

Networks that migrate from Seamless MPLS architecture to Seamless SR architecture, require that all the border nodes and PE devices be upgraded and enabled with new family on the BGP session. In cases where legacy nodes that cannot be upgraded, exporting from BGP-LU into BGP-CT and vice versa SHOULD be supported. Once the entire network is migrated to support BGP-CT, there is no need to run BGP-LU family on the BGP sessions. BGP-CT itself can advertise a best effort Transport Class and BGP-LU family can be removed.

5.15. Interworking with v6 transport technologies

A later version of this document will address interworking with other v6 technologies, including SRv6, SRm6, and MPLS over GRE6.

5.16. BGP based Multicast

BGP based multicast as described in draft [I-D.zzhang-bess-bgp-multicast] serves two main purposes. It can replace PIM/ mLDP inside a domain to natively do a BGP based multicast. It can also serve as an overlay stitching protocol to stitch multiple P2MP LSPs across the domain. This gives the ability to easily transition each domain independently from one technology to the other. BGP based multicast defines a new SAFI for carrying the MULTICAST TREE SAFI. Different route types are defined to support the various usecases.

6. Backward Compatibility

7. Security Considerations

TBD

8. IANA Considerations

9. Acknowledgements

Many thanks to Kireeti Kompella, Ron Bonica, Krzysztof Szarcowitz, Srihari Sangli,Julian Lucek, Ram Santhanakrishnan for discussions and inputs. Thanks to Joel Halpern for review and comments.

10. Contributors

1.Kaliraj Vairavakkalai

Juniper Networks

kaliraj@juniper.net

2. Jeffrey Zhang

Juniper Networks

zzhang@juniper.net

11. References

11.1. Normative References

[I-D.hegde-rtgwg-egress-protection-sr-networks]
Hegde, S. and W. Lin, "Egress Protection for Segment Routing (SR) networks", draft-hegde-rtgwg-egress-protection-sr-networks-00 (work in progress), March 2020.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-02 (work in progress), October 2019.

[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-01 (work in progress), July 2020.

[I-D.zzhang-bess-bgp-multicast]
Zhang, Z., Giuliano, L., Patel, K., Wijnands, I., mishra, m., and A. Gulko, "BGP Based Multicast", draft-zzhang-bess-bgp-multicast-03 (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>.

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

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

11.2. Informative References

[I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Node Protection for SR-TE Paths", draft-hegde-spring-node-protection-for-sr-te-paths-07 (work in progress), July 2020.

[I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-07 (work in progress), March 2018.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-19 (work in progress), September 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-11 (work in progress), September 2020.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-pce-segment-routing-policy-cp]
Koldychev, M., Sivabalan, S., Barth, C., Peng, S., and H. Bidgoli, "PCEP extension to support Segment Routing Policy Candidate Paths", draft-ietf-pce-segment-routing-policy-cp-00 (work in progress), June 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-04 (work in progress), August 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 2020.

[I-D.voyer-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (work in progress), July 2020.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

[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>.

[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>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <https://www.rfc-editor.org/info/rfc7510>.

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

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

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

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8679] Shen, Y., Jeganathan, M., Decraene, B., Gredler, H., Michel, C., and H. Chen, "MPLS Egress Protection Framework", RFC 8679, DOI 10.17487/RFC8679, December 2019, <https://www.rfc-editor.org/info/rfc8679>.

[TS.23.501-3GPP] 3rd Generation Partnership Project (3GPP), "System Architecture for 5G System; Stage 2, 3GPP TS 23.501 v16.4.0", March 2020.

Authors' Addresses

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

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks Inc.

Email: cbowers@juniper.net

Xiaohu Xu
Alibaba Inc.
Beijing
China

Email: xiaohu.xxh@alibaba-inc.com

Arkadiy Gulko
Refinitiv

Email: arkadiy.gulko@refinitiv.com

Alex Bogdanov
Google Inc.

Email: bogdanov@google.com

Jim Uttaro
ATT

Email: ju1738@att.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 16, 2020

C. Filsfils
S. Sivabalan, Ed.
Cisco Systems, Inc.
D. Voyer
Bell Canada
A. Bogdanov
Google, Inc.
P. Mattes
Microsoft

December 14, 2019

Segment Routing Policy Architecture
draft-ietf-spring-segment-routing-policy-06.txt

Abstract

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing. The headend node steers a flow into an SR Policy. The header of a packet steered in an SR Policy is augmented with an ordered list of segments associated with that SR Policy. This document details the concepts of SR Policy and steering into an SR Policy.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on June 16, 2020.

Table of Contents

1. Introduction

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing [RFC8402].

The headend node is said to steer a flow into an Segment Routing Policy (SR Policy).

The header of a packet steered into an SR Policy is augmented with an ordered list of segments associated with that SR Policy.

This document details the concepts of SR Policy and steering packets into an SR Policy. These apply equally to the MPLS and SRv6 instantiations of segment routing.

For reading simplicity, the illustrations are provided for the MPLS instantiations.

2. SR Policy

An SR Policy is a framework that enables instantiation of an ordered list of segments on a node for implementing a source routing policy with a specific intent for traffic steering from that node.

The Segment Routing architecture [RFC8402] specifies that any instruction can be bound to a segment. Thus, an SR Policy can be

built using any type of Segment Identifier (SID) including those associated with topological or service instructions.

This section defines the key aspects and constituents of an SR Policy.

2.1. Identification of an SR Policy

An SR Policy is identified through the tuple <headend, color, endpoint>. In the context of a specific headend, one may identify an SR policy by the <color, endpoint> tuple.

The headend is the node where the policy is instantiated/implemented. The headend is specified as an IPv4 or IPv6 address and is expected to be unique in the domain.

The endpoint indicates the destination of the policy. The endpoint is specified as an IPv4 or IPv6 address and is expected to be unique in the domain. In a specific case (refer to Section 8.8.1), the endpoint can be the null address (0.0.0.0 for IPv4, ::0 for IPv6).

The color is a 32-bit numerical value that associates the SR Policy with an intent (e.g. low-latency).

The endpoint and the color are used to automate the steering of service or transport routes on SR Policies (refer to Section 8).

An implementation MAY allow assignment of a symbolic name comprising of printable ASCII characters to an SR Policy to serve as a user-friendly attribute for debug and troubleshooting purposes. Such symbolic names may identify an SR Policy when the naming scheme ensures uniqueness.

2.2. Candidate Path and Segment List

An SR Policy is associated with one or more candidate paths. A candidate path is the unit for signaling of an SR Policy to a headend via protocols like Path Computation Element (PCE) Communication Protocol (PCEP) [RFC8281] or BGP SR Policy [I-D.ietf-idr-segment-routing-te-policy].

A Segment-List represents a specific source-routed path to send traffic from the headend to the endpoint of the corresponding SR policy.

A candidate path is either dynamic or explicit.

An explicit candidate path is expressed as a Segment-List or a set of Segment-Lists.

A dynamic candidate path expresses an optimization objective and a set of constraints. The headend (potentially with the help of a PCE) computes the solution Segment-List (or set of Segment-Lists) that solves the optimization problem.

If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with a weight for weighted load balancing (refer Section 2.11 for details). The default weight is 1.

2.3. Protocol-Origin of a Candidate Path

A headend may be informed about a candidate path for an SR Policy <color, endpoint> by various means including: via configuration, PCEP [RFC8281] or BGP [I-D.ietf-idr-segment-routing-te-policy].

Protocol-Origin of a candidate path is an 8-bit value which identifies the component or protocol that originates or signals the candidate path.

The table below specifies the RECOMMENDED default values:

Value	Protocol-Origin
10	PCEP
20	BGP SR Policy
30	Via Configuration

Table 1: Protocol-origin Identifier

Implementations MAY allow modifications of these default values assigned to protocols on the headend along similar lines as a routing administrative distance. Its application in the candidate path selection is described in Section 2.9.

2.4. Originator of a Candidate Path

Originator identifies the node which provisioned or signalled the candidate path on the headend. The originator is expressed in the form of a 160 bit numerical value formed by the concatenation of the fields of the tuple <ASN, node-address> as below:

- ASN : represented as a 4 byte number.

- Node Address : represented as a 128 bit value. IPv4 addresses are encoded in the lowest 32 bits.

Its application in the candidate path selection is described in Section 2.9.

When Protocol-Origin is Via Configuration, the ASN and node address MAY be set to either the headend or the provisioning controller/node ASN and address. Default value is 0 for both AS and node address.

When Protocol-Origin is PCEP, it is the IPv4 or IPv6 address of the PCE and the AS number SHOULD be set to 0 by default when not available or known.

Protocol-Origin is BGP SR Policy, it is provided by the BGP component on the headend and is:

- the BGP Router ID and ASN of the node/controller signalling the candidate path when it has a BGP session to the headend, OR
- the BGP Router ID of the eBGP peer signalling the candidate path along with ASN of origin when the signalling is done via one or more intermediate eBGP routers, OR
- the BGP Originator ID [RFC4456] and the ASN of the node/controller when the signalling is done via one or more route-reflectors over iBGP session.

2.5. Discriminator of a Candidate Path

The Discriminator is a 32 bit value associated with a candidate path that uniquely identifies it within the context of an SR Policy from a specific Protocol-Origin as specified below:

When Protocol-Origin is Via Configuration, this is an implementation's configuration model specific unique identifier for a candidate path. Default value is 0.

When PCEP is the Protocol-Origin, the method to uniquely identify signalled path will be specified in a future PCEP document. Default value is 0.

When BGP SR Policy is the Protocol-Origin, it is the distinguisher specified in Section 2.1 of [I-D.ietf-idr-segment-routing-te-policy].

Its application in the candidate path selection is described in Section 2.9.

2.6. Identification of a Candidate Path

A candidate path is identified in the context of a single SR Policy.

A candidate path is not shared across SR Policies.

A candidate path is not identified by its Segment-List(s).

> If CP1 is a candidate path of SR Policy Pol1 and CP2 is a candidate path of SR Policy Pol2, then these two candidate paths are independent, even if they happen to have the same Segment-List. The Segment-List does not identify a candidate path. The Segment-List is an attribute of a candidate path.

The identity of a candidate path MUST be uniquely established in the context of an SR Policy <headend, color, endpoint> in order to handle add, delete or modify operations on them in an unambiguous manner regardless of their source(s).

The tuple <Protocol-Origin, originator, discriminator> uniquely identifies a candidate path.

Candidate paths MAY also be assigned or signaled with a symbolic name comprising printable ASCII characters to serve as a user-friendly attribute for debug and troubleshooting purposes. Such symbolic names MUST NOT be considered as identifiers for a candidate path.

2.7. Preference of a Candidate Path

The preference of the candidate path is used to select the best candidate path for an SR Policy. The default preference is 100.

It is RECOMMENDED that each candidate path of a given SR policy has a different preference.

2.8. Validity of a Candidate Path

A candidate path is usable when it valid. A common path validity criterion is the reachability of its constituent SIDs. The validation rules are specified in Section 5.

2.9. Active Candidate Path

A candidate path is selected when it is valid and it is determined to be the best path of the SR Policy. The selected path is referred to as the "active path" of the SR policy in this document.

Whenever a new path is learned or an active path is deleted, the validity of an existing path changes or an existing path is changed, the selection process MUST be re-executed.

The candidate path selection process operates on the candidate path Preference. A candidate path is selected when it is valid and it has the highest preference value among all the candidate paths of the SR Policy.

In the case of multiple valid candidate paths of the same preference, the tie-breaking rules are evaluated on the identification tuple in the following order until only one valid best path is selected:

1. Higher value of Protocol-Origin is selected.

2. If specified by configuration, prefer the existing installed path.

3. Lower value of originator is selected.

4. Finally, the higher value of discriminator is selected.

The rules are framed with multiple protocols and sources in mind and hence may not follow the logic of a single protocol (e.g. BGP best path selection). The motivation behind these rules are as follows:

- The Protocol-Origin allows an operator to setup a default selection mechanism across protocol sources, e.g., to prefer configured over paths signalled via BGP SR Policy or PCEP.

- The preference, being the first tiebreaker, allows an operator to influence selection across paths thus allowing provisioning of multiple path options, e.g., CP1 is preferred and if it becomes invalid then fall-back to CP2 and so on. Since preference works across protocol sources it also enables (where necessary) selective override of the default protocol-origin preference, e.g., to prefer a path signalled via BGP SR Policy over what is configured.

- The originator allows an operator to have multiple redundant controllers and still maintain a deterministic behaviour over which of them are preferred even if they are providing the same candidate paths for the same SR policies to the headend.

- The discriminator performs the final tiebreaking step to ensure a deterministic outcome of selection regardless of the order in which candidate paths are signalled across multiple transport channels or sessions.

[I-D.filsfils-spring-sr-policy-considerations] provides a set of examples to illustrate the active candidate path selection rules.

2.10. Validity of an SR Policy

An SR Policy is valid when it has at least one valid candidate path.

2.11. Instantiation of an SR Policy in the Forwarding Plane

A valid SR Policy is instantiated in the forwarding plane.

Only the active candidate path SHOULD be used for forwarding traffic that is being steered onto that policy.

If a set of Segment-Lists is associated with the active path of the policy, then the steering is per flow and W-ECMP based according to the relative weight of each Segment-List.

The fraction of the flows associated with a given Segment-List is w/ Sw where w is the weight of the Segment-List and Sw is the sum of the weights of the Segment-Lists of the selected path of the SR Policy.

The accuracy of the weighted load-balancing depends on the platform implementation.

2.12. Priority of an SR Policy

Upon topological change, many policies could be recomputed or revalidated. An implementation MAY provide a per-policy priority configuration. The operator MAY set this field to indicate order in which the policies should be re-computed. Such a priority is represented by an integer in the range (0, 255) where the lowest value is the highest priority. The default value of priority is 128.

An SR Policy may comprise multiple Candidate Paths received from the same or different sources. A candidate path MAY be signaled with a priority value. When an SR Policy has multiple candidate paths with distinct signaled non-default priority values, the SR Policy as a whole takes the lowest value (i.e. the highest priority) amongst these signaled priority values.

2.13. Summary

In summary, the information model is the following:

```
          SR policy POL1 <headend, color, endpoint>
            Candidate-path CP1 <protocol-origin = 20, originator =
100:1.1.1.1, discriminator = 1>
```

```
              Preference 200
              Weight W1, SID-List1 <SID11...SID1i>
              Weight W2, SID-List2 <SID21...SID2j>
            Candidate-path CP2 <protocol-origin = 20, originator =
100:2.2.2.2, discriminator = 2>
              Preference 100
              Weight W3, SID-List3 <SID31...SID3i>
              Weight W4, SID-List4 <SID41...SID4j>
```

The SR Policy POL1 is identified by the tuple <headend, color, endpoint>. It has two candidate paths CP1 and CP2. Each is identified by a tuple <protocol-origin, originator, discriminator>. CP1 is the active candidate path (it is valid and it has the highest preference). The two Segment-Lists of CP1 are installed as the forwarding instantiation of SR policy Pol1. Traffic steered on Pol1 is flow-based hashed on Segment-List <SID11...SID1i> with a ratio W1/(W1+W2).

3. Segment Routing Database

An SR headend maintains the Segment Routing Database (SR-DB). The SR-DB is a conceptual database to illustrate the various pieces of information and their sources that may help in SR Policy computation and validation. There is no specific requirement for an implementation to create a new database as such.

An SR headend leverages the SR-DB to validate explicit candidate paths and compute dynamic candidate paths.

The information in the SR-DB MAY include:

- IGP information (topology, IGP metrics based on ISIS [RFC1195] and OSPF [RFC2328] [RFC5340])
- Segment Routing information (such as SRGB, SRLB, Prefix-SIDs, Adj-SIDs, BGP Peering SID, SRv6 SIDs) [RFC8402] [I-D.ietf-idr-bgpls-segment-routing-epe] [I-D.filsfils-spring-srv6-network-programming]
- TE Link Attributes (such as TE metric, SRLG, attribute-flag, extended admin group) [RFC5305] [RFC3630].
- Extended TE Link attributes (such as latency, loss) [RFC7810] [RFC7471]
- Inter-AS Topology information [I-D.ietf-idr-bgpls-segment-routing-epe].

The attached domain topology MAY be learned via IGP, BGP-LS or NETCONF.

A non-attached (remote) domain topology MAY be learned via BGP-LS or NETCONF.

In some use-cases, the SR-DB may only contain the attached domain topology while in others, the SR-DB may contain the topology of multiple domains and in this case it is multi-domain capable.

The SR-DB MAY also contain the SR Policies instantiated in the network. This can be collected via BGP-LS [I-D.ietf-idr-te-lsp-distribution] or PCEP [RFC8231] and [I-D.sivabalan-pce-binding-label-sid]. This information allows to build an end-to-end policy on the basis of intermediate SR policies (see Section 6 for further details).

The SR-DB MAY also contain the Maximum SID Depth (MSD) capability of nodes in the topology. This can be collected via ISIS [I-D.ietf-isis-segment-routing-msd], OSPF [I-D.ietf-ospf-segment-routing-msd], BGP-LS [I-D.ietf-idr-bgp-ls-segment-routing-msd] or PCEP [I-D.ietf-pce-segment-routing].

The use of the SR-DB for computation and validation of SR Policies is outside the scope of this document. Some implementation aspects related to this are covered in [I-D.filsfils-spring-sr-policy-considerations].

4. Segment Types

A Segment-List is an ordered set of segments represented as <S1, S2, ... Sn> where S1 is the first segment.

Based on the desired dataplane, either the MPLS label stack or the SRv6 SRH is built from the Segment-List. However, the Segment-List itself can be specified using different segment-descriptor types and the following are currently defined:

Type A: SR-MPLS Label:
: A MPLS label corresponding to any of the segment types defined for SR-MPLS (as defined in [RFC8402] or other SR-MPLS specifications) can be used. Additionally, reserved labels like explicit-null or in general any MPLS label may also be used. E.g. this type can be used to specify a label representation which maps to an optical transport path on a packet transport node. This type does not require the headend to perform SID resolution.

Type B: SRv6 SID:

An IPv6 address corresponding to any of the segment types defined for SRv6 (as defined in [I-D.filsfils-spring-srv6-network-programming] or other SRv6 specifications) can be used. This type does not require the headend to perform SID resolution.

Type C: IPv4 Prefix with optional SR Algorithm:
The headend is required to resolve the specified IPv4 Prefix Address to the SR-MPLS label corresponding to a Prefix SID segment (as defined in [RFC8402]). The SR algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type D: IPv6 Global Prefix with optional SR Algorithm for SR-MPLS:
In this case the headend is required to resolve the specified IPv6 Global Prefix Address to the SR-MPLS label corresponding to its Prefix SID segment (as defined in [RFC8402]). The SR Algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type E: IPv4 Prefix with Local Interface ID:
This type allows identification of Adjacency SID (as defined in [RFC8402]) or BGP EPE Peering SID (as defined in [I-D.ietf-idr-bgpls-segment-routing-epe]) label for point-to-point links including IP unnumbered links. The headend is required to resolve the specified IPv4 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. The Local Interface ID link descriptor follows semantics as specified in [RFC7752]. This type can also be used to indicate indirection into a layer 2 interface (i.e. without IP address) like a representation of an optical transport path or a layer 2 Ethernet port or circuit at the specified node.

Type F: IPv4 Addresses for link endpoints as Local, Remote pair:
This type allows identification of Adjacency SID (as defined in [RFC8402]) or BGP EPE Peering SID (as defined in [I-D.ietf-idr-bgpls-segment-routing-epe]) label for links. The headend is required to resolve the specified IPv4 Local Address to the Node originating it and then use the IPv4 Remote Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

Type G: IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SR-MPLS:
This type allows identification of Adjacency SID (as defined in [RFC8402]) or BGP EPE Peering SID (as defined in

[I-D.ietf-idr-bgpls-segment-routing-epe]) label for links including those with only Link Local IPv6 addresses. The headend is required to resolve the specified IPv6 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. For other than point-to-point links, additionally the specific adjacency over the link needs to be resolved using the Remote Prefix and Interface ID. The Local and Remote pair of Prefix and Interface ID link descriptor follows semantics as specified in [RFC7752]. This type can also be used to indicate indirection into a layer 2 interface (i.e. without IP address) like a representation of an optical transport path or a layer 2 Ethernet port or circuit at the specified node.

Type H: IPv6 Addresses for link endpoints as Local, Remote pair for SR-MPLS:

This type allows identification of Adjacency SID (as defined in [RFC8402]) or BGP EPE Peering SID (as defined in [I-D.ietf-idr-bgpls-segment-routing-epe]) label for links with Global IPv6 addresses. The headend is required to resolve the specified Local IPv6 Address to the Node originating it and then use the Remote IPv6 Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

Type I: IPv6 Global Prefix with optional SR Algorithm for SRv6:

The headend is required to resolve the specified IPv6 Global Prefix Address to the SRv6 END function SID (as defined in [I-D.filsfils-spring-srv6-network-programming]) corresponding to the node which is originating the prefix. The SR Algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type J: IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SRv6:

This type allows identification of SRv6 END.X SID (as defined in [I-D.filsfils-spring-srv6-network-programming]) for links with only Link Local IPv6 addresses. The headend is required to resolve the specified IPv6 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. For other than point-to-point links, additionally the specific adjacency needs to be resolved using the Remote Prefix and Interface ID. The Local and Remote pair of Prefix and Interface ID link descriptor follows semantics as specified in [RFC7752].

Type K: IPv6 Addresses for link endpoints as Local, Remote pair for SRv6:

This type allows identification of SRv6 END.X SID (as defined in [I-D.filsfils-spring-srv6-network-programming]) for links with Global IPv6 addresses. The headend is required to resolve the specified Local IPv6 Address to the Node originating it and then use the Remote IPv6 Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

When the algorithm is not specified for the SID types above which optionally allow for it, the headend SHOULD use the Strict Shortest Path algorithm if available; otherwise it SHOULD use the default Shortest Path algorithm. The specification of algorithm enables the use of IGP Flex Algorithm [I-D.ietf-lsr-flex-algo] specific SIDs in SR Policy.

For SID types C-through-K, a SID value may also be optionally provided to the headend for verification purposes. Section 5.1. describes the resolution and verification of the SIDs and Segment Lists on the headend.

When building the MPLS label stack or the IPv6 Segment list from the Segment List, the node instantiating the policy MUST interpret the set of Segments as follows:

- The first Segment represents the topmost label or the first IPv6 segment. It identifies the active segment the traffic will be directed toward along the explicit SR path.
- The last Segment represents the bottommost label or the last IPv6 segment the traffic will be directed toward along the explicit SR path.

4.1. Explicit Null

A Type A SID may be any MPLS label, including reserved labels.

For example, assuming that the desired traffic-engineered path from a headend 1 to an endpoint 4 can be expressed by the Segment-List <16002, 16003, 16004> where 16002, 16003 and 16004 respectively refer to the IPv4 Prefix SIDs bound to node 2, 3 and 4, then IPv6 traffic can be traffic-engineered from nodes 1 to 4 via the previously described path using an SR Policy with Segment-List <16002, 16003, 16004, 2> where mpls label value of 2 represents the "IPv6 Explicit NULL Label".

The penultimate node before node 4 will pop 16004 and will forward the frame on its directly connected interface to node 4.

The endpoint receives the traffic with top label "2" which indicates that the payload is an IPv6 packet.

When steering unlabeled IPv6 BGP destination traffic using an SR policy composed of Segment-List(s) based on IPv4 SIDs, the Explicit Null Label Policy is processed as specified in [I-D.ietf-idr-segment-routing-te-policy]) Section 2.4.4. When an "IPv6 Explicit NULL label" is not present as the bottom label, the headend SHOULD automatically impose one. Refer to Section 8 for more details.

5. Validity of a Candidate Path

5.1. Explicit Candidate Path

An explicit candidate path is associated with a Segment-List or a set of Segment-Lists.

An explicit candidate path is provisioned by the operator directly or via a controller.

The computation/logic that leads to the choice of the Segment-List is external to the SR Policy headend. The SR Policy headend does not compute the Segment-List. The SR Policy headend only confirms its validity.

A Segment-List of an explicit candidate path MUST be declared invalid when:

- It is empty.
- Its weight is 0.
- The headend is unable to perform path resolution for the first SID into one or more outgoing interface(s) and next-hop(s).
- The headend is unable to perform SID resolution for any non-first SID of type C-through-K into an MPLS label or an SRv6 SID.
- The headend verification fails for any SID for which verification has been explicitly requested.

"Unable to perform path resolution" means that the headend has no path to the SID in its SR database.

SID verification is performed when the headend is explicitly requested to verify SID(s) by the controller via the signaling protocol used. Implementations MAY provide a local configuration

option to enable verification on a global or per policy or per candidate path basis.

"Verification fails" for a SID means any of the following:

- The headend is unable to find the SID in its SR DB
- The headend detects mis-match between the SID value and its context provided for SIDs of type C-through-K in its SR DB.
- The headend is unable to perform SID resolution for any non-first SID of type C-through-K into an MPLS label or an SRv6 SID.

In multi-domain deployments, it is expected that the headend be unable to verify the reachability of the SIDs in remote domains. Types A or B MUST be used for the SIDs for which the reachability cannot be verified. Note that the first SID MUST always be reachable regardless of its type.

In addition, a Segment-List MAY be declared invalid when:

- Its last segment is not a Prefix SID (including BGP Peer Node-SID) advertised by the node specified as the endpoint of the corresponding SR policy.
- Its last segment is not an Adjacency SID (including BGP Peer Adjacency SID) of any of the links present on neighbor nodes and that terminate on the node specified as the endpoint of the corresponding SR policy.

An explicit candidate path is invalid as soon as it has no valid Segment-List.

5.2. Dynamic Candidate Path

A dynamic candidate path is specified as an optimization objective and constraints.

The headend of the policy leverages its SR database to compute a Segment-List ("solution Segment-List") that solves this optimization problem.

The headend re-computes the solution Segment-List any time the inputs to the problem change (e.g., topology changes).

When local computation is not possible (e.g., a policy's tailend is outside the topology known to the headend) or not desired, the headend MAY send path computation request to a PCE supporting PCEP extension specified in [I-D.ietf-pce-segment-routing].

If no solution is found to the optimization objective and constraints, then the dynamic candidate path MUST be declared invalid.

[I-D.filsfils-spring-sr-policy-considerations] discusses some of the optimization objectives and constraints that may be considered by a dynamic candidate path. It illustrates some of the desirable properties of the computation of the solution Segment-List.

6. Binding SID

The Binding SID (BSID) is fundamental to Segment Routing [RFC8402]. It provides scaling, network opacity and service independence. [I-D.filsfils-spring-sr-policy-considerations] illustrates some of these benefits. This section describes the association of BSID with an SR Policy.

6.1. BSID of a candidate path

Each candidate path MAY be defined with a BSID.

Candidate Paths of the same SR policy SHOULD have the same BSID.

Candidate Paths of different SR policies MUST NOT have the same BSID.

6.2. BSID of an SR Policy

The BSID of an SR Policy is the BSID of its active candidate path.

When the active candidate path has a specified BSID, the SR Policy uses that BSID if this value (label in MPLS, IPv6 address in SRv6) is available (i.e., not associated with any other usage: e.g. to another MPLS client, to another SID, to another SR Policy).

Optionally, instead of only checking that the BSID of the active path is available, a headend MAY check that it is available within a given SID range i.e., Segment Routing Local Block (SRLB) as specified in [RFC8402].

When the specified BSID is not available (optionally is not in the SRLB), an alert message MUST be generated.

In the cases (as described above) where SR Policy does not have a BSID available, then the SR Policy MAY dynamically bind a BSID to itself. Dynamically bound BSID SHOULD use an available SID outside the SRLB.

Assuming that at time t the BSID of the SR Policy is B1, if at time t+dt a different candidate path becomes active and this new active path does not have a specified BSID or its BSID is specified but is not available (e.g. it is in use by something else), then the SR Policy keeps the previous BSID B1.

The association of an SR Policy with a BSID thus MAY change over the life of the SR Policy (e.g., upon active path change). Hence, the BSID SHOULD NOT be used as an identification of an SR Policy.

6.2.1. Frequent use-case : unspecified BSID

All the candidate paths of the same SR Policy can have an unspecified BSID.

In such a case, a BSID MAY be dynamically bound to the SR Policy as soon as the first valid candidate path is received. That BSID is kept along all the life of the SR Policy and across changes of active candidate path.

6.2.2. Frequent use-case: all specified to the same BSID

All the paths of the SR Policy can have the same specified BSID.

6.2.3. Specified-BSID-only

An implementation MAY support the configuration of the Specified-BSID-only restrictive behavior on the headend for all SR Policies or individual SR Policies. Further, this restrictive behavior MAY also be signaled on a per SR Policy basis to the headend.

When this restrictive behavior is enabled, if the candidate path has an unspecified BSID or if the specified BSID is not available when the candidate path becomes active then no BSID is bound to it and it is considered invalid. An alert MUST be triggered for this error. Other candidate paths MUST then be evaluated for becoming the active candidate path.

6.3. Forwarding Plane

A valid SR Policy installs a BSID-keyed entry in the forwarding plane with the action of steering the packets matching this entry to the selected path of the SR Policy.

If the Specified-BSID-only restrictive behavior is enabled and the BSID of the active path is not available (optionally not in the SRLB), then the SR Policy does not install any entry indexed by a BSID in the forwarding plane.

6.4. Non-SR usage of Binding SID

An implementation MAY choose to associate a Binding SID with any type of interface (e.g. a layer 3 termination of an Optical Circuit) or a tunnel (e.g. IP tunnel, GRE tunnel, IP/UDP tunnel, MPLS RSVP-TE tunnel, etc). This enables the use of other non-SR enabled interfaces and tunnels as segments in an SR Policy Segment-List without the need of forming routing protocol adjacencies over them.

The details of this kind of usage are beyond the scope of this document. A specific packet optical integration use case is described in [I-D.anand-spring-poi-sr]

7. SR Policy State

The SR Policy State is maintained on the headend to represent the state of the policy and its candidate paths. This is to provide an accurate representation of whether the SR Policy is being instantiated in the forwarding plane and which of its candidate paths and segment-list(s) are active. The SR Policy state MUST also reflect the reason when a policy and/or its candidate path is not active due to validation errors or not being preferred.

The SR Policy state can be reported by the headend node via BGP-LS [I-D.ietf-idr-te-lsp-distribution] or PCEP [RFC8231] and [I-D.sivabalan-pce-binding-label-sid].

SR Policy state on the headend also includes traffic accounting information for the flows being steered via the policies. The details of the SR Policy accounting are beyond the scope of this document. The aspects related to the SR traffic counters and their usage in the broader context of traffic accounting in a SR network are covered in [I-D.filsfils-spring-sr-traffic-counters] and [I-D.ali-spring-sr-traffic-accounting] respectively.

Implementations MAY support an administrative state to control locally provisioned policies via mechanisms like CLI or NETCONF.

8. Steering into an SR Policy

A headend can steer a packet flow into a valid SR Policy in various ways:

- Incoming packets have an active SID matching a local BSID at the headend.
- Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.

- Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

For simplicity of illustration, this document uses the SR-MPLS example.

8.1. Validity of an SR Policy

An SR Policy is invalid when all its candidate paths are invalid as described in Section 5 and Section 2.10.

By default, upon transitioning to the invalid state,

- an SR Policy and its BSID are removed from the forwarding plane.
- any steering of a service (PW), destination (BGP-VPN), flow or packet on the related SR policy is disabled and the related service, destination, flow or packet is routed per the classic forwarding table (e.g. longest-match to the destination or the recursing next-hop).

8.2. Drop upon invalid SR Policy

An SR Policy MAY be enabled for the Drop-Upon-Invalid behavior:

- an invalid SR Policy and its BSID is kept in the forwarding plane with an action to drop.
- any steering of a service (PW), destination (BGP-VPN), flow or packet on the related SR policy is maintained with the action to drop all of this traffic.

The drop-upon-invalid behavior has been deployed in use-cases where the operator wants some PW to only be transported on a path with specific constraints. When these constraints are no longer met, the operator wants the PW traffic to be dropped. Specifically, the operator does not want the PW to be routed according to the IGP shortest-path to the PW endpoint.

8.3. Incoming Active SID is a BSID

Let us assume that headend H has a valid SR Policy P of Segment-List <S1, S2, S3> and BSID B.

When H receives a packet K with label stack <B, L2, L3>, H pops B and pushes <S1, S2, S3> and forwards the resulting packet according to SID S1.

"Forwarding the resulting packet according to S1" means: If S1 is an Adj SID or a PHP-enabled prefix SID advertised by a neighbor, H sends the resulting packet with label stack <S2, S3, L2, L3> on the outgoing interface associated with S1; Else H sends the resulting packet with label stack <S1, S2, S3, L2, L3> along the path of S1.

H has steered the packet into the SR policy P.

H did not have to classify the packet. The classification was done by a node upstream of H (e.g., the source of the packet or an intermediate ingress edge node of the SR domain) and the result of this classification was efficiently encoded in the packet header as a BSID.

This is another key benefit of the segment routing in general and the binding SID in particular: the ability to encode a classification and the resulting steering in the packet header to better scale and simplify intermediate aggregation nodes.

If the SR Policy P is invalid, the BSID B is not in the forwarding plane and hence the packet K is dropped by H.

8.4. Per-Destination Steering

Let us assume that headend H:

- learns a BGP route R/r via next-hop N, extended-color community C and VPN label V.
- has a valid SR Policy P to (color = C, endpoint = N) of Segment-List <S1, S2, S3> and BSID B.
- has a BGP policy which matches on the extended-color community C and allows its usage as SLA steering information.

If all these conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P of BSID B instead of via N.

Indeed, H's local BGP policy and the received BGP route indicate that the headend should associate R/r with an SR Policy path to endpoint N with the SLA associated with color C. The headend therefore installs the BGP route on that policy.

This can be implemented by using the BSID as a generalized next-hop and installing the BGP route on that generalized next-hop.

When H receives a packet K with a destination matching R/r, H pushes the label stack <S1, S2, S3, V> and sends the resulting packet along the path to S1.

Note that any SID associated with the BGP route is inserted after the Segment-List of the SR Policy (i.e., <S1, S2, S3, V>).

The same behavior is applicable to any type of service route: any AFI/SAFI of BGP [RFC4760] any AFI/SAFI of LISP [RFC6830].

8.4.1. Multiple Colors

When a BGP route has multiple extended-color communities each with a valid SR Policy NLRI, the BGP process installs the route on the SR policy whose color is of highest numerical value.

Let us assume that headend H:

- learns a BGP route R/r via next-hop N, extended-color communities C1 and C2 and VPN label V.
- has a valid SR Policy P1 to (color = C1, endpoint = N) of Segment-List <S1, S2, S3> and BSID B1.
- has a valid SR Policy P2 to (color = C2, endpoint = N) of Segment-List <S4, S5, S6> and BSID B2.
- has a BGP policy which matches on the extended-color communities C1 and C2 and allows their usage as SLA steering information

If all these conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P2 of BSID=B2 (instead of N) because C2 > C1.

8.5. Recursion on an on-demand dynamic BSID

In the previous section, it was assumed that H had a pre-established "explicit" SR Policy (color C, endpoint N).

In this section, independently to the a-priori existence of any explicit candidate path of the SR policy (C, N), it is to be noted that the BGP process at headend node H triggers the instantiation of a dynamic candidate path for the SR policy (C, N) as soon as:

- the BGP process learns of a route R/r via N and with color C.
- a local policy at node H authorizes the on-demand SR Policy path instantiation and maps the color to a dynamic SR Policy path optimization template.

8.5.1. Multiple Colors

When a BGP route R/r via N has multiple extended-color communities Ci (with i=1 ... n), an individual on-demand SR Policy dynamic path request (color Ci, endpoint N) is triggered for each color Ci.

8.6. Per-Flow Steering

Let us assume that headend H:

- has a valid SR Policy P1 to (color = C1, endpoint = N) of Segment-List <S1, S2, S3> and BSID B1.
- has a valid SR Policy P2 to (color = C2, endpoint = N) of Segment-List <S4, S5, S6> and BSID B2.
- is configured to instantiate an array of paths to N where the entry 0 is the IGP path to N, color C1 is the first entry and Color C2 is the second entry. The index into the array is called a Forwarding Class (FC). The index can have values 0 to 7.
- is configured to match flows in its ingress interfaces (upon any field such as Ethernet destination/source/vlan/tos or IP destination/source/DSCP or transport ports etc.) and color them with an internal per-packet forwarding-class variable (0, 1 or 2 in this example).

If all these conditions are met, H installs in RIB/FIB:

- N via a recursion on an array A (instead of the immediate outgoing link associated with the IGP shortest-path to N).
- Entry A(0) set to the immediate outgoing link of the IGP shortest-path to N.
- Entry A(1) set to SR Policy P1 of BSID=B1.
- Entry A(2) set to SR Policy P2 of BSID=B2.

H receives three packets K, K1 and K2 on its incoming interface. These three packets either longest-match on N or more likely on a BGP/service route which recurses on N. H colors these 3 packets respectively with forwarding-class 0, 1 and 2. As a result:

- H forwards K along the shortest-path to N (which in SR-MPLS results in the pushing of the prefix-SID of N).
- H pushes <S1, S2, S3> on packet K1 and forwards the resulting frame along the shortest-path to S1.
- H pushes <S4, S5, S6> on packet K2 and forwards the resulting frame along the shortest-path to S4.

If the local configuration does not specify any explicit forwarding information for an entry of the array, then this entry is filled with the same information as entry 0 (i.e. the IGP shortest-path).

If the SR Policy mapped to an entry of the array becomes invalid, then this entry is filled with the same information as entry 0. When all the array entries have the same information as entry0, the forwarding entry for N is updated to bypass the array and point directly to its outgoing interface and next-hop.

The array index values (e.g. 0, 1 and 2) and the notion of forwarding-class are implementation specific and only meant to describe the desired behavior. The same can be realized by other mechanisms.

This realizes per-flow steering: different flows bound to the same BGP endpoint are steered on different IGP or SR Policy paths.

A headend MAY support options to apply per-flow steering only for traffic matching specific prefixes (e.g. specific IGP or BGP prefixes).

8.7. Policy-based Routing

Finally, headend H may be configured with a local routing policy which overrides any BGP/IGP path and steer a specified packet on an SR Policy. This includes the use of mechanisms like IGP Shortcut for automatic routing of IGP prefixes over SR Policies intended for such purpose.

8.8. Optional Steering Modes for BGP Destinations

8.8.1. Color-Only BGP Destination Steering

In the previous section, it is seen that the steering on an SR Policy is governed by the matching of the BGP route's next-hop N and the authorized color C with an SR Policy defined by the tuple (N, C).

This is the most likely form of BGP destination steering and the one recommended for most use-cases.

This section defines an alternative steering mechanism based only on the color.

This color-only steering variation is governed by two new flags "C" and "O" defined in the color extended community [ref draft-ietf-idr-segment-routing-te-policy section 3].

The Color-Only flags "CO" are set to 00 by default.

When 00, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is the IPv4 or IPv6

            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

This is the classic case described in this document previously and what is recommended in most scenarios.

When 01, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is the IPv4 or IPv6

            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE IF there is a valid SR Policy (null endpoint, C) of the
            same address-family of N;
        Steer into SR Policy (null endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family null endpoint, C);
        Steer into SR Policy (any null endpoint, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

When 10, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is an IPv4 or IPv6
            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE IF there is a valid SR Policy (null endpoint, C)
            of the same address-family of N;
        Steer into SR Policy (null endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family null endpoint, C);
        Steer into SR Policy (any null endpoint, C);
    ELSE IF there is any valid SR Policy (any endpoint, C)
            of the same address-family of N;
        Steer into SR Policy (any endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family endpoint, C);
        Steer into SR Policy (any address-family endpoint, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

The null endpoint is 0.0.0.0 for IPv4 and ::0 for IPv6 (all bits set to the 0 value).

The value 11 is reserved for future use and SHOULD NOT be used. Upon reception, an implementations MUST treat it like 00.

8.8.2. Multiple Colors and CO flags

The steering preference is first based on highest color value and then CO-dependent for the color. Assuming a Prefix via (NH, C1(CO=01), C2(CO=01)); C1>C2 The steering preference order is:

- SR policy (NH, C1).
- SR policy (null, C1).
- SR policy (NH, C2).
- SR policy (null, C2).
- IGP to NH.

8.8.3. Drop upon Invalid

This document defined earlier that when all the following conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P of BSID B instead of via N.

- H learns a BGP route R/r via next-hop N, extended-color community C and VPN label V.
- H has a valid SR Policy P to (color = C, endpoint = N) of Segment-List <S1, S2, S3> and BSID B.
- H has a BGP policy which matches on the extended-color community C and allows its usage as SLA steering information.

This behavior is extended by noting that the BGP policy may require the BGP steering to always stay on the SR policy whatever its validity.

This is the "drop upon invalid" option described in Section 8.2 applied to BGP-based steering.

9. Protection

9.1. Leveraging TI-LFA local protection of the constituent IGP segments

In any topology, Topology-Independent Loop Free Alternate (TI-LFA) [I-D.bashandy-rtgwg-segment-routing-ti-lfa] provides a 50msec local protection technique for IGP SIDs. The backup path is computed on a per IGP SID basis along the post-convergence path.

In a network that has deployed TI-LFA, an SR Policy built on the basis of TI-LFA protected IGP segments leverages the local protection of the constituent segments.

In a network that has deployed TI-LFA, an SR Policy instantiated only with non-protected Adj SIDs does not benefit from any local protection.

9.2. Using an SR Policy to locally protect a link

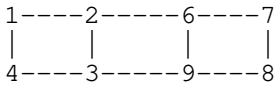


Figure 1: Local protection using SR Policy

An SR Policy can be instantiated at node 2 to protect the link 2to6. A typical explicit Segment-List would be <3, 9, 6>.

A typical use-case occurs for links outside an IGP domain: e.g. 1, 2, 3 and 4 are part of IGP/SR sub-domain 1 while 6, 7, 8 and 9 are part of IGP/SR sub-domain 2. In such a case, links 2to6 and 3to9 cannot benefit from TI-LFA automated local protection. The SR Policy with Segment-List <3, 9, 6> on node 2 can be locally configured to be a fast-reroute backup path for the link 2to6.

9.3. Using a Candidate Path for Path Protection

An SR Policy allows for multiple candidate paths, of which at any point in time there is a single active candidate path that is provisioned in the forwarding plane and used for traffic steering. However, another (lower preference) candidate path MAY be designated as the backup for a specific or all (active) candidate path(s). The following options are possible:

- A pair of disjoint candidate paths are provisioned with one of them as primary and the other is identified as its backup.
- A specific candidate path is provisioned as the backup for any (active) candidate path.
- The headend picks the next (lower) preference valid candidate path as the backup for the active candidate path.

The headend MAY compute a-priori and validate such backup candidate paths as well as provision them into forwarding plane as backup for the active path. A fast re-route mechanism MAY then be used to trigger sub 50msec switchover from the active to the backup candidate path in the forwarding plane. Mechanisms like BFD MAY be used for fast detection of such failures.

10. Security Considerations

This document does not define any new protocol extensions and does not impose any additional security challenges.

11. IANA Considerations

This document requests IANA to create a new top-level registry called "Segment Routing Parameters". This registry is being defined to serve as a top-level registry for keeping all other Segment Routing sub-registries.

The document also requests creation of a new sub-registry called "Segment Types" to be defined under the top-level "Segment Routing Parameters" registry. This sub-registry maintains the alphabetic identifiers for the segment types (as specified in section 4) that may be used within a Segment List of an SR Policy. This sub-registry would follow the Specification Required allocation policy as specified in [RFC8126].

The initial registrations for this sub-registry are as follows:

Value	Description	Reference
A	SR-MPLS Label	[This.ID]
B	SRv6 SID	[This.ID]
C	IPv4 Prefix with optional SR Algorithm	[This.ID]
D	IPv6 Global Prefix with optional SR Algorithm for SR-MPLS	[This.ID]
E	IPv4 Prefix with Local Interface ID	[This.ID]
F	IPv4 Addresses for link endpoints as Local, Remote pair	[This.ID]
G	IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SR-MPLS	[This.ID]
H	IPv6 Addresses for link endpoints as Local, Remote pair for SR-MPLS	[This.ID]
I	IPv6 Global Prefix with optional SR Algorithm for SRv6	[This.ID]
J	IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SRv6	[This.ID]
K	IPv6 Addresses for link endpoints as Local, Remote pair for SRv6	[This.ID]

Table 2: Initial IANA Registration

11.1. Guidance for Designated Experts

The Designated Expert (DE) is expected to ascertain the existence of suitable documentation (a specification) as described in [RFC8126] and to verify that the document is permanently and publicly

available. The DE is also expected to check the clarity of purpose and use of the requested assignment. Additionally, the DE must verify that any request for one of these assignments has been made available for review and comment within the IETF: the DE will post the request to the SPRING Working Group mailing list (or a successor mailing list designated by the IESG). If the request comes from within the IETF, it should be documented in an Internet-Draft. Lastly, the DE must ensure that any other request for a code point does not conflict with work that is active or already published within the IETF.

12. Acknowledgement

The authors would like to thank Tarek Saad, Dhanendra Jain, Ruediger Geib and Rob Shakir for their valuable comments and suggestions.

13. Contributors

The following people have contributed to this document:

Ketan Talaulikar
Cisco Systems
Email: ketant@cisco.com

Zafar Ali
Cisco Systems
Email: zali@cisco.com

Jose Liste
Cisco Systems
Email: jliste@cisco.com

Francois Clad
Cisco Systems
Email: fclad@cisco.com

Kamran Raza
Cisco Systems
Email: skraza@cisco.com

Shraddha Hegde
Juniper Networks
Email: shraddha@juniper.net

Steven Lin
Google, Inc.
Email: stevenlin@google.com

Przemyslaw Krol
Google, Inc.
Email: pkrol@google.com

Martin Horneffer
Deutsche Telekom
Email: martin.horneffer@telekom.de

Dirk Steinberg
Steinberg Consulting
Email: dws@steinbergnet.net

Bruno Decraene
Orange Business Services
Email: bruno.decraene@orange.com

Stephane Litkowski
Orange Business Services
Email: stephane.litkowski@orange.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

14. References

14.1. Normative References

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

14.2. Informative References

[I-D.ali-spring-sr-traffic-accounting]
Ali, Z., Filsfils, C., Talaulikar, K., Sivabalan, S., Horneffer, M., Raszuk, R., Litkowski, S., and D. Voyer, "Traffic Accounting in Segment Routing Networks", draft-ali-spring-sr-traffic-accounting-03 (work in progress), August 2019.

[I-D.anand-spring-poi-sr]
Anand, M., Bardhan, S., Subrahmaniam, R., Tantsura, J., Mukhopadhyaya, U., and C. Filsfils, "Packet-Optical Integration in Segment Routing", draft-anand-spring-poi-sr-08 (work in progress), July 2019.

[I-D.bashandy-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-05 (work in progress), October 2018.

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-04 (work in progress), October 2019.

[I-D.filsfils-spring-sr-traffic-counters]
Filsfils, C., Ali, Z., Horneffer, M., daniel.voyer@bell.ca, d., Durrani, M., and R. Raszuk, "Segment Routing Traffic Accounting Counters", draft-filsfils-spring-sr-traffic-counters-00 (work in progress), June 2018.

[I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-07 (work in progress), February 2019.

[I-D.ietf-idr-bgp-ls-segment-routing-msd]
Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G., and N. Triantafillis, "Signaling MSD (Maximum SID Depth) using Border Gateway Protocol Link-State", draft-ietf-idr-bgp-ls-segment-routing-msd-09 (work in progress), October 2019.

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

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-08 (work in progress), November 2019.

[I-D.ietf-idr-te-lsp-distribution]
Previdi, S., Talaulikar, K., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", draft-ietf-idr-te-lsp-distribution-12 (work in progress), October 2019.

[I-D.ietf-isis-segment-routing-msd]
Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling MSD (Maximum SID Depth) using IS-IS", draft-ietf-isis-segment-routing-msd-19 (work in progress), October 2018.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-05 (work in progress), November 2019.

[I-D.ietf-ospf-segment-routing-msd]
Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling MSD (Maximum SID Depth) using OSPF", draft-ietf-ospf-segment-routing-msd-25 (work in progress), October 2018.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-16 (work in progress), March 2019.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-07 (work in progress), July 2019.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

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

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <https://www.rfc-editor.org/info/rfc8281>.

Authors' Addresses

Clarence Filsfils
Cisco Systems, Inc.
Pegasus Parc
De kleetlaan 6a, DIEGEM BRABANT 1831
BELGIUM

Email: cfilsfil@cisco.com

Siva Sivabalan (editor)
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: msiva@cisco.com

Daniel Voyer
Bell Canada
671 de la gauchetiere W
Montreal, Quebec H3B 2M8
Canada

Email: daniel.voyer@bell.ca

Alex Bogdanov
Google, Inc.

Email: bogdanov@google.com

Paul Mattes
Microsoft
One Microsoft Way
Redmond, WA 98052-6399
USA

Email: pamattes@microsoft.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2021

C. Filsfils
K. Talaulikar, Ed.
Cisco Systems, Inc.
D. Voyer
Bell Canada
A. Bogdanov
Google, Inc.
P. Mattes
Microsoft

July 7, 2020

Segment Routing Policy Architecture
draft-ietf-spring-segment-routing-policy-08

Abstract

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing. The headend node steers a flow into an SR Policy. The header of a packet steered in an SR Policy is augmented with an ordered list of segments associated with that SR Policy. This document details the concepts of SR Policy and steering into an SR Policy.

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 8, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing [RFC8402].

The headend node is said to steer a flow into an Segment Routing Policy (SR Policy).

The header of a packet steered into an SR Policy is augmented with an ordered list of segments associated with that SR Policy.

This document details the concepts of SR Policy and steering packets into an SR Policy. These apply equally to the MPLS and SRv6 instantiations of segment routing.

For reading simplicity, the illustrations are provided for the MPLS instantiations.

1.1. Requirements Language

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

2. SR Policy

An SR Policy is a framework that enables instantiation of an ordered list of segments on a node for implementing a source routing policy with a specific intent for traffic steering from that node.

The Segment Routing architecture [RFC8402] specifies that any instruction can be bound to a segment. Thus, an SR Policy can be built using any type of Segment Identifier (SID) including those associated with topological or service instructions.

This section defines the key aspects and constituents of an SR Policy.

2.1. Identification of an SR Policy

An SR Policy is identified through the tuple <headend, color, endpoint>. In the context of a specific headend, one may identify an SR policy by the <color, endpoint> tuple.

The headend is the node where the policy is instantiated/implemented. The headend is specified as an IPv4 or IPv6 address and is expected to be unique in the domain.

The endpoint indicates the destination of the policy. The endpoint is specified as an IPv4 or IPv6 address and is expected to be unique in the domain. In a specific case (refer to Section 8.8.1), the endpoint can be the null address (0.0.0.0 for IPv4, ::0 for IPv6).

The color is a 32-bit numerical value that associates the SR Policy with an intent (e.g. low-latency).

The endpoint and the color are used to automate the steering of service or transport routes on SR Policies (refer to Section 8).

An implementation MAY allow assignment of a symbolic name comprising of printable ASCII characters to an SR Policy to serve as a user-friendly attribute for debug and troubleshooting purposes. Such symbolic names may identify an SR Policy when the naming scheme ensures uniqueness. The SR Policy name may also be signaled along with a candidate path of the SR Policy (refer Section 2.2). An SR Policy may have multiple names associated with it in the scenario where the headend receives different SR Policy names along with candidate paths for the same SR Policy.

2.2. Candidate Path and Segment List

An SR Policy is associated with one or more candidate paths. A candidate path is the unit for signaling of an SR Policy to a headend via protocols like Path Computation Element (PCE) Communication Protocol (PCEP) [RFC8664] or BGP SR Policy [I-D.ietf-idr-segment-routing-te-policy].

A Segment-List represents a specific source-routed path to send traffic from the headend to the endpoint of the corresponding SR policy.

A candidate path is either dynamic or explicit.

An explicit candidate path is expressed as a Segment-List or a set of Segment-Lists.

A dynamic candidate path expresses an optimization objective and a set of constraints. The headend (potentially with the help of a PCE) computes the solution Segment-List (or set of Segment-Lists) that solves the optimization problem.

If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with a weight for weighted load balancing (refer Section 2.11 for details). The default weight is 1.

2.3. Protocol-Origin of a Candidate Path

A headend may be informed about a candidate path for an SR Policy <color, endpoint> by various means including: via configuration, PCEP [RFC8664] or BGP [I-D.ietf-idr-segment-routing-te-policy].

Protocol-Origin of a candidate path is an 8-bit value which identifies the component or protocol that originates or signals the candidate path.

The head-end assigns different Protocol-Origin Priority values to each Protocol-Origin. The Protocol-Origin Priority is used as a tie-breaker between candidate-paths of equal preference, as described in Section 2.9. The table below specifies the RECOMMENDED default values of Protocol-Origin Priority:

Value	Protocol-Origin
10	PCEP
20	BGP SR Policy
30	Via Configuration

Table 1: Protocol-Origin Priority default values

Implementations MAY allow modifications of these default values assigned to protocols on the headend along similar lines as a routing administrative distance. Its application in the candidate path selection is described in Section 2.9.

2.4. Originator of a Candidate Path

Originator identifies the node which provisioned or signalled the candidate path on the headend. The originator is expressed in the form of a 160 bit numerical value formed by the concatenation of the fields of the tuple <ASN, node-address> as below:

- ASN : represented as a 4 byte number.
- Node Address : represented as a 128 bit value. IPv4 addresses are encoded in the lowest 32 bits.

Its application in the candidate path selection is described in Section 2.9.

When Protocol-Origin is Via Configuration, the ASN and node address MAY be set to either the headend or the provisioning controller/node ASN and address. Default value is 0 for both AS and node address.

When Protocol-Origin is PCEP, it is the IPv4 or IPv6 address of the PCE and the AS number SHOULD be set to 0 by default when not available or known.

When Protocol-Origin is BGP SR Policy, the ASN and Node Address are provided by BGP (refer [I-D.ietf-idr-segment-routing-te-policy]) on the headend.

2.5. Discriminator of a Candidate Path

The Discriminator is a 32 bit value associated with a candidate path that uniquely identifies it within the context of an SR Policy from a specific Protocol-Origin as specified below:

When Protocol-Origin is Via Configuration, this is an implementation's configuration model specific unique identifier for a candidate path. Default value is 0.

When PCEP is the Protocol-Origin, the method to uniquely identify signalled path will be specified in a future PCEP document. Default value is 0.

When BGP SR Policy is the Protocol-Origin, it is the distinguisher (refer Section 2.1 of [I-D.ietf-idr-segment-routing-te-policy]).

Its application in the candidate path selection is described in Section 2.9.

2.6. Identification of a Candidate Path

A candidate path is identified in the context of a single SR Policy.

A candidate path is not shared across SR Policies.

A candidate path is not identified by its Segment-List(s).

> If CP1 is a candidate path of SR Policy Pol1 and CP2 is a candidate path of SR Policy Pol2, then these two candidate paths are independent, even if they happen to have the same Segment-List. The Segment-List does not identify a candidate path. The Segment-List is an attribute of a candidate path.

The identity of a candidate path MUST be uniquely established in the context of an SR Policy <headend, color, endpoint> in order to handle add, delete or modify operations on them in an unambiguous manner regardless of their source(s).

The tuple <Protocol-Origin, originator, discriminator> uniquely identifies a candidate path.

Candidate paths MAY also be assigned or signaled with a symbolic name comprising printable ASCII characters to serve as a user-friendly attribute for debug and troubleshooting purposes. Such symbolic names MUST NOT be considered as identifiers for a candidate path.

2.7. Preference of a Candidate Path

The preference of the candidate path is used to select the best candidate path for an SR Policy. The default preference is 100.

It is RECOMMENDED that each candidate path of a given SR policy has a different preference.

2.8. Validity of a Candidate Path

A candidate path is usable when it valid. A common path validity criterion is the reachability of its constituent SIDs. The validation rules are specified in Section 5.

2.9. Active Candidate Path

A candidate path is selected when it is valid and it is determined to be the best path of the SR Policy. The selected path is referred to as the "active path" of the SR policy in this document.

Whenever a new path is learned or an active path is deleted, the validity of an existing path changes or an existing path is changed, the selection process MUST be re-executed.

The candidate path selection process operates on the candidate path Preference. A candidate path is selected when it is valid and it has the highest preference value among all the candidate paths of the SR Policy.

In the case of multiple valid candidate paths of the same preference, the tie-breaking rules are evaluated on the identification tuple in the following order until only one valid best path is selected:

1. Higher value of Protocol-Origin Priority is selected.

2. If specified by configuration, prefer the existing installed path.

3. Lower value of originator is selected.

4. Finally, the higher value of discriminator is selected.

The rules are framed with multiple protocols and sources in mind and hence may not follow the logic of a single protocol (e.g. BGP best path selection). The motivation behind these rules are as follows:

- The Protocol-Origin allows an operator to setup a default selection mechanism across protocol sources, e.g., to prefer configured over paths signalled via BGP SR Policy or PCEP.

- The preference, being the first tiebreaker, allows an operator to influence selection across paths thus allowing provisioning of multiple path options, e.g., CP1 is preferred and if it becomes invalid then fall-back to CP2 and so on. Since preference works across protocol sources it also enables (where necessary) selective override of the default protocol-origin preference, e.g., to prefer a path signalled via BGP SR Policy over what is configured.

- The originator allows an operator to have multiple redundant controllers and still maintain a deterministic behaviour over which of them are preferred even if they are providing the same candidate paths for the same SR policies to the headend.

- The discriminator performs the final tiebreaking step to ensure a deterministic outcome of selection regardless of the order in which candidate paths are signalled across multiple transport channels or sessions.

[I-D.filsfils-spring-sr-policy-considerations] provides a set of examples to illustrate the active candidate path selection rules.

2.10. Validity of an SR Policy

An SR Policy is valid when it has at least one valid candidate path.

2.11. Instantiation of an SR Policy in the Forwarding Plane

A valid SR Policy is instantiated in the forwarding plane.

Only the active candidate path SHOULD be used for forwarding traffic that is being steered onto that policy.

If a set of Segment-Lists is associated with the active path of the policy, then the steering is per flow and W-ECMP based according to the relative weight of each Segment-List.

The fraction of the flows associated with a given Segment-List is w/ Sw where w is the weight of the Segment-List and Sw is the sum of the weights of the Segment-Lists of the selected path of the SR Policy.

The accuracy of the weighted load-balancing depends on the platform implementation.

2.12. Priority of an SR Policy

Upon topological change, many policies could be recomputed or revalidated. An implementation MAY provide a per-policy priority configuration. The operator MAY set this field to indicate order in which the policies should be re-computed. Such a priority is represented by an integer in the range (0, 255) where the lowest value is the highest priority. The default value of priority is 128.

An SR Policy may comprise multiple Candidate Paths received from the same or different sources. A candidate path MAY be signaled with a priority value. When an SR Policy has multiple candidate paths with distinct signaled non-default priority values, the SR Policy as a whole takes the lowest value (i.e. the highest priority) amongst these signaled priority values.

2.13. Summary

In summary, the information model is the following:

SR policy POL1 <headend = H1, color = 1, endpoint = E1>
Candidate-path CP1 <protocol-origin = 20, originator = 100:1.1.1.1, discriminator = 1>

```
            Preference 200
            Weight W1, SID-List1 <SID11...SID1i>
            Weight W2, SID-List2 <SID21...SID2j>
       Candidate-path CP2 <protocol-origin = 20, originator =
  100:2.2.2.2, discriminator = 2>
            Preference 100
            Weight W3, SID-List3 <SID31...SID3i>
            Weight W4, SID-List4 <SID41...SID4j>
```

The SR Policy POL1 is identified by the tuple <headend, color, endpoint>. It has two candidate paths CP1 and CP2. Each is identified by a tuple <protocol-origin, originator, discriminator>. CP1 is the active candidate path (it is valid and it has the highest preference). The two Segment-Lists of CP1 are installed as the forwarding instantiation of SR policy POL1. Traffic steered on POL1 is flow-based hashed on Segment-List <SID11...SID1i> with a ratio W1/(W1+W2).

3. Segment Routing Database

An SR headend maintains the Segment Routing Database (SR-DB). The SR-DB is a conceptual database to illustrate the various pieces of information and their sources that may help in SR Policy computation and validation. There is no specific requirement for an implementation to create a new database as such.

An SR headend leverages the SR-DB to validate explicit candidate paths and compute dynamic candidate paths.

The information in the SR-DB MAY include:

- IGP information (topology, IGP metrics based on ISIS [RFC1195] and OSPF [RFC2328] [RFC5340])
- Segment Routing information (such as SRGB, SRLB, Prefix-SIDs, Adj-SIDs, BGP Peering SID, SRv6 SIDs) [RFC8402] [I-D.ietf-spring-srv6-network-programming]
- TE Link Attributes (such as TE metric, SRLG, attribute-flag, extended admin group) [RFC5305] [RFC3630].
- Extended TE Link attributes (such as latency, loss) [RFC8570] [RFC7471]
- Inter-AS Topology information [I-D.ietf-idr-bgpls-segment-routing-epe].

The attached domain topology MAY be learned via IGP, BGP-LS or NETCONF.

A non-attached (remote) domain topology MAY be learned via BGP-LS or NETCONF.

In some use-cases, the SR-DB may only contain the attached domain topology while in others, the SR-DB may contain the topology of multiple domains and in this case it is multi-domain capable.

The SR-DB MAY also contain the SR Policies instantiated in the network. This can be collected via BGP-LS [I-D.ietf-idr-te-lsp-distribution] or PCEP [RFC8231] and [I-D.ietf-pce-binding-label-sid]. This information allows to build an end-to-end policy on the basis of intermediate SR policies (see Section 6 for further details).

The SR-DB MAY also contain the Maximum SID Depth (MSD) capability of nodes in the topology. This can be collected via ISIS [RFC8491], OSPF [RFC8476], BGP-LS [I-D.ietf-idr-bgp-ls-segment-routing-msd] or PCEP [RFC8664].

The use of the SR-DB for computation and validation of SR Policies is outside the scope of this document. Some implementation aspects related to this are covered in [I-D.filsfils-spring-sr-policy-considerations].

4. Segment Types

A Segment-List is an ordered set of segments represented as <S1, S2, ... Sn> where S1 is the first segment.

Based on the desired dataplane, either the MPLS label stack or the SRv6 SRH is built from the Segment-List. However, the Segment-List itself can be specified using different segment-descriptor types and the following are currently defined:

Type A: SR-MPLS Label:
: A MPLS label corresponding to any of the segment types defined for SR-MPLS (as defined in [RFC8402] or other SR-MPLS specifications) can be used. Additionally, reserved labels like explicit-null or in general any MPLS label may also be used. E.g. this type can be used to specify a label representation which maps to an optical transport path on a packet transport node. This type does not require the headend to perform SID resolution.

Type B: SRv6 SID:
: An IPv6 address corresponding to any of the SID behaviors for SRv6 (as defined in [I-D.ietf-spring-srv6-network-programming] or other SRv6 specifications) can be used. This type does not require the headend to perform SID resolution.

Type C: IPv4 Prefix with optional SR Algorithm:

The headend is required to resolve the specified IPv4 Prefix Address to the SR-MPLS label corresponding to a Prefix SID segment (as defined in [RFC8402]). The SR algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type D: IPv6 Global Prefix with optional SR Algorithm for SR-MPLS:
In this case the headend is required to resolve the specified IPv6 Global Prefix Address to the SR-MPLS label corresponding to its Prefix SID segment (as defined in [RFC8402]). The SR Algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type E: IPv4 Prefix with Local Interface ID:
This type allows identification of Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]) label for point-to-point links including IP unnumbered links. The headend is required to resolve the specified IPv4 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. The Local Interface ID link descriptor follows semantics as specified in [RFC7752]. This type can also be used to indicate indirection into a layer 2 interface (i.e. without IP address) like a representation of an optical transport path or a layer 2 Ethernet port or circuit at the specified node.

Type F: IPv4 Addresses for link endpoints as Local, Remote pair:
This type allows identification of Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]) label for links. The headend is required to resolve the specified IPv4 Local Address to the Node originating it and then use the IPv4 Remote Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

Type G: IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SR-MPLS:
This type allows identification of Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]) label for links including those with only Link Local IPv6 addresses. The headend is required to resolve the specified IPv6 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. For other than point-to-point links, additionally the specific adjacency over the link needs to be resolved using the Remote Prefix and Interface ID. The Local and Remote pair of Prefix and Interface ID link descriptor follows semantics as specified in [RFC7752]. This

type can also be used to indicate indirection into a layer 2 interface (i.e. without IP address) like a representation of an optical transport path or a layer 2 Ethernet port or circuit at the specified node.

Type H: IPv6 Addresses for link endpoints as Local, Remote pair for SR-MPLS:
This type allows identification of Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]) label for links with Global IPv6 addresses. The headend is required to resolve the specified Local IPv6 Address to the Node originating it and then use the Remote IPv6 Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

Type I: IPv6 Global Prefix with optional SR Algorithm for SRv6:
The headend is required to resolve the specified IPv6 Global Prefix Address to an SRv6 SID corresponding to a Prefix SID segment (as defined in [RFC8402]), such as a SID associated with the End behavior (as defined in [I-D.ietf-spring-srv6-network-programming]), of the node which is originating the prefix. The SR Algorithm (refer to Section 3.1.1 of [RFC8402]) to be used MAY also be provided.

Type J: IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SRv6:
This type allows identification of an SRv6 SID corresponding to an Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]), such as a SID associated with the End.X behavior (as defined in [I-D.ietf-spring-srv6-network-programming]), associated with link or adjacency with only Link Local IPv6 addresses. The headend is required to resolve the specified IPv6 Prefix Address to the Node originating it and then use the Local Interface ID to identify the point-to-point link whose adjacency is being referred to. For other than point-to-point links, additionally the specific adjacency needs to be resolved using the Remote Prefix and Interface ID. The Local and Remote pair of Prefix and Interface ID link descriptor follows semantics as specified in [RFC7752].

Type K: IPv6 Addresses for link endpoints as Local, Remote pair for SRv6:
This type allows identification of an SRv6 SID corresponding to an Adjacency SID or BGP Peer Adjacency SID (as defined in [RFC8402]), such as a SID associated with the End.X behavior (as defined in [I-D.ietf-spring-srv6-network-programming]), associated with link or adjacency with Global IPv6 addresses. The headend is required to resolve the specified Local IPv6

Address to the Node originating it and then use the Remote IPv6 Address to identify the link adjacency being referred to. The Local and Remote Address pair link descriptors follows semantics as specified in [RFC7752].

Type L: SRv6 SID with Behavior
An SRv6 SID along with its behavior (as defined in [I-D.ietf-spring-srv6-network-programming] or other SRv6 specifications) and structure (as defined in [I-D.ietf-spring-srv6-network-programming]) can be used. This type enables the headend to optionally perform validation of the SID when using it for building the Segment List.

When the algorithm is not specified for the SID types above which optionally allow for it, the headend SHOULD use the Strict Shortest Path algorithm if available; otherwise it SHOULD use the default Shortest Path algorithm. The specification of algorithm enables the use of IGP Flex Algorithm [I-D.ietf-lsr-flex-algo] specific SIDs in SR Policy.

For SID types C-through-L, a SID value may also be optionally provided to the headend for verification purposes. Section 5.1. describes the resolution and verification of the SIDs and Segment Lists on the headend.

When building the MPLS label stack or the IPv6 Segment list from the Segment List, the node instantiating the policy MUST interpret the set of Segments as follows:

- The first Segment represents the topmost label or the first IPv6 segment. It identifies the active segment the traffic will be directed toward along the explicit SR path.
- The last Segment represents the bottommost label or the last IPv6 segment the traffic will be directed toward along the explicit SR path.

4.1. Explicit Null

A Type A SID may be any MPLS label, including reserved labels.

For example, assuming that the desired traffic-engineered path from a headend 1 to an endpoint 4 can be expressed by the Segment-List <16002, 16003, 16004> where 16002, 16003 and 16004 respectively refer to the IPv4 Prefix SIDs bound to node 2, 3 and 4, then IPv6 traffic can be traffic-engineered from nodes 1 to 4 via the previously described path using an SR Policy with Segment-List <16002, 16003,

16004, 2> where mpls label value of 2 represents the "IPv6 Explicit NULL Label".

The penultimate node before node 4 will pop 16004 and will forward the frame on its directly connected interface to node 4.

The endpoint receives the traffic with top label "2" which indicates that the payload is an IPv6 packet.

When steering unlabeled IPv6 BGP destination traffic using an SR policy composed of Segment-List(s) based on IPv4 SIDs, the Explicit Null Label Policy is processed as specified in [I-D.ietf-idr-segment-routing-te-policy]) Section 2.4.4. When an "IPv6 Explicit NULL label" is not present as the bottom label, the headend SHOULD automatically impose one. Refer to Section 8 for more details.

5. Validity of a Candidate Path

5.1. Explicit Candidate Path

An explicit candidate path is associated with a Segment-List or a set of Segment-Lists.

An explicit candidate path is provisioned by the operator directly or via a controller.

The computation/logic that leads to the choice of the Segment-List is external to the SR Policy headend. The SR Policy headend does not compute the Segment-List. The SR Policy headend only confirms its validity.

An explicit candidate path MAY consist of a single explicit Segment-List containing only an implicit-null label to indicate pop-and-forward behavior. The BSID is popped and the traffic is forwarded based on the inner label or an IP lookup in the case of unlabeled IP packets. Such an explicit path can serve as a fallback or path of last resort for traffic being steered into an SR Policy using its BSID (refer to Section 8.3).

A Segment-List of an explicit candidate path MUST be declared invalid when:

- It is empty.
- Its weight is 0.
- The headend is unable to perform path resolution for the first SID into one or more outgoing interface(s) and next-hop(s).

- The headend is unable to perform SID resolution for any non-first SID of type C-through-L into an MPLS label or an SRv6 SID.
- The headend verification fails for any SID for which verification has been explicitly requested.

"Unable to perform path resolution" means that the headend has no path to the SID in its SR database.

SID verification is performed when the headend is explicitly requested to verify SID(s) by the controller via the signaling protocol used. Implementations MAY provide a local configuration option to enable verification on a global or per policy or per candidate path basis.

"Verification fails" for a SID means any of the following:

- The headend is unable to find the SID in its SR DB
- The headend detects mis-match between the SID value and its context provided for SIDs of type C-through-L in its SR DB.
- The headend is unable to perform SID resolution for any non-first SID of type C-through-L into an MPLS label or an SRv6 SID.

In multi-domain deployments, it is expected that the headend be unable to verify the reachability of the SIDs in remote domains. Types A or B MUST be used for the SIDs for which the reachability cannot be verified. Note that the first SID MUST always be reachable regardless of its type.

In addition, a Segment-List MAY be declared invalid when:

- Its last segment is not a Prefix SID (including BGP Peer Node-SID) advertised by the node specified as the endpoint of the corresponding SR policy.
- Its last segment is not an Adjacency SID (including BGP Peer Adjacency SID) of any of the links present on neighbor nodes and that terminate on the node specified as the endpoint of the corresponding SR policy.

An explicit candidate path is invalid as soon as it has no valid Segment-List.

5.2. Dynamic Candidate Path

A dynamic candidate path is specified as an optimization objective and constraints.

The headend of the policy leverages its SR database to compute a Segment-List ("solution Segment-List") that solves this optimization problem.

The headend re-computes the solution Segment-List any time the inputs to the problem change (e.g., topology changes).

When local computation is not possible (e.g., a policy's tailend is outside the topology known to the headend) or not desired, the headend MAY send path computation request to a PCE supporting PCEP extension specified in [RFC8664].

If no solution is found to the optimization objective and constraints, then the dynamic candidate path MUST be declared invalid.

[I-D.filsfils-spring-sr-policy-considerations] discusses some of the optimization objectives and constraints that may be considered by a dynamic candidate path. It illustrates some of the desirable properties of the computation of the solution Segment-List.

6. Binding SID

The Binding SID (BSID) is fundamental to Segment Routing [RFC8402]. It provides scaling, network opacity and service independence. [I-D.filsfils-spring-sr-policy-considerations] illustrates some of these benefits. This section describes the association of BSID with an SR Policy.

6.1. BSID of a candidate path

Each candidate path MAY be defined with a BSID.

Candidate Paths of the same SR policy SHOULD have the same BSID.

Candidate Paths of different SR policies MUST NOT have the same BSID.

6.2. BSID of an SR Policy

The BSID of an SR Policy is the BSID of its active candidate path.

When the active candidate path has a specified BSID, the SR Policy uses that BSID if this value (label in MPLS, IPv6 address in SRv6) is available (i.e., not associated with any other usage: e.g. to another MPLS client, to another SID, to another SR Policy).

Optionally, instead of only checking that the BSID of the active path is available, a headend MAY check that it is available within a given

SID range i.e., Segment Routing Local Block (SRLB) as specified in [RFC8402].

When the specified BSID is not available (optionally is not in the SRLB), an alert message MUST be generated.

In the cases (as described above) where SR Policy does not have a BSID available, then the SR Policy MAY dynamically bind a BSID to itself. Dynamically bound BSID SHOULD use an available SID outside the SRLB.

Assuming that at time t the BSID of the SR Policy is B1, if at time t+dt a different candidate path becomes active and this new active path does not have a specified BSID or its BSID is specified but is not available (e.g. it is in use by something else), then the SR Policy keeps the previous BSID B1.

The association of an SR Policy with a BSID thus MAY change over the life of the SR Policy (e.g., upon active path change). Hence, the BSID SHOULD NOT be used as an identification of an SR Policy.

6.2.1. Frequent use-case : unspecified BSID

All the candidate paths of the same SR Policy can have an unspecified BSID.

In such a case, a BSID MAY be dynamically bound to the SR Policy as soon as the first valid candidate path is received. That BSID is kept along all the life of the SR Policy and across changes of active candidate path.

6.2.2. Frequent use-case: all specified to the same BSID

All the paths of the SR Policy can have the same specified BSID.

6.2.3. Specified-BSID-only

An implementation MAY support the configuration of the Specified-BSID-only restrictive behavior on the headend for all SR Policies or individual SR Policies. Further, this restrictive behavior MAY also be signaled on a per SR Policy basis to the headend.

When this restrictive behavior is enabled, if the candidate path has an unspecified BSID or if the specified BSID is not available when the candidate path becomes active then no BSID is bound to it and it is considered invalid. An alert MUST be triggered for this error. Other candidate paths MUST then be evaluated for becoming the active candidate path.

6.3. Forwarding Plane

A valid SR Policy installs a BSID-keyed entry in the forwarding plane with the action of steering the packets matching this entry to the selected path of the SR Policy.

If the Specified-BSID-only restrictive behavior is enabled and the BSID of the active path is not available (optionally not in the SRLB), then the SR Policy does not install any entry indexed by a BSID in the forwarding plane.

6.4. Non-SR usage of Binding SID

An implementation MAY choose to associate a Binding SID with any type of interface (e.g. a layer 3 termination of an Optical Circuit) or a tunnel (e.g. IP tunnel, GRE tunnel, IP/UDP tunnel, MPLS RSVP-TE tunnel, etc). This enables the use of other non-SR enabled interfaces and tunnels as segments in an SR Policy Segment-List without the need of forming routing protocol adjacencies over them.

The details of this kind of usage are beyond the scope of this document. A specific packet optical integration use case is described in [I-D.anand-spring-poi-sr].

7. SR Policy State

The SR Policy State is maintained on the headend to represent the state of the policy and its candidate paths. This is to provide an accurate representation of whether the SR Policy is being instantiated in the forwarding plane and which of its candidate paths and segment-list(s) are active. The SR Policy state MUST also reflect the reason when a policy and/or its candidate path is not active due to validation errors or not being preferred.

The SR Policy state can be reported by the headend node via BGP-LS [I-D.ietf-idr-te-lsp-distribution] or PCEP [RFC8231] and [I-D.ietf-pce-binding-label-sid].

SR Policy state on the headend also includes traffic accounting information for the flows being steered via the policies. The details of the SR Policy accounting are beyond the scope of this document. The aspects related to the SR traffic counters and their usage in the broader context of traffic accounting in a SR network are covered in [I-D.filsfils-spring-sr-traffic-counters] and [I-D.ali-spring-sr-traffic-accounting] respectively.

Implementations MAY support an administrative state to control locally provisioned policies via mechanisms like CLI or NETCONF.

8. Steering into an SR Policy

A headend can steer a packet flow into a valid SR Policy in various ways:

- Incoming packets have an active SID matching a local BSID at the headend.
- Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.
- Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

For simplicity of illustration, this document uses the SR-MPLS example.

8.1. Validity of an SR Policy

An SR Policy is invalid when all its candidate paths are invalid as described in Section 5 and Section 2.10.

By default, upon transitioning to the invalid state,

- an SR Policy and its BSID are removed from the forwarding plane.
- any steering of a service (PW), destination (BGP-VPN), flow or packet on the related SR policy is disabled and the related service, destination, flow or packet is routed per the classic forwarding table (e.g. longest-match to the destination or the recursing next-hop).

8.2. Drop upon invalid SR Policy

An SR Policy MAY be enabled for the Drop-Upon-Invalid behavior:

- an invalid SR Policy and its BSID is kept in the forwarding plane with an action to drop.
- any steering of a service (PW), destination (BGP-VPN), flow or packet on the related SR policy is maintained with the action to drop all of this traffic.

The drop-upon-invalid behavior has been deployed in use-cases where the operator wants some PW to only be transported on a path with specific constraints. When these constraints are no longer met, the operator wants the PW traffic to be dropped. Specifically, the operator does not want the PW to be routed according to the IGP shortest-path to the PW endpoint.

8.3. Incoming Active SID is a BSID

Let us assume that headend H has a valid SR Policy P of Segment-List <S1, S2, S3> and BSID B.

When H receives a packet K with label stack <B, L2, L3>, H pops B and pushes <S1, S2, S3> and forwards the resulting packet according to SID S1.

> "Forwarding the resulting packet according to S1" means: If S1 is an Adj SID or a PHP-enabled prefix SID advertised by a neighbor, H sends the resulting packet with label stack <S2, S3, L2, L3> on the outgoing interface associated with S1; Else H sends the resulting packet with label stack <S1, S2, S3, L2, L3> along the path of S1.

H has steered the packet into the SR policy P.

H did not have to classify the packet. The classification was done by a node upstream of H (e.g., the source of the packet or an intermediate ingress edge node of the SR domain) and the result of this classification was efficiently encoded in the packet header as a BSID.

This is another key benefit of the segment routing in general and the binding SID in particular: the ability to encode a classification and the resulting steering in the packet header to better scale and simplify intermediate aggregation nodes.

If the SR Policy P is invalid, the BSID B is not in the forwarding plane and hence the packet K is dropped by H.

8.4. Per-Destination Steering

Let us assume that headend H:

- learns a BGP route R/r via next-hop N, extended-color community C and VPN label V.
- has a valid SR Policy P to (color = C, endpoint = N) of Segment-List <S1, S2, S3> and BSID B.
- has a BGP policy which matches on the extended-color community C and allows its usage as SLA steering information.

If all these conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P of BSID B instead of via N.

Indeed, H's local BGP policy and the received BGP route indicate that the headend should associate R/r with an SR Policy path to endpoint N

with the SLA associated with color C. The headend therefore installs the BGP route on that policy.

This can be implemented by using the BSID as a generalized next-hop and installing the BGP route on that generalized next-hop.

When H receives a packet K with a destination matching R/r, H pushes the label stack <S1, S2, S3, V> and sends the resulting packet along the path to S1.

Note that any SID associated with the BGP route is inserted after the Segment-List of the SR Policy (i.e., <S1, S2, S3, V>).

The same behavior is applicable to any type of service route: any AFI/SAFI of BGP [RFC4760] any AFI/SAFI of LISP [RFC6830].

8.4.1. Multiple Colors

When a BGP route has multiple extended-color communities each with a valid SR Policy NLRI, the BGP process installs the route on the SR policy whose color is of highest numerical value.

Let us assume that headend H:

- learns a BGP route R/r via next-hop N, extended-color communities C1 and C2 and VPN label V.
- has a valid SR Policy P1 to (color = C1, endpoint = N) of Segment-List <S1, S2, S3> and BSID B1.
- has a valid SR Policy P2 to (color = C2, endpoint = N) of Segment-List <S4, S5, S6> and BSID B2.
- has a BGP policy which matches on the extended-color communities C1 and C2 and allows their usage as SLA steering information

If all these conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P2 of BSID=B2 (instead of N) because C2 > C1.

8.5. Recursion on an on-demand dynamic BSID

In the previous section, it was assumed that H had a pre-established "explicit" SR Policy (color C, endpoint N).

In this section, independently to the a-priori existence of any explicit candidate path of the SR policy (C, N), it is to be noted that the BGP process at headend node H triggers the instantiation of a dynamic candidate path for the SR policy (C, N) as soon as:

- the BGP process learns of a route R/r via N and with color C.

- a local policy at node H authorizes the on-demand SR Policy path instantiation and maps the color to a dynamic SR Policy path optimization template.

8.5.1. Multiple Colors

When a BGP route R/r via N has multiple extended-color communities Ci (with i=1 ... n), an individual on-demand SR Policy dynamic path request (color Ci, endpoint N) is triggered for each color Ci.

8.6. Per-Flow Steering

Let us assume that headend H:

- has a valid SR Policy P1 to (color = C1, endpoint = N) of Segment-List <S1, S2, S3> and BSID B1.
- has a valid SR Policy P2 to (color = C2, endpoint = N) of Segment-List <S4, S5, S6> and BSID B2.
- is configured to instantiate an array of paths to N where the entry 0 is the IGP path to N, color C1 is the first entry and Color C2 is the second entry. The index into the array is called a Forwarding Class (FC). The index can have values 0 to 7.
- is configured to match flows in its ingress interfaces (upon any field such as Ethernet destination/source/vlan/tos or IP destination/source/DSCP or transport ports etc.) and color them with an internal per-packet forwarding-class variable (0, 1 or 2 in this example).

If all these conditions are met, H installs in RIB/FIB:

- N via a recursion on an array A (instead of the immediate outgoing link associated with the IGP shortest-path to N).
- Entry A(0) set to the immediate outgoing link of the IGP shortest-path to N.
- Entry A(1) set to SR Policy P1 of BSID=B1.
- Entry A(2) set to SR Policy P2 of BSID=B2.

H receives three packets K, K1 and K2 on its incoming interface. These three packets either longest-match on N or more likely on a BGP/service route which recurses on N. H colors these 3 packets respectively with forwarding-class 0, 1 and 2. As a result:

- H forwards K along the shortest-path to N (which in SR-MPLS results in the pushing of the prefix-SID of N).
- H pushes <S1, S2, S3> on packet K1 and forwards the resulting frame along the shortest-path to S1.
- H pushes <S4, S5, S6> on packet K2 and forwards the resulting frame along the shortest-path to S4.

If the local configuration does not specify any explicit forwarding information for an entry of the array, then this entry is filled with the same information as entry 0 (i.e. the IGP shortest-path).

If the SR Policy mapped to an entry of the array becomes invalid, then this entry is filled with the same information as entry 0. When all the array entries have the same information as entry0, the forwarding entry for N is updated to bypass the array and point directly to its outgoing interface and next-hop.

The array index values (e.g. 0, 1 and 2) and the notion of forwarding-class are implementation specific and only meant to describe the desired behavior. The same can be realized by other mechanisms.

This realizes per-flow steering: different flows bound to the same BGP endpoint are steered on different IGP or SR Policy paths.

A headend MAY support options to apply per-flow steering only for traffic matching specific prefixes (e.g. specific IGP or BGP prefixes).

8.7. Policy-based Routing

Finally, headend H may be configured with a local routing policy which overrides any BGP/IGP path and steer a specified packet on an SR Policy. This includes the use of mechanisms like IGP Shortcut for automatic routing of IGP prefixes over SR Policies intended for such purpose.

8.8. Optional Steering Modes for BGP Destinations

8.8.1. Color-Only BGP Destination Steering

In the previous section, it is seen that the steering on an SR Policy is governed by the matching of the BGP route’s next-hop N and the authorized color C with an SR Policy defined by the tuple (N, C).

This is the most likely form of BGP destination steering and the one recommended for most use-cases.

This section defines an alternative steering mechanism based only on the color.

This color-only steering variation is governed by two new flags "C" and "O" defined in the color extended community [ref draft-ietf-idr-segment-routing-te-policy section 3].

The Color-Only flags "CO" are set to 00 by default.

When 00, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is the IPv4 or IPv6

            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

This is the classic case described in this document previously and what is recommended in most scenarios.

When 01, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is the IPv4 or IPv6

            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE IF there is a valid SR Policy (null endpoint, C) of the
            same address-family of N;
        Steer into SR Policy (null endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family null endpoint, C);
        Steer into SR Policy (any null endpoint, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

When 10, the BGP destination is steered as follows:

```
    IF there is a valid SR Policy (N, C) where N is an IPv4 or IPv6
            endpoint address and C is a color;
        Steer into SR Policy (N, C);
    ELSE IF there is a valid SR Policy (null endpoint, C)
            of the same address-family of N;
        Steer into SR Policy (null endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family null endpoint, C);
        Steer into SR Policy (any null endpoint, C);
    ELSE IF there is any valid SR Policy (any endpoint, C)
            of the same address-family of N;
        Steer into SR Policy (any endpoint, C);
    ELSE IF there is any valid SR Policy
            (any address-family endpoint, C);
        Steer into SR Policy (any address-family endpoint, C);
    ELSE;
        Steer on the IGP path to the next-hop N.
```

The null endpoint is 0.0.0.0 for IPv4 and ::0 for IPv6 (all bits set to the 0 value).

The value 11 is reserved for future use and SHOULD NOT be used. Upon reception, an implementations MUST treat it like 00.

8.8.2. Multiple Colors and CO flags

The steering preference is first based on highest color value and then CO-dependent for the color. Assuming a Prefix via (NH, C1(CO=01), C2(CO=01)); C1>C2 The steering preference order is:

- SR policy (NH, C1).
- SR policy (null, C1).
- SR policy (NH, C2).
- SR policy (null, C2).
- IGP to NH.

8.8.3. Drop upon Invalid

This document defined earlier that when all the following conditions are met, H installs R/r in RIB/FIB with next-hop = SR Policy P of BSID B instead of via N.

- H learns a BGP route R/r via next-hop N, extended-color community C and VPN label V.
- H has a valid SR Policy P to (color = C, endpoint = N) of Segment-List <S1, S2, S3> and BSID B.
- H has a BGP policy which matches on the extended-color community C and allows its usage as SLA steering information.

This behavior is extended by noting that the BGP policy may require the BGP steering to always stay on the SR policy whatever its validity.

This is the "drop upon invalid" option described in Section 8.2 applied to BGP-based steering.

9. Protection

9.1. Leveraging TI-LFA local protection of the constituent IGP segments

In any topology, Topology-Independent Loop Free Alternate (TI-LFA) [I-D.ietf-rtgwg-segment-routing-ti-lfa] provides a 50msec local protection technique for IGP SIDs. The backup path is computed on a per IGP SID basis along the post-convergence path.

In a network that has deployed TI-LFA, an SR Policy built on the basis of TI-LFA protected IGP segments leverages the local protection of the constituent segments.

In a network that has deployed TI-LFA, an SR Policy instantiated only with non-protected Adj SIDs does not benefit from any local protection.

9.2. Using an SR Policy to locally protect a link

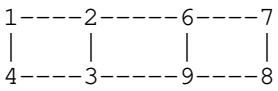


Figure 1: Local protection using SR Policy

An SR Policy can be instantiated at node 2 to protect the link 2to6. A typical explicit Segment-List would be <3, 9, 6>.

A typical use-case occurs for links outside an IGP domain: e.g. 1, 2, 3 and 4 are part of IGP/SR sub-domain 1 while 6, 7, 8 and 9 are part of IGP/SR sub-domain 2. In such a case, links 2to6 and 3to9 cannot benefit from TI-LFA automated local protection. The SR Policy with Segment-List <3, 9, 6> on node 2 can be locally configured to be a fast-reroute backup path for the link 2to6.

9.3. Using a Candidate Path for Path Protection

An SR Policy allows for multiple candidate paths, of which at any point in time there is a single active candidate path that is provisioned in the forwarding plane and used for traffic steering. However, another (lower preference) candidate path MAY be designated as the backup for a specific or all (active) candidate path(s). The following options are possible:

- A pair of disjoint candidate paths are provisioned with one of them as primary and the other is identified as its backup.
- A specific candidate path is provisioned as the backup for any (active) candidate path.
- The headend picks the next (lower) preference valid candidate path as the backup for the active candidate path.

The headend MAY compute a-priori and validate such backup candidate paths as well as provision them into forwarding plane as backup for the active path. A fast re-route mechanism MAY then be used to trigger sub 50msec switchover from the active to the backup candidate

path in the forwarding plane. Mechanisms like BFD MAY be used for fast detection of such failures.

10. Security Considerations

This document specifies in detail the SR Policy construct introduced in [RFC8402] and its instantiation on a router supporting SR along with descriptions of mechanisms for steering of traffic flows over it. Therefore, the security considerations of [RFC8402] apply. This document does not define any new protocol extensions and does not introduce any further security considerations.

11. IANA Considerations

This document requests IANA to create a new top-level registry called "Segment Routing Parameters". This registry is being defined to serve as a top-level registry for keeping all other Segment Routing sub-registries.

The document also requests creation of a new sub-registry called "Segment Types" to be defined under the top-level "Segment Routing Parameters" registry. This sub-registry maintains the alphabetic identifiers for the segment types (as specified in section 4) that may be used within a Segment List of an SR Policy. This sub-registry would follow the Specification Required allocation policy as specified in [RFC8126].

The initial registrations for this sub-registry are as follows:

Value	Description	Reference
A	SR-MPLS Label	[This.ID]
B	SRv6 SID	[This.ID]
C	IPv4 Prefix with optional SR Algorithm	[This.ID]
D	IPv6 Global Prefix with optional SR Algorithm for SR-MPLS	[This.ID]
E	IPv4 Prefix with Local Interface ID	[This.ID]
F	IPv4 Addresses for link endpoints as Local, Remote pair	[This.ID]
G	IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SR-MPLS	[This.ID]
H	IPv6 Addresses for link endpoints as Local, Remote pair for SR-MPLS	[This.ID]
I	IPv6 Global Prefix with optional SR Algorithm for SRv6	[This.ID]
J	IPv6 Prefix and Interface ID for link endpoints as Local, Remote pair for SRv6	[This.ID]
K	IPv6 Addresses for link endpoints as Local, Remote pair for SRv6	[This.ID]
L	SRv6 SID with Behavior	[This.ID]

Table 2: Initial IANA Registration

11.1. Guidance for Designated Experts

The Designated Expert (DE) is expected to ascertain the existence of suitable documentation (a specification) as described in [RFC8126] and to verify that the document is permanently and publicly available. The DE is also expected to check the clarity of purpose and use of the requested assignment. Additionally, the DE must verify that any request for one of these assignments has been made available for review and comment within the IETF: the DE will post the request to the SPRING Working Group mailing list (or a successor mailing list designated by the IESG). If the request comes from within the IETF, it should be documented in an Internet-Draft. Lastly, the DE must ensure that any other request for a code point does not conflict with work that is active or already published within the IETF.

12. Acknowledgement

The authors would like to thank Tarek Saad, Dhanendra Jain, Ruediger Geib, Rob Shakir, Cheng Li and Dhruv Dhody for their review comments and suggestions.

13. Contributors

The following people have contributed to this document:

Siva Sivabalan
Cisco Systems
Email: msiva@cisco.com

Zafar Ali
Cisco Systems
Email: zali@cisco.com

Jose Liste
Cisco Systems
Email: jliste@cisco.com

Francois Clad
Cisco Systems
Email: fclad@cisco.com

Kamran Raza
Cisco Systems
Email: skraza@cisco.com

Mike Koldychev
Cisco Systems
Email: mkoldych@cisco.com

Shraddha Hegde
Juniper Networks
Email: shraddha@juniper.net

Steven Lin
Google, Inc.
Email: stevenlin@google.com

Przemyslaw Krol
Google, Inc.
Email: pkrol@google.com

Martin Horneffer
Deutsche Telekom
Email: martin.horneffer@telekom.de

Dirk Steinberg
Steinberg Consulting
Email: dws@steinbergnet.net

Bruno Decraene
Orange Business Services
Email: bruno.decraene@orange.com

Stephane Litkowski
Orange Business Services
Email: stephane.litkowski@orange.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

14. References

14.1. Normative References

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

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

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

[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>.

14.2. Informative References

[I-D.ali-spring-sr-traffic-accounting]
Filsfils, C., Talaulikar, K., Sivabalan, S., Horneffer, M., Raszuk, R., Litkowski, S., Voyer, D., and R. Morton, "Traffic Accounting in Segment Routing Networks", draft-ali-spring-sr-traffic-accounting-04 (work in progress), February 2020.

[I-D.anand-spring-poi-sr]
Anand, M., Bardhan, S., Subrahmaniam, R., Tantsura, J., Mukhopadhyaya, U., and C. Filsfils, "Packet-Optical Integration in Segment Routing", draft-anand-spring-poi-sr-08 (work in progress), July 2019.

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-05 (work in progress), April 2020.

[I-D.filsfils-spring-sr-traffic-counters]
Filsfils, C., Ali, Z., Horneffer, M., daniel.voyer@bell.ca, d., Durrani, M., and R. Raszuk, "Segment Routing Traffic Accounting Counters", draft-filsfils-spring-sr-traffic-counters-00 (work in progress), June 2018.

[I-D.ietf-idr-bgp-ls-segment-routing-msd]
Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G., and N. Triantafillis, "Signaling MSD (Maximum SID Depth) using Border Gateway Protocol - Link State", draft-ietf-idr-bgp-ls-segment-routing-msd-18 (work in progress), May 2020.

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

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[I-D.ietf-idr-te-lsp-distribution]
Previdi, S., Talaulikar, K., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", draft-ietf-idr-te-lsp-distribution-13 (work in progress), April 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-07 (work in progress), April 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-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-03 (work in progress), March 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.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

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

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

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

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

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

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

Authors' Addresses

Clarence Filsfils
Cisco Systems, Inc.
Pegasus Parc
De kleetlaan 6a, DIEGEM BRABANT 1831
BELGIUM

Email: cfilsfil@cisco.com

Ketan Talaulikar (editor)
Cisco Systems, Inc.
India

Email: ketant@cisco.com

Daniel Voyer
Bell Canada
671 de la gauchetiere W
Montreal, Quebec H3B 2M8
Canada

Email: daniel.voyer@bell.ca

Alex Bogdanov
Google, Inc.

Email: bogdanov@google.com

Paul Mattes
Microsoft
One Microsoft Way
Redmond, WA 98052-6399
USA

Email: pamattes@microsoft.com

SPRING
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

F. Clad, Ed.
Cisco Systems, Inc.
X. Xu, Ed.
Alibaba
C. Filsfils
Cisco Systems, Inc.
D. Bernier
Bell Canada
C. Li
Huawei
B. Decraene
Orange
S. Ma
Mellanox
C. Yadlapalli
AT&T
W. Henderickx
Nokia
S. Salsano
Universita di Roma "Tor Vergata"

March 09, 2020

Service Programming with Segment Routing
draft-ietf-spring-sr-service-programming-02

Abstract

This document defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IPv6 networks, as described in the Segment Routing architecture.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 10, 2020.

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] is an architecture based on the source routing paradigm that seeks the right balance between distributed intelligence and centralized programmability. SR can be used with an MPLS or an IPv6 data plane to steer packets through an ordered list of instructions, called segments. These segments may encode simple routing instructions for forwarding packets along a specific network path, but also steer them through Virtual Network Functions (VNFs) or physical service appliances available in the network.

In an SR network, each of these services, running either on a physical appliance or in a virtual environment, are associated with a segment identifier (SID). These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined together in a SID-list to achieve service programming, but also with other types of segments as defined in [RFC8402]. SR thus provides a fully integrated solution for overlay, underlay and service programming. Furthermore, the IPv6 instantiation of SR (SRv6) [I-D.ietf-spring-srv6-network-programming] supports metadata transportation in the Segment Routing Header [I-D.ietf-6man-segment-routing-header], either natively in the tag field or with extensions such as TLVs.

This document describes how a service can be associated with a SID, including legacy services with no SR capabilities, and how these service SIDs are integrated within an SR policy. The definition of an SR Policy and the traffic steering mechanisms are covered in [I-D.ietf-spring-segment-routing-policy] and hence outside the scope of this document.

The definition of control plane components, such as service segment discovery, is outside the scope of this data plane document. For reference, the option of using BGP extensions to support SR service programming is proposed in [I-D.dawra-idr-bgp-sr-service-chaining].

2. Terminology

This document leverages the terminology proposed in [RFC8402], [RFC8660], [I-D.ietf-6man-segment-routing-header], [I-D.ietf-spring-srv6-network-programming] and [I-D.ietf-spring-segment-routing-policy]. It also introduces the following new terms.

Service segment: A segment associated with a service. The service may either run on a physical appliance or in a virtual environment such as a virtual machine or container.

SR-aware service: A service that is fully capable of processing SR traffic. An SR-aware service can be directly associated with a service segment.

SR-unaware service: A service that is unable to process SR traffic or may behave incorrectly due to presence of SR information in the packet headers. An SR-unaware service can be associated with a service segment through an SR proxy function.

3. Classification and Steering

Classification and steering mechanisms are defined in section 8 of [I-D.ietf-spring-segment-routing-policy] and are independent from the purpose of the SR policy. From the perspective of a headend node classifying and steering traffic into an SR policy, there is no difference whether this policy contains IGP, BGP, peering, VPN or service segments, or any combination of these.

As documented in the above reference, traffic is classified when entering an SR domain. The SR policy headend may, depending on its capabilities, classify the packets on a per-destination basis, via simple FIB entries, or apply more complex policy routing rules requiring to look deeper into the packet. These rules are expected to support basic policy routing such as 5-tuple matching. In addition, the IPv6 SRH tag field defined in [I-D.ietf-6man-segment-routing-header] can be used to identify and classify packets sharing the same set of properties. Classified traffic is then steered into the appropriate SR policy and forwarded as per the SID-list(s) of the active candidate path.

SR traffic can be re-classified by an SR endpoint along the original SR policy (e.g., DPI service) or a transit node intercepting the traffic. This node is the head-end of a new SR policy that is imposed onto the packet, either as a stack of MPLS labels or as an IPv6 SRH.

4. Service Segments

In the context of this document, the term service refers to a physical appliance running on dedicated hardware, a virtualized service inside an isolated environment such as a Virtual Machine (VM), container or namespace, or any process running on a compute element. A service may also comprise multiple sub-components running in different processes or containers. Unless otherwise stated, this document does not make any assumption on the type or execution environment of a service.

The execution of a service can be integrated as part of an SR policy by assigning a segment identifier, or SID, to the service and including this service SID in the SR policy SID-list. Such a service SID may be of local or global significance. In the former case, other segments, such as prefix or adjacency segments, can be used to steer the traffic up to the node where the service segment is instantiated. In the latter case, the service is directly reachable from anywhere in the routing domain. This is realized with SR-MPLS by assigning a SID from the global label block ([I-D.ietf-spring-segment-routing-mpls]), or with SRv6 by advertising the SID locator in the routing protocol ([I-D.ietf-spring-srv6-network-programming]). It is up to the network operator to define the scope and reachability of each service SID. This decision can be based on various considerations such as infrastructure dynamicity, available control plane or orchestration system capabilities.

This document categorizes services in two types, depending on whether they are able to behave properly in the presence of SR information or not. These are respectively named SR-aware and SR-unaware services.

4.1. SR-Aware Services

An SR-aware service can process the SR information in the packets it receives. This means being able to identify the active segment as a local instruction and move forward in the segment list, but also that the service's own behavior is not hindered due to the presence of SR information. For example, an SR-aware firewall filtering SRv6 traffic based on its final destination must retrieve that information from the last entry in the SRH rather than the Destination Address field of the IPv6 header.

An SR-aware service is associated with a locally instantiated service segment, which is used to steer traffic through it.

If the service is configured to intercept all the packets passing through the appliance, the underlying routing system only has to

implement a default SR endpoint behavior (e.g., SR-MPLS node segment or SRv6 End behavior), and the corresponding SID will be used to steer traffic through the service.

If the service requires the packets to be directed to a specific virtual interface, networking queue or process, a dedicated SR behavior may be required to steer the packets to the appropriate location. The definition of such service-specific functions is out of the scope of this document.

SR-aware services also enable advanced network programming functionalities such as conditional branching and jumping to arbitrary SIDs in the segment list. In addition, SRv6 provides several ways of passing and exchanging information between services (e.g., SID arguments, tag field and TLVs). An example scenario involving these features is described in [IFIP18], which discusses the implementation of an SR-aware Intrusion Detection System.

Examples of SR-aware services are provided in section Section 8.1.

4.2. SR-Unaware Services

Any service that does not meet the above criteria for SR-awareness is considered as SR-unaware.

An SR-unaware service is not able to process the SR information in the traffic that it receives. It may either drop the traffic or take erroneous decisions due to the unrecognized routing information. In order to include such services in an SR policy, it is thus required to remove the SR information as well as any other encapsulation header before the service receives the packet, or to alter it in such a way that the service can correctly process the packet.

In this document, we define the concept of an SR proxy as an entity, separate from the service, that performs these modifications and handle the SR processing on behalf of a service. The SR proxy can run as a separate process on the service appliance, on a virtual switch or router on the compute node or on a different host.

An SR-unaware service is associated with a service segment instantiated on the SR proxy, which is used to steer traffic through the service. Section 6 describes several SR proxy behaviors to handle the encapsulation headers and SR information under various circumstances.

5. SR Service Policies

An SR service policy is an SR policy, as defined in [I-D.ietf-spring-segment-routing-policy], that includes at least one service. This service is represented in the SID-list by its associated service SID. In case the policy should include several services, the service traversal order is indicated by the relative position of each service SID in the SID-list. Using the mechanisms described in [I-D.ietf-spring-segment-routing-policy], it is possible to load balance the traffic over several services, or instances of the same service, by associating with the SR service policy a weighted set of SID-lists, each containing a possible sequence of service SIDs to be traversed. Similarly, several candidate paths can be specified for the SR service policy, each with its own set of SID-lists, for resiliency purposes.

Furthermore, binding SIDs (BSIDs) can be leveraged in the context of service policies to reduce the number of SIDs imposed by the headend, provide opacity between domains and improve scalability, as described in [I-D.filsfils-spring-sr-policy-considerations]. For example, a network operator may want a policy in its core domain to include services that are running in one of its datacenters. One option is to define an SR policy at ingress edge of the core domain that explicitly includes all the SIDs needed to steer the traffic through the core and in the DC, but that may result in a long SID-list and requires to update the ingress edge configuration every time the DC part of the policy is modified. Alternatively, a separate policy can be defined at the ingress edge of the datacenter with only the SIDs that needs to be executed there and its BSID included in the core domain policy. That BSID remains stable when the DC policy is modified and can even be shared among several core domain policies that would require the same type of processing in the DC.

This section describes how services can be integrated within an SR-MPLS or SRv6 service policy.

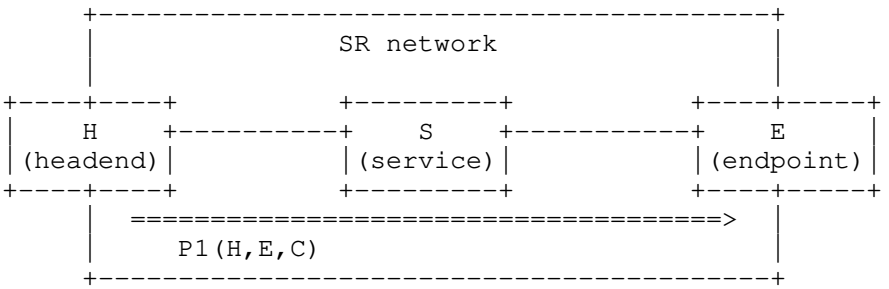


Figure 1: SR service policy

Figure 1 illustrates a basic SR service policy instantiated on a headend node H towards an endpoint E and traversing a service S. The SR policy may also include additional requirements, such as traffic engineering or VPN. On the head-end H, the SR policy P1 is created with a color C and endpoint E and associated with an SR path that can either be explicitly configured, dynamically computed on H or provisioned by a network controller.

In its most basic form, the SR policy P1 would be resolved into the SID-list < SID(S), SID(E) >. This is assuming that SID(S) and SID(E) are directly reachable from H and S, respectively, and that the forwarding path meets the policy requirement. However, depending on the dataplane and the segments available in the network, additional SIDs may be required to enforce the SR policy.

This model applies regardless of the SR-awareness of the service. If it is SR-unaware, then S simply represents the proxy that takes care of transmitting the packet to the actual service.

Traffic can then be steered into this policy using any of the mechanisms described in [I-D.ietf-spring-segment-routing-policy].

The following subsections describe the specificities of each SR dataplane.

5.1. SR-MPLS Data Plane

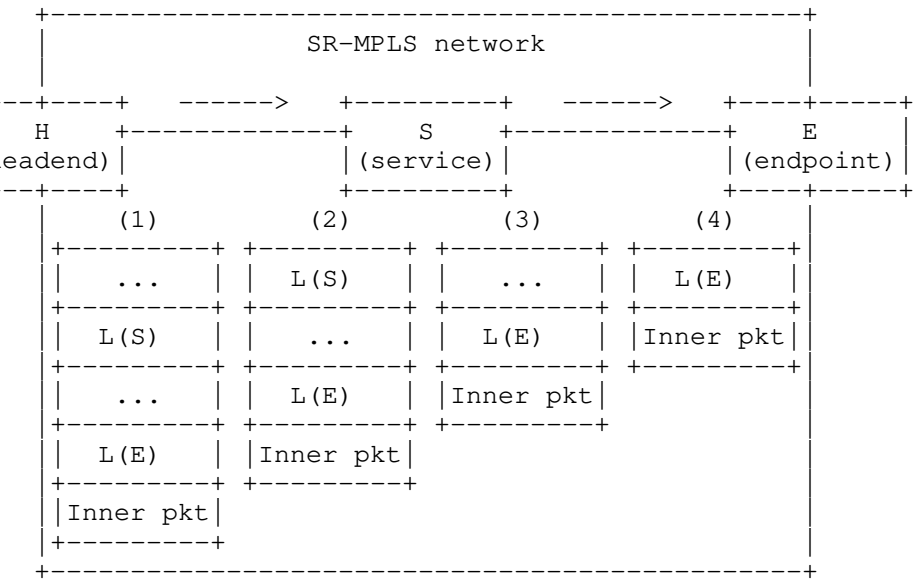


Figure 2: Packet walk in an SR-MPLS network

In an SR-MPLS network, the SR policy SID-list is encoded as a stack of MPLS labels[I-D.ietf-spring-segment-routing-mpls] and pushed on top of the packet.

In the example shown on Figure 2, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an MPLS label stack that includes at least a label L(S) associated to service S and a label L(E) associated to the endpoint E. The label stack may also include additional intermediate SIDs if these are required for traffic engineering (e.g., to encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID L(S) may be taken from the global or local SID block of node S and, in the latter case, one or more SIDs might be needed before L(S) in order for the packet to reach node S (e.g., a prefix-SID of S), where L(S) can be interpreted. The same applies for the SID L(E) at the SR policy endpoint.

Special consideration must be taken into account when using Local SIDs for service identification due to increased label stack depth and the associated impacts.

When the packet arrives at S, this node determines the MPLS payload type and the appropriate behavior for processing the packet based on the semantic locally associated to the top label L(S). If S is an

SR-aware service, the SID L(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, L(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, L(S) is also popped from the label stack in order to expose the next SID, which may be L(E) or another intermediate SID.

5.2. SRv6 Data Plane

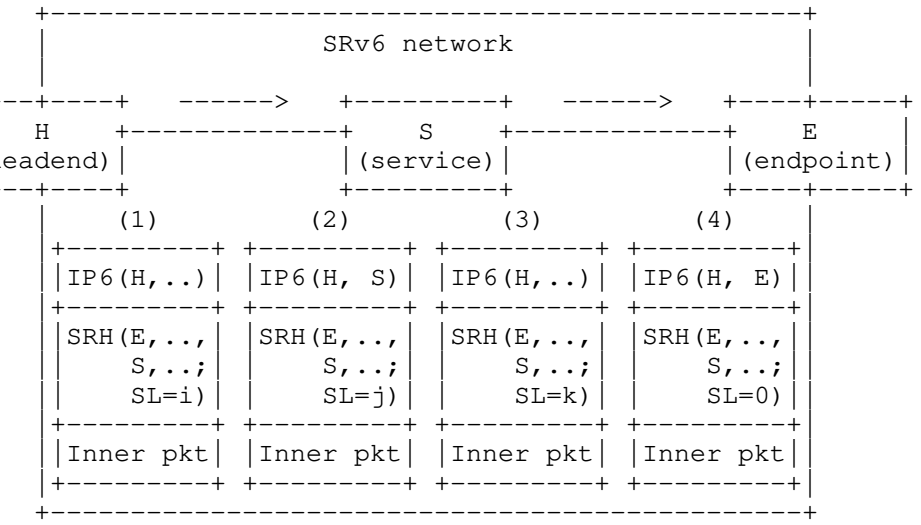


Figure 3: Packet walk in an SRv6 network

In an SRv6 network, the SR Policy is encoded into the packet as an IPv6 header possibly followed by a Segment Routing Header (SRH) [I-D.ietf-6man-segment-routing-header].

In the example shown on Figure 3, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into Segment-List that includes at least a segment SID(S) to the service, or service proxy, S and a segment SID(E) to the endpoint E. The Segment-List may also include additional intermediate SIDs if these are required for traffic engineering (e.g., the encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID locator may or may not be advertised in the routing protocol and, in the latter case, one or more SIDs might be needed before SID(S) in order to bring the packet up to node S, where SID(S) can be

interpreted. The same applies for the segment SID(E) at the SR policy endpoint.

When the packet arrives at S, this node determines how to process the packet based on the semantic locally associated to the active segment SID(S). If S is an SR-aware service, then SID(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, SID(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, the SRv6 End function is also applied in order to make the next SID, which may be SID(E) or another intermediate SID, active.

The "Inner pkt" on Figure 3 represents the SRv6 payload, which may be an encapsulated IP packet, an Ethernet frame or a transport-layer payload, for example.

6. SR Proxy Behaviors

This section describes several SR proxy behaviors designed to enable SR service programming through SR-unaware services. A system implementing one of these behaviors may handle the SR processing on behalf of an SR-unaware service and allows the service to properly process the traffic that is steered through it.

A service may be located at any hop in an SR policy, including the last segment. However, the SR proxy behaviors defined in this section are dedicated to supporting SR-unaware services at intermediate hops in the segment list. In case an SR-unaware service is at the last segment, it is sufficient to ensure that the SR information is ignored (IPv6 routing extension header with Segments Left equal to 0) or removed before the packet reaches the service (MPLS PHP, SRv6 decapsulation behavior or PSP flavor).

As illustrated on Figure 4, the generic behavior of an SR proxy has two parts. The first part is in charge of passing traffic from the network to the service. It intercepts the SR traffic destined for the service via a locally instantiated service segment, modifies it in such a way that it appears as non-SR traffic to the service, then sends it out on a given interface, IFACE-OUT, connected to the service. The second part receives the traffic coming back from the service on IFACE-IN, restores the SR information and forwards it according to the next segment in the list. IFACE-OUT and IFACE-IN are respectively the proxy interface used for sending traffic to the service and the proxy interface that receives the traffic coming back from the service. These can be physical interfaces or sub-interfaces

(VLANs) and, unless otherwise stated, IFACE-OUT and IFACE-IN can represent the same interface.

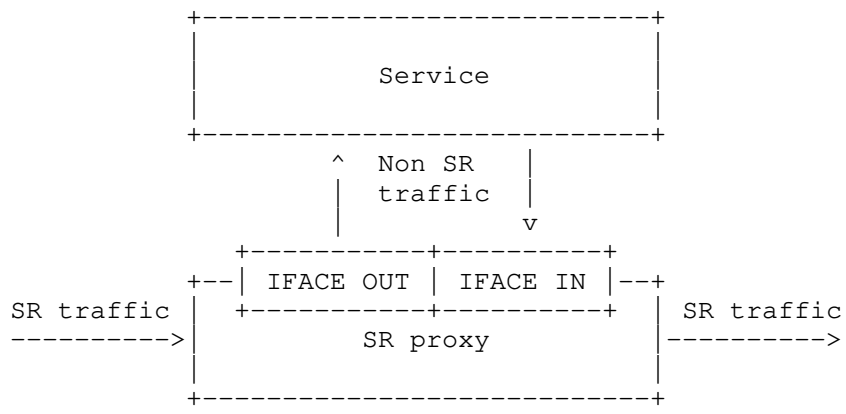


Figure 4: Generic SR proxy

In the next subsections, the following SR proxy mechanisms are defined:

- Static proxy
- Dynamic proxy
- Shared-memory proxy
- Masquerading proxy

Each mechanism has its own characteristics and constraints, which are summarized in the below table. It is up to the operator to select the best one based on the proxy node capabilities, the service behavior and the traffic type. It is also possible to use different proxy mechanisms within the same service policy.

		Static	Dynamic	Shared mem.	Masquerading
SR flavors	SR-MPLS	Y	Y	Y	-
	Inline SRv6	P	P	P	Y
	SRv6 encapsulation	Y	Y	Y	-
Chain agnostic configuration		N	N	Y	Y
Transparent to chain changes		N	Y	Y	Y
Service support	DA modification	Y	Y	Y	NAT
	Payload modification	Y	Y	Y	Y
	Packet generation	Y	Y	cache	cache
	Packet deletion	Y	Y	Y	Y
	Packet re-ordering	Y	Y	Y	Y
	Transport endpoint	Y	Y	cache	cache
Supported traffic	Ethernet	Y	Y	Y	-
	IPv4	Y	Y	Y	-
	IPv6	Y	Y	Y	Y

Figure 5: SR proxy summary

Note: The use of a shared memory proxy requires both the service (VNF) and the proxy to be running on the same node.

6.1. Static SR Proxy

The static proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an MPLS label stack or an IPv6 header on top of an inner packet, which can be Ethernet, IPv4 or IPv6.

A static SR proxy segment is associated with the following mandatory parameters

- INNER-TYPE: Inner packet type
- NH-ADDR: Next hop Ethernet address (only for inner type IPv4 and IPv6)
- IFACE-OUT: Local interface for sending traffic towards the service
- IFACE-IN: Local interface receiving the traffic coming back from the service
- CACHE: SR information to be attached on the traffic coming back from the service, including at least
  - CACHE.SA: IPv6 source address (SRv6 only)
  - CACHE.LIST: Segment list expressed as MPLS labels or IPv6 address

A static SR proxy segment is thus defined for a specific service, inner packet type and cached SR information. It is also bound to a pair of directed interfaces on the proxy. These may be both directions of a single interface, or opposite directions of two different interfaces. The latter is recommended in case the service is to be used as part of a bi-directional SR service policy. If the proxy and the service both support 802.1Q, IFACE-OUT and IFACE-IN can also represent sub-interfaces.

The first part of this behavior is triggered when the proxy node receives a packet whose active segment matches a segment associated with the static proxy behavior. It removes the SR information from the packet then sends it on a specific interface towards the associated service. This SR information corresponds to the full label stack for SR-MPLS or to the encapsulation IPv6 header with any attached extension header in the case of SRv6.

The second part is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This

policy attaches to the incoming traffic the cached SR information associated with the SR proxy segment. If the proxy segment uses the SR-MPLS data plane, CACHE contains a stack of labels to be pushed on top of the packets. With the SRv6 data plane, CACHE is defined as a source address, an active segment and an optional SRH (tag, segments left, segment list and metadata). The proxy encapsulates the packets with an IPv6 header that has the source address, the active segment as destination address and the SRH as a routing extension header. After the SR information has been attached, the packets are forwarded according to the active segment, which is represented by the top MPLS label or the IPv6 Destination Address. An MPLS TTL or IPv6 Hop Limit value may also be configured in CACHE. If it is not, the proxy should set these values according to the node's default setting for MPLS or IPv6 encapsulation.

In this scenario, there are no restrictions on the operations that can be performed by the service on the stream of packets. It may operate at all protocol layers, terminate transport layer connections, generate new packets and initiate transport layer connections. This behavior may also be used to integrate an IPv4-only service into an SRv6 policy. However, a static SR proxy segment can be used in only one service policy at a time. As opposed to most other segment types, a static SR proxy segment is bound to a unique list of segments, which represents a directed SR service policy. This is due to the cached SR information being defined in the segment configuration. This limitation only prevents multiple segment lists from using the same static SR proxy segment at the same time, but a single segment list can be shared by any number of traffic flows. Besides, since the returning traffic from the service is re-classified based on the incoming interface, an interface can be used as receiving interface (IFACE-IN) only for a single SR proxy segment at a time. In the case of a bi-directional SR service policy, a different SR proxy segment and receiving interface are required for the return direction.

The static proxy behavior may also be used for sending traffic through "bump in the wire" services that are transparent to the IP and Ethernet layers. This type of processing is assumed when the inner traffic type is Ethernet, since the original destination address of the Ethernet frame is preserved when the packet is steered into the SR Policy and likely associated with a node downstream of the policy tail-end. In case the inner type is IP (IPv4 or IPv6), the NH-ADDR parameter may be set to a dummy or broadcast Ethernet address, or simply to the address of the proxy receiving interface (IFACE-IN).

6.1.1. SR-MPLS Pseudocode

6.1.1.1. Static Proxy for Inner Type Ethernet

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the frame to the Ethernet module for transmission via
     interface IFACE-OUT.
```

Figure 6: SID processing for MPLS static proxy (Ethernet)

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Remove the preamble or Frame Check Sequence (FCS).
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 7: Inbound policy for MPLS static proxy (Ethernet)

6.1.1.2. Static Proxy for Inner Type IPv4

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 8: SID processing for MPLS static proxy (IPv4)

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the TTL and adjust the checksum accordingly.
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 9: Inbound policy for MPLS static proxy (IPv4)

6.1.1.3. Static Proxy for Inner Type IPv6

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the packet to the IPv6 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 10: SID processing for MPLS static proxy (IPv6)

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the Hop Limit.
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 11: Inbound policy for MPLS static proxy (IPv6)

6.1.2. SRv6 Pseudocode

6.1.2.1. Static Proxy for Inner Type Ethernet

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[Segments Left] from the SRH to the
       Destination Address of the IPv6 header.
S15.   If (Upper-layer header type != 143 (Ethernet)) {
S16.     Resubmit the packet to the IPv6 module for transmission to
         the new destination.
S17.   }
S18.   Perform IPv6 decapsulation.
S19.   Submit the frame to the Ethernet module for transmission via
       interface IFACE-OUT.
S20. }
```

Figure 12: SID processing for SRv6 static proxy (Ethernet)

S15: 143 (Ethernet) refers to the value assigned by IANA for "Ethernet" in the "Internet Protocol Numbers" registry.

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 143 (Ethernet)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the frame to the Ethernet module for transmission via
     interface IFACE-OUT.
```

Figure 13: Upper-layer header processing for SRv6 static proxy (Ethernet)

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Remove the preamble or Frame Check Sequence (FCS).
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 143 (Ethernet),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 14: Inbound policy for SRv6 static proxy (Ethernet)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the IPv6 header set to 143 (Ethernet).

6.1.2.2. Static Proxy for Inner Type IPv4

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
  S01. When an SRH is processed {
  S02.   If (Segments Left == 0) {
  S03.     Proceed to process the next header in the packet.
  S04.   }
  S05.   If (IPv6 Hop Limit <= 1) {
  S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (hop limit exceeded in transit),
           Interrupt packet processing and discard the packet.
  S07.   }
  S08.   max_last_entry = (Hdr Ext Len / 2) - 1
  S09.   If ((Last Entry > max_last_entry) or
              (Segments Left > (Last Entry + 1))) {
  S10.     Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
  S11.   }
  S12.   Decrement Hop Limit by 1.
  S13.   Decrement Segments Left by 1.
  S14.   Copy Segment List[Segments Left] from the SRH to the
         Destination Address of the IPv6 header.
  S15.   If (Upper-layer header type != 4 (IPv4)) {
  S16.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
  S17.   }
  S18.   Perform IPv6 decapsulation.
  S19.   Submit the packet to the IPv4 module for transmission on
         interface IFACE-OUT via NH-ADDR.
  S20. }
```

Figure 15: SID processing for SRv6 static proxy (IPv4)

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 4 (IPv4)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 16: Upper-layer header processing for SRv6 static proxy (IPv4)

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the TTL and adjust the checksum accordingly.
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 4 (IPv4),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 17: Inbound policy for SRv6 static proxy (IPv4)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the IPv6 header set to 4 (IPv4).

6.1.2.3. Static Proxy for Inner Type IPv6

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
  S01. When an SRH is processed {
  S02.   If (Segments Left == 0) {
  S03.     Proceed to process the next header in the packet.
  S04.   }
  S05.   If (IPv6 Hop Limit <= 1) {
  S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (hop limit exceeded in transit),
           Interrupt packet processing and discard the packet.
  S07.   }
  S08.   max_last_entry = (Hdr Ext Len / 2) - 1
  S09.   If ((Last Entry > max_last_entry) or
              (Segments Left > (Last Entry + 1))) {
  S10.     Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
  S11.   }
  S12.   Decrement Hop Limit by 1.
  S13.   Decrement Segments Left by 1.
  S14.   Copy Segment List[Segments Left] from the SRH to the
         Destination Address of the IPv6 header.
  S15.   If (Upper-layer header type != 41 (IPv6)) {
  S16.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
  S17.   }
  S18.   Perform IPv6 decapsulation.
  S19.   Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
  S20. }
```

Figure 18: SID processing for SRv6 static proxy (IPv6)

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 41 (IPv6)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the packet to the IPv6 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 19: Upper-layer header processing for SRv6 static proxy (IPv6)

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the Hop Limit.
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 41 (IPv6),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 20: Inbound policy for SRv6 static proxy (IPv6)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the (outer) IPv6 header set to 41 (IPv6).

6.2. Dynamic SR Proxy

The dynamic proxy is an improvement over the static proxy that dynamically learns the SR information before removing it from the incoming traffic. The same information can then be re-attached to the traffic returning from the service. As opposed to the static SR proxy, no CACHE information needs to be configured. Instead, the dynamic SR proxy relies on a local caching mechanism on the node instantiating this segment.

Upon receiving a packet whose active segment matches a dynamic SR proxy function, the proxy node pops the top MPLS label or applies the SRv6 End behavior, then compares the updated SR information with the cache entry for the current segment. If the cache is empty or different, it is updated with the new SR information. The SR information is then removed and the inner packet is sent towards the service.

The cache entry is not mapped to any particular packet, but instead to an SR service policy identified by the receiving interface (IFACE-IN). Any non-link-local IP packet or non-local Ethernet frame received on that interface will be re-encapsulated with the cached headers as described in Section 6.1. The service may thus drop, modify or generate new packets without affecting the proxy.

6.2.1. SR-MPLS Pseudocode

The dynamic proxy SR-MPLS pseudocode is obtained by inserting the following instructions at the beginning of the static SR-MPLS pseudocode (Section 6.1.1).

```
S01. If the top label S bit is different from 0 {
S02.   Discard the packet.
S03. }
S04. POP the top label.
S05. Copy the IPv6 encapsulation in a CACHE entry associated with the
     interface IFACE-IN.
```

Figure 21: SID processing for MPLS dynamic proxy

S01: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

S05: An implementation may optimize the caching procedure by copying information into the cache only if it is different from the current content of the cache entry. Furthermore, a TTL margin can be configured for the top label stack entry to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a TTL difference smaller than the configured margin should not trigger a cache update (provided that the labels are the same).

When processing an Ethernet frame, an IPv4 packet or an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the pseudocode reported in Figure 7, Figure 9 or Figure 11, respectively, is executed.

6.2.2. SRv6 Pseudocode

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 dynamic proxy SID, the same pseudocode as described in Figure 12, Figure 15 and Figure 18, respectively for Ethernet, IPv4 and IPv6 traffic, is executed with the following addition between lines S17 and S18.

```
(... S17.     })
S17.1.   Copy the IPv6 encapsulation in a CACHE entry associated with
         the interface IFACE-IN.
(S18.     Perform IPv6 decapsulation...)
```

Figure 22: SID processing for SRv6 dynamic proxy

An implementation may optimize the caching procedure by copying information into the cache only if it is different from the current content of the cache entry. A Hop Limit margin can be configured to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a Hop Limit difference smaller than the configured margin should not trigger a cache update. Similarly, the Flow Label value can be ignored when comparing the current packet IPv6 header with the cache entry. In this case, the Flow Label should be re-computed by the proxy node when it restores the IPv6 encapsulation from the cache entry.

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 dynamic proxy SID, process the packet as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1.

When processing an Ethernet frame, an IPv4 packet or an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the same pseudocode as in Figure 14, Figure 17 or Figure 20, respectively, is executed.

6.3. Shared Memory SR Proxy

The shared memory proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy behavior leverages a shared-memory interface with a virtualized service (VNF) in order to hide the SR information from an SR-unaware service while keeping it attached to the packet. We assume in this case that the proxy and the VNF are running on the same compute node. A typical scenario is an SR-capable vrouter running on a container host and forwarding traffic to VNFs isolated within their respective container.

6.4. Masquerading SR Proxy

The masquerading proxy is an SR endpoint behavior for processing SRv6 traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an IPv6 header and an SRH on top of an inner payload. The masquerading behavior is independent from the inner payload type. Hence, the inner payload can be of any type but it is usually expected to be a transport layer packet, such as TCP or UDP.

A masquerading SR proxy segment is associated with the following mandatory parameters:

- NH-ADDR: Next hop Ethernet address

- IFACE-OUT: Local interface for sending traffic towards the service
- IFACE-IN: Local interface receiving the traffic coming back from the service

A masquerading SR proxy segment is thus defined for a specific service and bound to a pair of directed interfaces or sub-interfaces on the proxy. As opposed to the static and dynamic SR proxies, a masquerading segment can be present at the same time in any number of SR service policies and the same interfaces can be bound to multiple masquerading proxy segments. The only restriction is that a masquerading proxy segment cannot be the last segment in an SR service policy.

The first part of the masquerading behavior is triggered when the proxy node receives an IPv6 packet whose Destination Address matches a masquerading proxy SID. The proxy inspects the IPv6 extension headers and substitutes the Destination Address with the last SID in the SRH attached to the IPv6 header, which represents the final destination of the IPv6 packet. The packet is then sent out towards the service.

The service receives an IPv6 packet whose source and destination addresses are respectively the original source and final destination. It does not attempt to inspect the SRH, as RFC8200 specifies that routing extension headers are not examined or processed by transit nodes. Instead, the service simply forwards the packet based on its current Destination Address. In this scenario, we assume that the service can only inspect, drop or perform limited changes to the packets. For example, Intrusion Detection Systems, Deep Packet Inspectors and non-NAT Firewalls are among the services that can be supported by a masquerading SR proxy. Flavors of the masquerading behavior are defined in Section 6.4.2 and Section 6.4.3 to support a wider range of services.

The second part of the masquerading behavior, also called de-masquerading, is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This policy inspects the incoming traffic and triggers a regular SRv6 endpoint processing (End) on any IPv6 packet that contains an SRH. This processing occurs before any lookup on the packet Destination Address is performed and it is sufficient to restore the right active SID as the Destination Address of the IPv6 packet.

6.4.1. SRv6 Masquerading Proxy Pseudocode

Masquerading: When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 masquerading proxy SID, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[0] from the SRH to the Destination Address
       of the IPv6 header.
S15.   Submit the packet to the IPv6 module for transmission on
       interface IFACE-OUT via NH-ADDR.
S16. }
```

Figure 23: SID processing for SRv6 masquerading proxy

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 masquerading proxy SID, process the packet as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1.

De-masquerading: When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
S10.     Copy Segment List[Segments Left] from the SRH to the
         Destination Address of the IPv6 header.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

Figure 24: Inbound policy for SRv6 masquerading proxy

6.4.2. Destination NAT Flavor

Services modifying the destination address in the packets they process, such as NATs, can be supported by reporting the updated Destination Address back into the Segment List field of the SRH.

The Destination NAT flavor of the SRv6 masquerading proxy is enabled by adding the following instruction between lines S09 and S10 of the de-masquerading pseudocode in Figure 24.

```
(... S09.     })
S09.1.   Copy the Destination Address of the IPv6 header to the
         Segment List[0] entry of the SRH.
(S10.     Copy Segment List[Segments Left] from the SRH to the
          Destination Address of the IPv6 header...)
```

6.4.3. Caching Flavor

Services generating packets or acting as endpoints for transport connections can be supported by adding a dynamic caching mechanism similar to the one described in Section 6.2.

The caching flavor of the SRv6 masquerading proxy is enabled by:

- Adding the following instruction between lines S14 and S15 of the masquerading pseudocode in Figure 23.

```
(... S14.   Copy Segment List[0] from the SRH to the Destination
            Address of the IPv6 header.
S14.1. Copy the IPv6 encapsulation in a CACHE entry associated with
       the interface IFACE-IN.
(S15.   Submit the packet to the IPv6 module for transmission on
        interface IFACE-OUT via NH-ADDR.)
```

- Updating the de-masquerading pseudocode such that, in addition to the SRH processing in Figure 24, the following pseudocode is executed when processing an IPv6 packet (received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN) that does not contain an SRH.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   If (IPv6 Hop Limit <= 1) {
S04.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S05.   }
S06.   Decrement Hop Limit by 1.
S07.   Update the IPv6 encapsulation according to the retrieved CACHE
       entry.
S08.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S09. }
```

7. Metadata

7.1. MPLS Data Plane

Metadata can be carried for SR-MPLS traffic in a Segment Routing Header inserted between the last MPLS label and the MPLS payload. When used solely as a metadata container, the SRH does not carry any segment but only the mandatory header fields, including the tag and flags, and any TLVs that is required for transporting the metadata.

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata in an MPLS packet is a potential issue to be addressed. One possible solution is to add the indication about the presence of metadata in the semantic of the SIDs. Note that only the SIDs whose behavior involves looking at the metadata or the MPLS payload would need to include such semantic (e.g., service segments). Other segments, such as topological

segments, are not affected by the presence of metadata. Another, more generic, solution is to introduce a protocol identifier field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

7.2. IPv6 Data Plane

7.2.1. SRH TLV Objects

The IPv6 SRH TLV objects are designed to carry all sorts of metadata. TLV objects can be imposed by the ingress edge router that steers the traffic into the SR service policy.

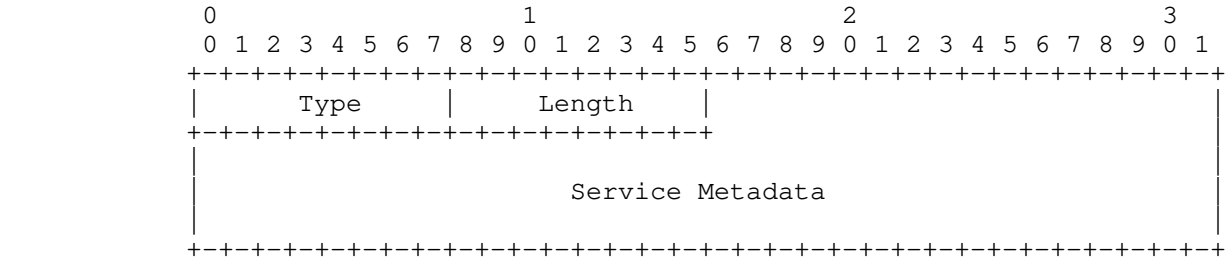
An SR-aware service may impose, modify or remove any TLV object attached to the first SRH, either by directly modifying the packet headers or via a control channel between the service and its forwarding plane.

An SR-aware service that re-classifies the traffic and steers it into a new SR service policy (e.g. DPI) may attach any TLV object to the new SRH.

Metadata imposition and handling will be further discussed in a future version of this document.

7.2.1.1. Opaque Metadata TLV

This document defines an SRv6 TLV called Opaque Metadata TLV. This is a fixed-length container to carry any type of Service Metadata. No assumption is made by this document on the structure or the content of the carried metadata. The Opaque Metadata TLV has the following format:



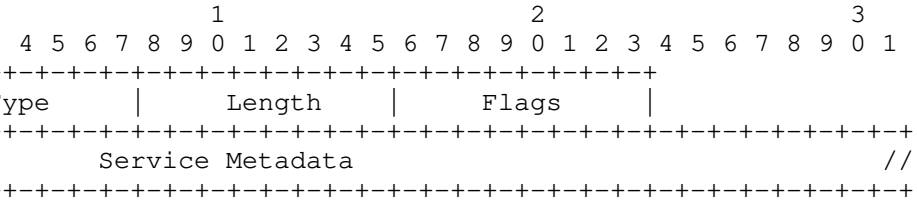
where:

- Type: to be assigned by IANA.
- Length: 14.

o Service Metadata: 14 octets of opaque data.

7.2.1.2. NSH Carrier TLV

This document defines an SRv6 TLV called NSH Carrier TLV. It is a container to carry Service Metadata in the form of Variable-Length Metadata as defined in [RFC8300] for NSH MD Type 2. The NSH Carrier TLV has the following format:



where:

o Type: to be assigned by IANA.

o Length: the total length of the TLV.

o Flags: 8 bits. No flags are defined in this document. SHOULD be set to 0 on transmission and MUST be ignored on receipt.

o Service Metadata: a list of Service Metadata TLV as defined in [RFC8300] for NSH MD Type 2.

7.2.2. SRH Tag

The SRH tag identifies a packet as part of a group or class of packets [I-D.ietf-6man-segment-routing-header].

In the context of service programming, this field can be used to encode basic metadata in the SRH. An example use-case is to leverage the SRH tag to encode a policy ID. This policy ID can then be used by an SR-aware function to identify a particular processing policy to be applied on that packet.

8. Implementation Status

This section is to be removed prior to publishing as an RFC.

8.1. SR-Aware Services

Specific SRv6 support has been implemented for the below open-source services:

- Iptables (1.6.2 and later) [IPTABLES]
- Nftables (0.8.4 and later) [NFTABLES]
- Snort [SNORT]

In addition, any service relying on the Linux kernel, version 4.10 and later, or FD.io VPP for packet forwarding can be considered as SR-aware.

8.2. Proxy Behaviors

The static SR proxy is available for SR-MPLS and SRv6 on various Cisco hardware and software platforms. Furthermore, the following proxies are available on open-source software.

		VPP	Linux
MPLS	Static proxy	Available	In progress
	Dynamic proxy	In progress	In progress
	Shared memory proxy	In progress	In progress
SRv6	Static proxy	Available	In progress
	Dynamic proxy	Available	Available
	Shared memory proxy	In progress	In progress
	Masquerading proxy	Available	Available

Figure 25: Open-source implementation status table

9. Related Works

The Segment Routing solution addresses a wide problem that covers both topological and service policies. The topological and service instructions can be either deployed in isolation or in combination. SR has thus a wider applicability than the architecture defined in [RFC7665]. Furthermore, the inherent property of SR is a stateless

network fabric. In SR, there is no state within the fabric to recognize a flow and associate it with a policy. State is only present at the ingress edge of the SR domain, where the policy is encoded into the packets. This is completely different from other proposals such as [RFC8300] and the MPLS label swapping mechanism described in [I-D.ietf-mpls-sfc], which rely on state configured at every hop of the service chain.

10. IANA Considerations

10.1. SRv6 Endpoint Behaviors

This I-D requests the IANA to allocate, within the "SRv6 Endpoint Behaviors" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry, the following allocations:

Value	Description	Reference
TBA1-1	End.AN - SR-aware function (native)	[This.ID]
TBA1-2	End.AS - Static proxy	[This.ID]
TBA1-3	End.AD - Dynamic proxy	[This.ID]
TBA1-4	End.AM - Masquerading proxy	[This.ID]
TBA1-5	End.AM - Masquerading proxy with NAT	[This.ID]
TBA1-6	End.AM - Masquerading proxy with Caching	[This.ID]
TBA1-7	End.AM - Masquerading proxy with NAT & Caching	[This.ID]

10.2. Segment Routing Header TLVs

This I-D requests the IANA to allocate, within the "Segment Routing Header TLVs" registry, the following allocations:

Value	Description	Reference
TBA2-1	Opaque Metadata TLV	[This.ID]
TBA2-2	NSH Carrier TLV	[This.ID]

11. Security Considerations

The security requirements and mechanisms described in [RFC8402], [I-D.ietf-6man-segment-routing-header] and [I-D.ietf-spring-srv6-network-programming] also apply to this document.

This document does not introduce any new security vulnerabilities.

12. Acknowledgements

The authors would like to thank Thierry Couture, Ketan Talaulikar, Loa Andersson, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein and Joel M. Halpern for their valuable comments and suggestions on the document.

13. Contributors

The following people have contributed to this document:

Pablo Camarillo
Cisco Systems, Inc.
Spain

Email: pcamaril@cisco.com

Bart Peirens
Proximus
Belgium

Email: bart.peirens@proximus.com

Dirk Steinberg
Lapishills Consulting Limited
Cyprus

Email: dirk@lapishills.com

Ahmed AbdelSalam
Cisco Systems, Inc.
Italy

Email: ahabdels@cisco.com

Gaurav Dawra
LinkedIn
United States of America

Email: gdawra@linkedin.com

Stewart Bryant
Futurewei Technologies Inc

Email: stewart.bryant@gmail.com

Hamid Assarpour
Broadcom

Email: hamid.assarpour@broadcom.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura
Individual

Email: jefftant@gmail.com

Martin Vigoureux
Nokia

Email: martin.vigoureux@nokia.com

Jisu Bhattacharya
Cisco Systems, Inc.
United States of America

Email: jisu@cisco.com

14. References

14.1. Normative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

[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-06 (work in progress), December 2019.

[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-10 (work in progress), February 2020.

[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>.

14.2. Informative References

[I-D.dawra-idr-bgp-sr-service-chaining]
Dawra, G., Filsfils, C., daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Clad, F., and K. Talaulikar, "BGP Control Plane Extensions for Segment Routing based Service Chaining", draft-dawra-idr-bgp-sr-service-chaining-02 (work in progress), January 2018.

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-04 (work in progress), October 2019.

[I-D.ietf-mpls-sfc]
Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", draft-ietf-mpls-sfc-07 (work in progress), March 2019.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X., Assarpour, H., Ma, S., and F. Clad, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-06 (work in progress), March 2019.

[IFIP18] Abdelsalam, A., Salsano, S., Clad, F., Camarillo, P., and C. Filsfils, "SEgment Routing Aware Firewall For Service Function Chaining scenarios", IFIP Networking conference , May 2018.

[IPTABLES]
"iptables-1.6.2 changes", February 2018, <https://netfilter.org/projects/iptables/files/changes-iptables-1.6.2.txt>.

[NFTABLES]
"nftables-0.8.4 changes", May 2018, <https://netfilter.org/projects/nftables/files/changes-nftables-0.8.4.txt>.

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

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

[SNORT] "SR-Snort", March 2018, <https://github.com/SRouting/SR-Snort>.

Authors' Addresses

Francois Clad (editor)
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Xiaohu Xu (editor)
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Cheng Li
Huawei

Email: chengli13@huawei.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Shaowen Ma
Mellanox

Email: mashaowen@gmail.com

Chaitanya Yadlapalli
AT&T
USA

Email: cy098d@att.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

SPRING
Internet-Draft
Intended status: Standards Track
Expires: March 12, 2021

F. Clad, Ed.
Cisco Systems, Inc.
X. Xu, Ed.
Alibaba
C. Filsfils
Cisco Systems, Inc.
D. Bernier
Bell Canada
C. Li
Huawei
B. Decraene
Orange
S. Ma
Mellanox
C. Yadlapalli
AT&T
W. Henderickx
Nokia
S. Salsano
Universita di Roma "Tor Vergata"

September 08, 2020

Service Programming with Segment Routing
draft-ietf-spring-sr-service-programming-03

Abstract

This document defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IPv6 networks, as described in the Segment Routing architecture.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 12, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] is an architecture based on the source routing paradigm that seeks the right balance between distributed intelligence and centralized programmability. SR can be used with an MPLS or an IPv6 data plane to steer packets through an ordered list of instructions, called segments. These segments may encode simple routing instructions for forwarding packets along a specific network path, but also steer them through Virtual Network Functions (VNFs) or physical service appliances available in the network.

In an SR network, each of these services, running either on a physical appliance or in a virtual environment, are associated with a segment identifier (SID). These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined together in a SID-list to achieve service programming, but also with other types of segments as defined in [RFC8402]. SR thus provides a fully integrated solution for overlay, underlay and service programming. Furthermore, the IPv6 instantiation of SR (SRv6) [I-D.ietf-spring-srv6-network-programming] supports metadata transportation in the Segment Routing Header [I-D.ietf-6man-segment-routing-header], either natively in the tag field or with extensions such as TLVs.

This document describes how a service can be associated with a SID, including legacy services with no SR capabilities, and how these service SIDs are integrated within an SR policy. The definition of an SR Policy and the traffic steering mechanisms are covered in [I-D.ietf-spring-segment-routing-policy] and hence outside the scope of this document.

The definition of control plane components, such as service segment discovery, is outside the scope of this data plane document. For reference, the option of using BGP extensions to support SR service programming is proposed in [I-D.dawra-idr-bgp-sr-service-chaining].

2. Terminology

This document leverages the terminology proposed in [RFC8402], [RFC8660], [I-D.ietf-6man-segment-routing-header], [I-D.ietf-spring-srv6-network-programming] and [I-D.ietf-spring-segment-routing-policy]. It also introduces the following new terms.

Service segment: A segment associated with a service. The service may either run on a physical appliance or in a virtual environment such as a virtual machine or container.

SR-aware service: A service that is fully capable of processing SR traffic. An SR-aware service can be directly associated with a service segment.

SR-unaware service: A service that is unable to process SR traffic or may behave incorrectly due to presence of SR information in the packet headers. An SR-unaware service can be associated with a service segment through an SR proxy function.

3. Classification and Steering

Classification and steering mechanisms are defined in section 8 of [I-D.ietf-spring-segment-routing-policy] and are independent from the purpose of the SR policy. From the perspective of a headend node classifying and steering traffic into an SR policy, there is no difference whether this policy contains IGP, BGP, peering, VPN or service segments, or any combination of these.

As documented in the above reference, traffic is classified when entering an SR domain. The SR policy headend may, depending on its capabilities, classify the packets on a per-destination basis, via simple FIB entries, or apply more complex policy routing rules requiring to look deeper into the packet. These rules are expected to support basic policy routing such as 5-tuple matching. In addition, the IPv6 SRH tag field defined in [I-D.ietf-6man-segment-routing-header] can be used to identify and classify packets sharing the same set of properties. Classified traffic is then steered into the appropriate SR policy and forwarded as per the SID-list(s) of the active candidate path.

SR traffic can be re-classified by an SR endpoint along the original SR policy (e.g., DPI service) or a transit node intercepting the traffic. This node is the head-end of a new SR policy that is imposed onto the packet, either as a stack of MPLS labels or as an IPv6 SRH.

4. Service Segments

In the context of this document, the term service refers to a physical appliance running on dedicated hardware, a virtualized service inside an isolated environment such as a Virtual Machine (VM), container or namespace, or any process running on a compute element. A service may also comprise multiple sub-components running in different processes or containers. Unless otherwise stated, this document does not make any assumption on the type or execution environment of a service.

The execution of a service can be integrated as part of an SR policy by assigning a segment identifier, or SID, to the service and including this service SID in the SR policy SID-list. Such a service SID may be of local or global significance. In the former case, other segments, such as prefix or adjacency segments, can be used to steer the traffic up to the node where the service segment is instantiated. In the latter case, the service is directly reachable from anywhere in the routing domain. This is realized with SR-MPLS by assigning a SID from the global label block ([I-D.ietf-spring-segment-routing-mpls]), or with SRv6 by advertising the SID locator in the routing protocol ([I-D.ietf-spring-srv6-network-programming]). It is up to the network operator to define the scope and reachability of each service SID. This decision can be based on various considerations such as infrastructure dynamicity, available control plane or orchestration system capabilities.

This document categorizes services in two types, depending on whether they are able to behave properly in the presence of SR information or not. These are respectively named SR-aware and SR-unaware services.

4.1. SR-Aware Services

An SR-aware service can process the SR information in the packets it receives. This means being able to identify the active segment as a local instruction and move forward in the segment list, but also that the service's own behavior is not hindered due to the presence of SR information. For example, an SR-aware firewall filtering SRv6 traffic based on its final destination must retrieve that information from the last entry in the SRH rather than the Destination Address field of the IPv6 header.

An SR-aware service is associated with a locally instantiated service segment, which is used to steer traffic through it.

If the service is configured to intercept all the packets passing through the appliance, the underlying routing system only has to

implement a default SR endpoint behavior (e.g., SR-MPLS node segment or SRv6 End behavior), and the corresponding SID will be used to steer traffic through the service.

If the service requires the packets to be directed to a specific virtual interface, networking queue or process, a dedicated SR behavior may be required to steer the packets to the appropriate location. The definition of such service-specific functions is out of the scope of this document.

SR-aware services also enable advanced network programming functionalities such as conditional branching and jumping to arbitrary SIDs in the segment list. In addition, SRv6 provides several ways of passing and exchanging information between services (e.g., SID arguments, tag field and TLVs). An example scenario involving these features is described in [IFIP18], which discusses the implementation of an SR-aware Intrusion Detection System.

Examples of SR-aware services are provided in section Section 8.1.

4.2. SR-Unaware Services

Any service that does not meet the above criteria for SR-awareness is considered as SR-unaware.

An SR-unaware service is not able to process the SR information in the traffic that it receives. It may either drop the traffic or take erroneous decisions due to the unrecognized routing information. In order to include such services in an SR policy, it is thus required to remove the SR information as well as any other encapsulation header before the service receives the packet, or to alter it in such a way that the service can correctly process the packet.

In this document, we define the concept of an SR proxy as an entity, separate from the service, that performs these modifications and handle the SR processing on behalf of a service. The SR proxy can run as a separate process on the service appliance, on a virtual switch or router on the compute node or on a different host.

An SR-unaware service is associated with a service segment instantiated on the SR proxy, which is used to steer traffic through the service. Section 6 describes several SR proxy behaviors to handle the encapsulation headers and SR information under various circumstances.

5. SR Service Policies

An SR service policy is an SR policy, as defined in [I-D.ietf-spring-segment-routing-policy], that includes at least one service. This service is represented in the SID-list by its associated service SID. In case the policy should include several services, the service traversal order is indicated by the relative position of each service SID in the SID-list. Using the mechanisms described in [I-D.ietf-spring-segment-routing-policy], it is possible to load balance the traffic over several services, or instances of the same service, by associating with the SR service policy a weighted set of SID-lists, each containing a possible sequence of service SIDs to be traversed. Similarly, several candidate paths can be specified for the SR service policy, each with its own set of SID-lists, for resiliency purposes.

Furthermore, binding SIDs (BSIDs) can be leveraged in the context of service policies to reduce the number of SIDs imposed by the headend, provide opacity between domains and improve scalability, as described in [I-D.filsfils-spring-sr-policy-considerations]. For example, a network operator may want a policy in its core domain to include services that are running in one of its datacenters. One option is to define an SR policy at ingress edge of the core domain that explicitly includes all the SIDs needed to steer the traffic through the core and in the DC, but that may result in a long SID-list and requires to update the ingress edge configuration every time the DC part of the policy is modified. Alternatively, a separate policy can be defined at the ingress edge of the datacenter with only the SIDs that needs to be executed there and its BSID included in the core domain policy. That BSID remains stable when the DC policy is modified and can even be shared among several core domain policies that would require the same type of processing in the DC.

This section describes how services can be integrated within an SR-MPLS or SRv6 service policy.

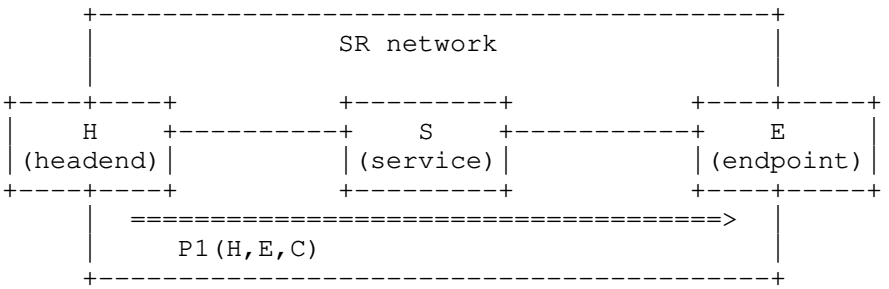


Figure 1: SR service policy

Figure 1 illustrates a basic SR service policy instantiated on a headend node H towards an endpoint E and traversing a service S. The SR policy may also include additional requirements, such as traffic engineering or VPN. On the head-end H, the SR policy P1 is created with a color C and endpoint E and associated with an SR path that can either be explicitly configured, dynamically computed on H or provisioned by a network controller.

In its most basic form, the SR policy P1 would be resolved into the SID-list < SID(S), SID(E) >. This is assuming that SID(S) and SID(E) are directly reachable from H and S, respectively, and that the forwarding path meets the policy requirement. However, depending on the dataplane and the segments available in the network, additional SIDs may be required to enforce the SR policy.

This model applies regardless of the SR-awareness of the service. If it is SR-unaware, then S simply represents the proxy that takes care of transmitting the packet to the actual service.

Traffic can then be steered into this policy using any of the mechanisms described in [I-D.ietf-spring-segment-routing-policy].

The following subsections describe the specificities of each SR dataplane.

5.1. SR-MPLS Data Plane

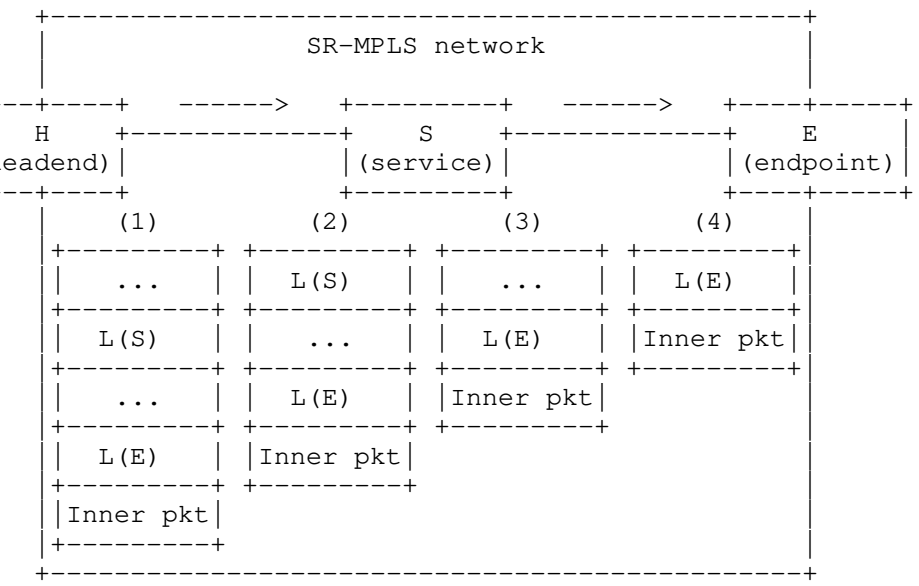


Figure 2: Packet walk in an SR-MPLS network

In an SR-MPLS network, the SR policy SID-list is encoded as a stack of MPLS labels[I-D.ietf-spring-segment-routing-mpls] and pushed on top of the packet.

In the example shown on Figure 2, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into an MPLS label stack that includes at least a label L(S) associated to service S and a label L(E) associated to the endpoint E. The label stack may also include additional intermediate SIDs if these are required for traffic engineering (e.g., to encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID L(S) may be taken from the global or local SID block of node S and, in the latter case, one or more SIDs might be needed before L(S) in order for the packet to reach node S (e.g., a prefix-SID of S), where L(S) can be interpreted. The same applies for the SID L(E) at the SR policy endpoint.

Special consideration must be taken into account when using Local SIDs for service identification due to increased label stack depth and the associated impacts.

When the packet arrives at S, this node determines the MPLS payload type and the appropriate behavior for processing the packet based on the semantic locally associated to the top label L(S). If S is an

SR-aware service, the SID L(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, L(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, L(S) is also popped from the label stack in order to expose the next SID, which may be L(E) or another intermediate SID.

5.2. SRv6 Data Plane

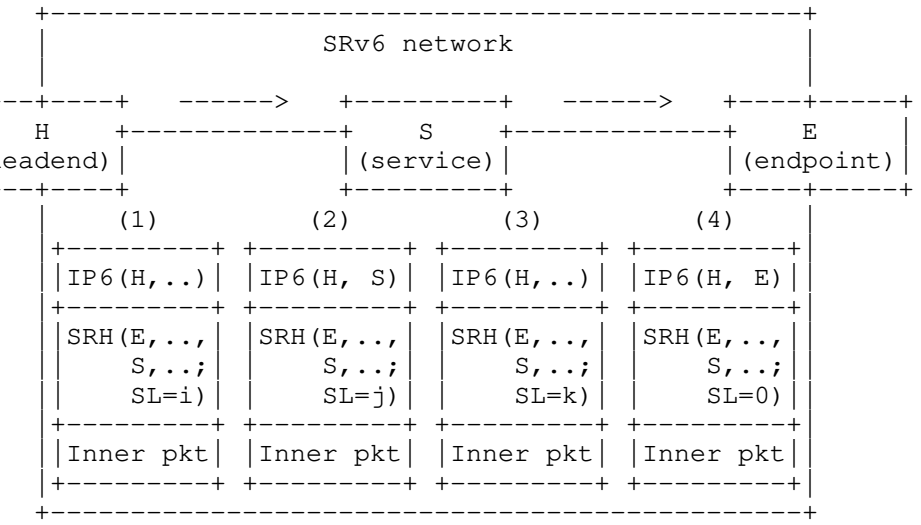


Figure 3: Packet walk in an SRv6 network

In an SRv6 network, the SR Policy is encoded into the packet as an IPv6 header possibly followed by a Segment Routing Header (SRH) [I-D.ietf-6man-segment-routing-header].

In the example shown on Figure 3, the SR policy should steer the traffic from the head-end H to the endpoint E via a service S. This translates into Segment-List that includes at least a segment SID(S) to the service, or service proxy, S and a segment SID(E) to the endpoint E. The Segment-List may also include additional intermediate SIDs if these are required for traffic engineering (e.g., the encode a low latency path between H and S and / or between S and E) or simply for reachability purposes. Indeed, the service SID locator may or may not be advertised in the routing protocol and, in the latter case, one or more SIDs might be needed before SID(S) in order to bring the packet up to node S, where SID(S) can be

interpreted. The same applies for the segment SID(E) at the SR policy endpoint.

When the packet arrives at S, this node determines how to process the packet based on the semantic locally associated to the active segment SID(S). If S is an SR-aware service, then SID(S) may provide additional context or indication on how to process the packet (e.g., a firewall SID may indicate which rule set should be applied onto the packet). If S is a proxy in front of an SR-unaware service, SID(S) indicates how and to which service attached to this proxy the packet should be transmitted. At some point in the process, the SRv6 End function is also applied in order to make the next SID, which may be SID(E) or another intermediate SID, active.

The "Inner pkt" on Figure 3 represents the SRv6 payload, which may be an encapsulated IP packet, an Ethernet frame or a transport-layer payload, for example.

6. SR Proxy Behaviors

This section describes several SR proxy behaviors designed to enable SR service programming through SR-unaware services. A system implementing one of these behaviors may handle the SR processing on behalf of an SR-unaware service and allows the service to properly process the traffic that is steered through it.

A service may be located at any hop in an SR policy, including the last segment. However, the SR proxy behaviors defined in this section are dedicated to supporting SR-unaware services at intermediate hops in the segment list. In case an SR-unaware service is at the last segment, it is sufficient to ensure that the SR information is ignored (IPv6 routing extension header with Segments Left equal to 0) or removed before the packet reaches the service (MPLS PHP, SRv6 decapsulation behavior or PSP flavor).

As illustrated on Figure 4, the generic behavior of an SR proxy has two parts. The first part is in charge of passing traffic from the network to the service. It intercepts the SR traffic destined for the service via a locally instantiated service segment, modifies it in such a way that it appears as non-SR traffic to the service, then sends it out on a given interface, IFACE-OUT, connected to the service. The second part receives the traffic coming back from the service on IFACE-IN, restores the SR information and forwards it according to the next segment in the list. IFACE-OUT and IFACE-IN are respectively the proxy interface used for sending traffic to the service and the proxy interface that receives the traffic coming back from the service. These can be physical interfaces or sub-interfaces

(VLANs) and, unless otherwise stated, IFACE-OUT and IFACE-IN can represent the same interface.

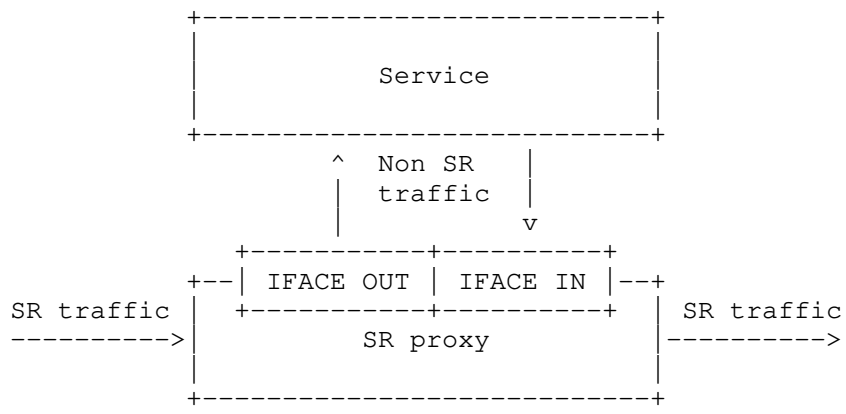


Figure 4: Generic SR proxy

In the next subsections, the following SR proxy mechanisms are defined:

- Static proxy
- Dynamic proxy
- Shared-memory proxy
- Masquerading proxy

Each mechanism has its own characteristics and constraints, which are summarized in the below table. It is up to the operator to select the best one based on the proxy node capabilities, the service behavior and the traffic type. It is also possible to use different proxy mechanisms within the same service policy.

		Static	Dynamic	Shared mem.	Masquerading
SR flavors	SR-MPLS	Y	Y	Y	-
	Inline SRv6	P	P	P	Y
	SRv6 encapsulation	Y	Y	Y	-
Chain agnostic configuration		N	N	Y	Y
Transparent to chain changes		N	Y	Y	Y
Service support	DA modification	Y	Y	Y	NAT
	Payload modification	Y	Y	Y	Y
	Packet generation	Y	Y	cache	cache
	Packet deletion	Y	Y	Y	Y
	Packet re-ordering	Y	Y	Y	Y
	Transport endpoint	Y	Y	cache	cache
Supported traffic	Ethernet	Y	Y	Y	-
	IPv4	Y	Y	Y	-
	IPv6	Y	Y	Y	Y

Figure 5: SR proxy summary

Note: The use of a shared memory proxy requires both the service (VNF) and the proxy to be running on the same node.

6.1. Static SR Proxy

The static proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an MPLS label stack or an IPv6 header on top of an inner packet, which can be Ethernet, IPv4 or IPv6.

A static SR proxy segment is associated with the following mandatory parameters

- INNER-TYPE: Inner packet type
- NH-ADDR: Next hop Ethernet address (only for inner type IPv4 and IPv6)
- IFACE-OUT: Local interface for sending traffic towards the service
- IFACE-IN: Local interface receiving the traffic coming back from the service
- CACHE: SR information to be attached on the traffic coming back from the service, including at least
  - CACHE.SA: IPv6 source address (SRv6 only)
  - CACHE.LIST: Segment list expressed as MPLS labels or IPv6 address

A static SR proxy segment is thus defined for a specific service, inner packet type and cached SR information. It is also bound to a pair of directed interfaces on the proxy. These may be both directions of a single interface, or opposite directions of two different interfaces. The latter is recommended in case the service is to be used as part of a bi-directional SR service policy. If the proxy and the service both support 802.1Q, IFACE-OUT and IFACE-IN can also represent sub-interfaces.

The first part of this behavior is triggered when the proxy node receives a packet whose active segment matches a segment associated with the static proxy behavior. It removes the SR information from the packet then sends it on a specific interface towards the associated service. This SR information corresponds to the full label stack for SR-MPLS or to the encapsulation IPv6 header with any attached extension header in the case of SRv6.

The second part is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This

policy attaches to the incoming traffic the cached SR information associated with the SR proxy segment. If the proxy segment uses the SR-MPLS data plane, CACHE contains a stack of labels to be pushed on top of the packets. With the SRv6 data plane, CACHE is defined as a source address, an active segment and an optional SRH (tag, segments left, segment list and metadata). The proxy encapsulates the packets with an IPv6 header that has the source address, the active segment as destination address and the SRH as a routing extension header. After the SR information has been attached, the packets are forwarded according to the active segment, which is represented by the top MPLS label or the IPv6 Destination Address. An MPLS TTL or IPv6 Hop Limit value may also be configured in CACHE. If it is not, the proxy should set these values according to the node's default setting for MPLS or IPv6 encapsulation.

In this scenario, there are no restrictions on the operations that can be performed by the service on the stream of packets. It may operate at all protocol layers, terminate transport layer connections, generate new packets and initiate transport layer connections. This behavior may also be used to integrate an IPv4-only service into an SRv6 policy. However, a static SR proxy segment can be used in only one service policy at a time. As opposed to most other segment types, a static SR proxy segment is bound to a unique list of segments, which represents a directed SR service policy. This is due to the cached SR information being defined in the segment configuration. This limitation only prevents multiple segment lists from using the same static SR proxy segment at the same time, but a single segment list can be shared by any number of traffic flows. Besides, since the returning traffic from the service is re-classified based on the incoming interface, an interface can be used as receiving interface (IFACE-IN) only for a single SR proxy segment at a time. In the case of a bi-directional SR service policy, a different SR proxy segment and receiving interface are required for the return direction.

The static proxy behavior may also be used for sending traffic through "bump in the wire" services that are transparent to the IP and Ethernet layers. This type of processing is assumed when the inner traffic type is Ethernet, since the original destination address of the Ethernet frame is preserved when the packet is steered into the SR Policy and likely associated with a node downstream of the policy tail-end. In case the inner type is IP (IPv4 or IPv6), the NH-ADDR parameter may be set to a dummy or broadcast Ethernet address, or simply to the address of the proxy receiving interface (IFACE-IN).

6.1.1. SR-MPLS Pseudocode

6.1.1.1. Static Proxy for Inner Type Ethernet

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the frame to the Ethernet module for transmission via
     interface IFACE-OUT.
```

Figure 6: SID processing for MPLS static proxy (Ethernet)

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Remove the preamble or Frame Check Sequence (FCS).
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 7: Inbound policy for MPLS static proxy (Ethernet)

6.1.1.2. Static Proxy for Inner Type IPv4

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 8: SID processing for MPLS static proxy (IPv4)

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the TTL and adjust the checksum accordingly.
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 9: Inbound policy for MPLS static proxy (IPv4)

6.1.1.3. Static Proxy for Inner Type IPv6

When processing an MPLS packet whose top label matches a locally instantiated MPLS static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. POP all labels in the MPLS label stack.
S02. Submit the packet to the IPv6 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 10: SID processing for MPLS static proxy (IPv6)

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the Hop Limit.
S04.   PUSH all labels from the retrieved CACHE entry.
S05.   Submit the packet to the MPLS module for transmission as per
       the top label in the MPLS label stack.
S06. }
```

Figure 11: Inbound policy for MPLS static proxy (IPv6)

6.1.2. SRv6 Pseudocode

6.1.2.1. Static Proxy for Inner Type Ethernet

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[Segments Left] from the SRH to the
       Destination Address of the IPv6 header.
S15.   If (Upper-layer header type != 143 (Ethernet)) {
S16.     Resubmit the packet to the IPv6 module for transmission to
         the new destination.
S17.   }
S18.   Perform IPv6 decapsulation.
S19.   Submit the frame to the Ethernet module for transmission via
       interface IFACE-OUT.
S20. }
```

Figure 12: SID processing for SRv6 static proxy (Ethernet)

S15: 143 (Ethernet) refers to the value assigned by IANA for "Ethernet" in the "Internet Protocol Numbers" registry.

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for Ethernet traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 143 (Ethernet)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the frame to the Ethernet module for transmission via
     interface IFACE-OUT.
```

Figure 13: Upper-layer header processing for SRv6 static proxy (Ethernet)

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Remove the preamble or Frame Check Sequence (FCS).
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 143 (Ethernet),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 14: Inbound policy for SRv6 static proxy (Ethernet)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the IPv6 header set to 143 (Ethernet).

6.1.2.2. Static Proxy for Inner Type IPv4

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[Segments Left] from the SRH to the
       Destination Address of the IPv6 header.
S15.   If (Upper-layer header type != 4 (IPv4)) {
S16.     Resubmit the packet to the IPv6 module for transmission to
         the new destination.
S17.   }
S18.   Perform IPv6 decapsulation.
S19.   Submit the packet to the IPv4 module for transmission on
       interface IFACE-OUT via NH-ADDR.
S20. }
```

Figure 15: SID processing for SRv6 static proxy (IPv4)

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv4 traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 4 (IPv4)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the packet to the IPv4 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 16: Upper-layer header processing for SRv6 static proxy (IPv4)

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the TTL and adjust the checksum accordingly.
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 4 (IPv4),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 17: Inbound policy for SRv6 static proxy (IPv4)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the IPv6 header set to 4 (IPv4).

6.1.2.3. Static Proxy for Inner Type IPv6

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[Segments Left] from the SRH to the
       Destination Address of the IPv6 header.
S15.   If (Upper-layer header type != 41 (IPv6)) {
S16.     Resubmit the packet to the IPv6 module for transmission to
         the new destination.
S17.   }
S18.   Perform IPv6 decapsulation.
S19.   Submit the packet to the IPv6 module for transmission on
       interface IFACE-OUT via NH-ADDR.
S20. }
```

Figure 18: SID processing for SRv6 static proxy (IPv6)

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 41 (IPv6)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
S03. }
S04. Perform IPv6 decapsulation.
S05. Submit the packet to the IPv6 module for transmission on
     interface IFACE-OUT via NH-ADDR.
```

Figure 19: Upper-layer header processing for SRv6 static proxy (IPv6)

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   Decrement the Hop Limit.
S04.   Perform IPv6 encapsulation with an SRH
         Source Address of the IPv6 header is set to CACHE.SA,
         Destination Address of the IPv6 header is set to
         CACHE.LIST[0],
         Next Header of the SRH is set to 41 (IPv6),
         Segment List of the SRH is set to CACHE.LIST.
S05.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S06. }
```

Figure 20: Inbound policy for SRv6 static proxy (IPv6)

S04: CACHE.LIST[0] represents the first entry in CACHE.LIST. Unless a local configuration indicates otherwise, the SIDs in CACHE.LIST should be encoded in the Segment List field in reversed order, the Segment Left and Last Entry values should be set of the length of CACHE.LIST minus 1. If CACHE.LIST contains a single entry, the SRH can be omitted and the Next Header field of the (outer) IPv6 header set to 41 (IPv6).

6.2. Dynamic SR Proxy

The dynamic proxy is an improvement over the static proxy that dynamically learns the SR information before removing it from the incoming traffic. The same information can then be re-attached to the traffic returning from the service. As opposed to the static SR proxy, no CACHE information needs to be configured. Instead, the dynamic SR proxy relies on a local caching mechanism on the node instantiating this segment.

Upon receiving a packet whose active segment matches a dynamic SR proxy function, the proxy node pops the top MPLS label or applies the SRv6 End behavior, then compares the updated SR information with the cache entry for the current segment. If the cache is empty or different, it is updated with the new SR information. The SR information is then removed and the inner packet is sent towards the service.

The cache entry is not mapped to any particular packet, but instead to an SR service policy identified by the receiving interface (IFACE-IN). Any non-link-local IP packet or non-local Ethernet frame received on that interface will be re-encapsulated with the cached headers as described in Section 6.1. The service may thus drop, modify or generate new packets without affecting the proxy.

6.2.1. SR-MPLS Pseudocode

The dynamic proxy SR-MPLS pseudocode is obtained by inserting the following instructions at the beginning of the static SR-MPLS pseudocode (Section 6.1.1).

```
S01. If the top label S bit is different from 0 {
S02.   Discard the packet.
S03. }
S04. POP the top label.
S05. Copy the IPv6 encapsulation in a CACHE entry associated with the
     interface IFACE-IN.
```

Figure 21: SID processing for MPLS dynamic proxy

S01: As mentioned at the beginning of Section 6, an SR proxy is not needed to include an SR-unaware service at the end of an SR policy.

S05: An implementation may optimize the caching procedure by copying information into the cache only if it is different from the current content of the cache entry. Furthermore, a TTL margin can be configured for the top label stack entry to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a TTL difference smaller than the configured margin should not trigger a cache update (provided that the labels are the same).

When processing an Ethernet frame, an IPv4 packet or an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the pseudocode reported in Figure 7, Figure 9 or Figure 11, respectively, is executed.

6.2.2. SRv6 Pseudocode

When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 dynamic proxy SID, the same pseudocode as described in Figure 12, Figure 15 and Figure 18, respectively for Ethernet, IPv4 and IPv6 traffic, is executed with the following addition between lines S17 and S18.

```
(... S17.     })
S17.1.   Copy the IPv6 encapsulation in a CACHE entry associated with
         the interface IFACE-IN.
(S18.     Perform IPv6 decapsulation...)
```

Figure 22: SID processing for SRv6 dynamic proxy

An implementation may optimize the caching procedure by copying information into the cache only if it is different from the current content of the cache entry. A Hop Limit margin can be configured to prevent constant cache updates when multiple equal-cost paths with different hop counts are used towards the SR proxy node. In that case, a Hop Limit difference smaller than the configured margin should not trigger a cache update. Similarly, the Flow Label value can be ignored when comparing the current packet IPv6 header with the cache entry. In this case, the Flow Label should be re-computed by the proxy node when it restores the IPv6 encapsulation from the cache entry.

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 dynamic proxy SID, process the packet as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1.

When processing an Ethernet frame, an IPv4 packet or an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the same pseudocode as in Figure 14, Figure 17 or Figure 20, respectively, is executed.

6.3. Shared Memory SR Proxy

The shared memory proxy is an SR endpoint behavior for processing SR-MPLS or SRv6 encapsulated traffic on behalf of an SR-unaware service. This proxy behavior leverages a shared-memory interface with a virtualized service (VNF) in order to hide the SR information from an SR-unaware service while keeping it attached to the packet. We assume in this case that the proxy and the VNF are running on the same compute node. A typical scenario is an SR-capable vrouter running on a container host and forwarding traffic to VNFs isolated within their respective container.

6.4. Masquerading SR Proxy

The masquerading proxy is an SR endpoint behavior for processing SRv6 traffic on behalf of an SR-unaware service. This proxy thus receives SR traffic that is formed of an IPv6 header and an SRH on top of an inner payload. The masquerading behavior is independent from the inner payload type. Hence, the inner payload can be of any type but it is usually expected to be a transport layer packet, such as TCP or UDP.

A masquerading SR proxy segment is associated with the following mandatory parameters:

- NH-ADDR: Next hop Ethernet address

- IFACE-OUT: Local interface for sending traffic towards the service

- IFACE-IN: Local interface receiving the traffic coming back from the service

A masquerading SR proxy segment is thus defined for a specific service and bound to a pair of directed interfaces or sub-interfaces on the proxy. As opposed to the static and dynamic SR proxies, a masquerading segment can be present at the same time in any number of SR service policies and the same interfaces can be bound to multiple masquerading proxy segments. The only restriction is that a masquerading proxy segment cannot be the last segment in an SR service policy.

The first part of the masquerading behavior is triggered when the proxy node receives an IPv6 packet whose Destination Address matches a masquerading proxy SID. The proxy inspects the IPv6 extension headers and substitutes the Destination Address with the last SID in the SRH attached to the IPv6 header, which represents the final destination of the IPv6 packet. The packet is then sent out towards the service.

The service receives an IPv6 packet whose source and destination addresses are respectively the original source and final destination. It does not attempt to inspect the SRH, as RFC8200 specifies that routing extension headers are not examined or processed by transit nodes. Instead, the service simply forwards the packet based on its current Destination Address. In this scenario, we assume that the service can only inspect, drop or perform limited changes to the packets. For example, Intrusion Detection Systems, Deep Packet Inspectors and non-NAT Firewalls are among the services that can be supported by a masquerading SR proxy. Flavors of the masquerading behavior are defined in Section 6.4.2 and Section 6.4.3 to support a wider range of services.

The second part of the masquerading behavior, also called de-masquerading, is an inbound policy attached to the proxy interface receiving the traffic returning from the service, IFACE-IN. This policy inspects the incoming traffic and triggers a regular SRv6 endpoint processing (End) on any IPv6 packet that contains an SRH. This processing occurs before any lookup on the packet Destination Address is performed and it is sufficient to restore the right active SID as the Destination Address of the IPv6 packet.

6.4.1. SRv6 Masquerading Proxy Pseudocode

Masquerading: When processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 masquerading proxy SID, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Proceed to process the next header in the packet.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S07.   }
S08.   max_last_entry = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_last_entry) or
           (Segments Left > (Last Entry + 1))) {
S10.     Send an ICMP Parameter Problem message to the Source Address,
         Code 0 (Erroneous header field encountered),
         Pointer set to the Segments Left field,
         Interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Decrement Segments Left by 1.
S14.   Copy Segment List[0] from the SRH to the Destination Address
       of the IPv6 header.
S15.   Submit the packet to the IPv6 module for transmission on
       interface IFACE-OUT via NH-ADDR.
S16. }
```

Figure 23: SID processing for SRv6 masquerading proxy

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 masquerading proxy SID, process the packet as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1.

De-masquerading: When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the following pseudocode is executed.

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
S10.     Copy Segment List[Segments Left] from the SRH to the
         Destination Address of the IPv6 header.
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

Figure 24: Inbound policy for SRv6 masquerading proxy

6.4.2. Destination NAT Flavor

Services modifying the destination address in the packets they process, such as NATs, can be supported by reporting the updated Destination Address back into the Segment List field of the SRH.

The Destination NAT flavor of the SRv6 masquerading proxy is enabled by adding the following instruction between lines S09 and S10 of the de-masquerading pseudocode in Figure 24.

```
(... S09.     })
S09.1.   Copy the Destination Address of the IPv6 header to the
         Segment List[0] entry of the SRH.
(S10.     Copy Segment List[Segments Left] from the SRH to the
          Destination Address of the IPv6 header...)
```

6.4.3. Caching Flavor

Services generating packets or acting as endpoints for transport connections can be supported by adding a dynamic caching mechanism similar to the one described in Section 6.2.

The caching flavor of the SRv6 masquerading proxy is enabled by:

- Adding the following instruction between lines S14 and S15 of the masquerading pseudocode in Figure 23.

```
(... S14.   Copy Segment List[0] from the SRH to the Destination
            Address of the IPv6 header.
S14.1. Copy the IPv6 encapsulation in a CACHE entry associated with
       the interface IFACE-IN.
(S15.   Submit the packet to the IPv6 module for transmission on
        interface IFACE-OUT via NH-ADDR.)
```

- Updating the de-masquerading pseudocode such that, in addition to the SRH processing in Figure 24, the following pseudocode is executed when processing an IPv6 packet (received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN) that does not contain an SRH.

```
S01. Retrieve the CACHE entry associated with IFACE-IN.
S02. If the CACHE entry is not empty {
S03.   If (IPv6 Hop Limit <= 1) {
S04.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S05.   }
S06.   Decrement Hop Limit by 1.
S07.   Update the IPv6 encapsulation according to the retrieved CACHE
       entry.
S08.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S09. }
```

7. Metadata

7.1. MPLS Data Plane

Metadata can be carried for SR-MPLS traffic in a Segment Routing Header inserted between the last MPLS label and the MPLS payload. When used solely as a metadata container, the SRH does not carry any segment but only the mandatory header fields, including the tag and flags, and any TLVs that is required for transporting the metadata.

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata in an MPLS packet is a potential issue to be addressed. One possible solution is to add the indication about the presence of metadata in the semantic of the SIDs. Note that only the SIDs whose behavior involves looking at the metadata or the MPLS payload would need to include such semantic (e.g., service segments). Other segments, such as topological

segments, are not affected by the presence of metadata. Another, more generic, solution is to introduce a protocol identifier field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

7.2. IPv6 Data Plane

7.2.1. SRH TLV Objects

The IPv6 SRH TLV objects are designed to carry all sorts of metadata. TLV objects can be imposed by the ingress edge router that steers the traffic into the SR service policy.

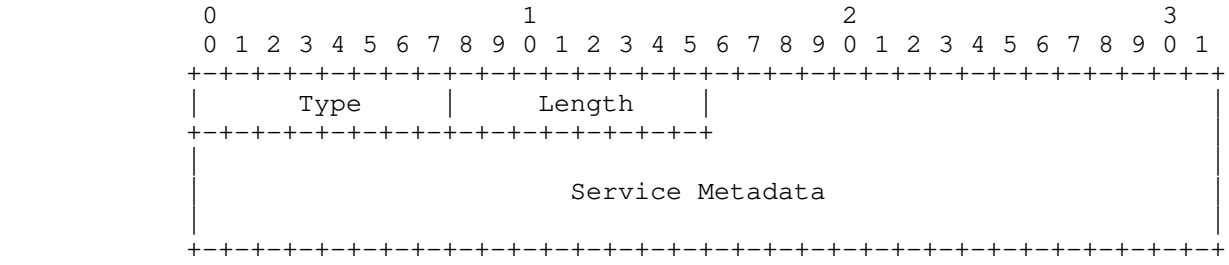
An SR-aware service may impose, modify or remove any TLV object attached to the first SRH, either by directly modifying the packet headers or via a control channel between the service and its forwarding plane.

An SR-aware service that re-classifies the traffic and steers it into a new SR service policy (e.g. DPI) may attach any TLV object to the new SRH.

Metadata imposition and handling will be further discussed in a future version of this document.

7.2.1.1. Opaque Metadata TLV

This document defines an SRv6 TLV called Opaque Metadata TLV. This is a fixed-length container to carry any type of Service Metadata. No assumption is made by this document on the structure or the content of the carried metadata. The Opaque Metadata TLV has the following format:



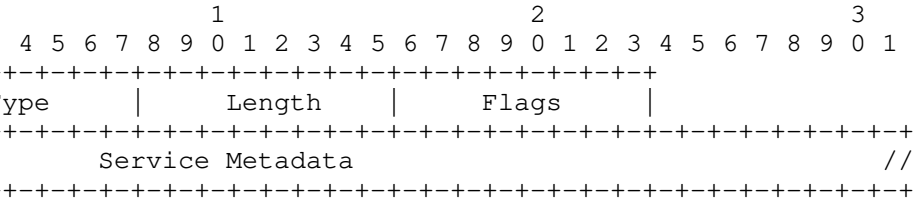
where:

- Type: to be assigned by IANA.
- Length: 14.

   o Service Metadata: 14 octets of opaque data.

7.2.1.2. NSH Carrier TLV

   This document defines an SRv6 TLV called NSH Carrier TLV. It is a container to carry Service Metadata in the form of Variable-Length Metadata as defined in [RFC8300] for NSH MD Type 2. The NSH Carrier TLV has the following format:



   where:

   o Type: to be assigned by IANA.

   o Length: the total length of the TLV.

   o Flags: 8 bits. No flags are defined in this document. SHOULD be set to 0 on transmission and MUST be ignored on receipt.

   o Service Metadata: a list of Service Metadata TLV as defined in [RFC8300] for NSH MD Type 2.

7.2.2. SRH Tag

   The SRH tag identifies a packet as part of a group or class of packets [I-D.ietf-6man-segment-routing-header].

   In the context of service programming, this field can be used to encode basic metadata in the SRH. An example use-case is to leverage the SRH tag to encode a policy ID. This policy ID can then be used by an SR-aware function to identify a particular processing policy to be applied on that packet.

8. Implementation Status

   This section is to be removed prior to publishing as an RFC.

8.1. SR-Aware Services

Specific SRv6 support has been implemented for the below open-source services:

- Iptables (1.6.2 and later) [IPTABLES]
- Nftables (0.8.4 and later) [NFTABLES]
- Snort [SNORT]

In addition, any service relying on the Linux kernel, version 4.10 and later, or FD.io VPP for packet forwarding can be considered as SR-aware.

8.2. Proxy Behaviors

The static SR proxy is available for SR-MPLS and SRv6 on various Cisco hardware and software platforms. Furthermore, the following proxies are available on open-source software.

		VPP	Linux
MPLS	Static proxy	Available	In progress
	Dynamic proxy	In progress	In progress
	Shared memory proxy	In progress	In progress
SRv6	Static proxy	Available	In progress
	Dynamic proxy	Available	Available
	Shared memory proxy	In progress	In progress
	Masquerading proxy	Available	Available

Figure 25: Open-source implementation status table

9. Related Works

The Segment Routing solution addresses a wide problem that covers both topological and service policies. The topological and service instructions can be either deployed in isolation or in combination. SR has thus a wider applicability than the architecture defined in [RFC7665]. Furthermore, the inherent property of SR is a stateless

network fabric. In SR, there is no state within the fabric to recognize a flow and associate it with a policy. State is only present at the ingress edge of the SR domain, where the policy is encoded into the packets. This is completely different from other proposals such as [RFC8300] and the MPLS label swapping mechanism described in [I-D.ietf-mpls-sfc], which rely on state configured at every hop of the service chain.

10. IANA Considerations

10.1. SRv6 Endpoint Behaviors

This I-D requests the IANA to allocate, within the "SRv6 Endpoint Behaviors" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry, the following allocations:

Value	Description	Reference
TBA1-1	End.AN - SR-aware function (native)	[This.ID]
TBA1-2	End.AS - Static proxy	[This.ID]
TBA1-3	End.AD - Dynamic proxy	[This.ID]
TBA1-4	End.AM - Masquerading proxy	[This.ID]
TBA1-5	End.AM - Masquerading proxy with NAT	[This.ID]
TBA1-6	End.AM - Masquerading proxy with Caching	[This.ID]
TBA1-7	End.AM - Masquerading proxy with NAT & Caching	[This.ID]

10.2. Segment Routing Header TLVs

This I-D requests the IANA to allocate, within the "Segment Routing Header TLVs" registry, the following allocations:

Value	Description	Reference
TBA2-1	Opaque Metadata TLV	[This.ID]
TBA2-2	NSH Carrier TLV	[This.ID]

11. Security Considerations

The security requirements and mechanisms described in [RFC8402], [I-D.ietf-6man-segment-routing-header] and [I-D.ietf-spring-srv6-network-programming] also apply to this document.

This document does not introduce any new security vulnerabilities.

12. Acknowledgements

The authors would like to thank Thierry Couture, Ketan Talaulikar, Loa Andersson, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein and Joel M. Halpern for their valuable comments and suggestions on the document.

13. Contributors

The following people have contributed to this document:

Pablo Camarillo
Cisco Systems, Inc.
Spain

Email: pcamaril@cisco.com

Bart Peirens
Proximus
Belgium

Email: bart.peirens@proximus.com

Dirk Steinberg
Lapishills Consulting Limited
Cyprus

Email: dirk@lapishills.com

Ahmed AbdelSalam
Cisco Systems, Inc.
Italy

Email: ahabdels@cisco.com

Gaurav Dawra
LinkedIn
United States of America

Email: gdawra@linkedin.com

Stewart Bryant
Futurewei Technologies Inc

Email: stewart.bryant@gmail.com

Hamid Assarpour
Broadcom

Email: hamid.assarpour@broadcom.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura
Individual

Email: jefftant@gmail.com

Martin Vigoureux
Nokia

Email: martin.vigoureux@nokia.com

Jisu Bhattacharya
Cisco Systems, Inc.
United States of America

Email: jisu@cisco.com

14. References

14.1. Normative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-22 (work in progress), May 2019.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-08 (work in progress), July 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-18 (work in progress), August 2020.

[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>.

14.2. Informative References

[I-D.dawra-idr-bgp-sr-service-chaining]
Dawra, G., Filsfils, C., daniel.bernier@bell.ca, d., Uttaro, J., Decraene, B., Elmalky, H., Xu, X., Clad, F., and K. Talaulikar, "BGP Control Plane Extensions for Segment Routing based Service Chaining", draft-dawra-idr-bgp-sr-service-chaining-02 (work in progress), January 2018.

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-05 (work in progress), April 2020.

[I-D.ietf-mpls-sfc]
Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", draft-ietf-mpls-sfc-07 (work in progress), March 2019.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X., Assarpour, H., Ma, S., and F. Clad, "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-06 (work in progress), March 2019.

[IFIP18] Abdelsalam, A., Salsano, S., Clad, F., Camarillo, P., and C. Filsfils, "SEgment Routing Aware Firewall For Service Function Chaining scenarios", IFIP Networking conference , May 2018.

[IPTABLES]
"iptables-1.6.2 changes", February 2018, <https://netfilter.org/projects/iptables/files/changes-iptables-1.6.2.txt>.

[NFTABLES]
"nftables-0.8.4 changes", May 2018, <https://netfilter.org/projects/nftables/files/changes-nftables-0.8.4.txt>.

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

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

[SNORT] "SR-Snort", March 2018, <https://github.com/SRouting/SR-Snort>.

Authors' Addresses

Francois Clad (editor)
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Xiaohu Xu (editor)
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Cheng Li
Huawei

Email: chengli13@huawei.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Shaowen Ma
Mellanox

Email: mashaowen@gmail.com

Chaitanya Yadlapalli
AT&T
USA

Email: cy098d@att.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: October 3, 2020

Yimin Shen
Zhaohui Zhang
Juniper Networks
Rishabh Parekh
Cisco Systems
Hooman Bidgoli
Nokia
Yuji Kamite
NTT Communications

April 1, 2020

Point-to-Multipoint Transport Using Chain Replication in Segment Routing
draft-shen-spring-p2mp-transport-chain-02

Abstract

This document specifies a point-to-multipoint (P2MP) transport mechanism based on chain replication. It can be used in segment routing to achieve traffic optimization.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 3, 2020.

Table of Contents

1. Introduction

The Segment Routing Architecture [RFC8402] describes segment routing (SR) and its instantiation in two data planes, i.e. MPLS and IPv6. In SR, point-to-multipoint (P2MP) transport is currently achieved by using ingress replication, where a point-to-point (P2P) SR tunnel is constructed from a root node to each leaf node, and every ingress packet is replicated and sent via a bundle of such P2P SR tunnels to all the leaf nodes. Although this approach provides P2MP reachability, it does not consider traffic optimization across the tunnels, as the path of each tunnel is computed or decided independently.

An alternative approach would be to use P2MP-tree based transport. Such approach can achieve maximum traffic optimization, but it relies a controller or path computation element (PCE) to dynamically provision and manage "replication segments" on branch nodes. The replication segments are essentially per-P2MP-tree (i.e. per-tunnel) state on transit routers. Therefore, this approach is not fully aligned with SR's principles of single-point (i.e. ingress router) provisioning and stateless core.

This document introduces a new solution for P2MP transport in SR, based on "chain replication". In this solution, P2MP transport is achieved by constructing a set of "P2MP chain tunnels" (or simply "P2MP chains") from a root node to leaf nodes. Each P2MP chain is a tunnel with a leaf node at tail end and some transit leaf nodes along the path, resembling a chain. The leaf node at the tail end behaves as a normal receiver. Each transit leaf node replicates a packet once for local processing off the chain, and also forwards the original packet down the chain. The root node replicates and sends packets via the set of P2MP chains to all the leaf nodes.

As a P2MP chain can reach multiple leaf nodes, it is considered to be more efficient than the multiple P2P tunnels which would be needed in ingress replication to reach these leaf nodes. Compared with ingress replication and the P2MP-tree based approach, this solution provides a middle ground by achieving a certain level of traffic optimization, while aligning with the fundamental principles of SR, including single-point provisioning and stateless core. The solution can be used to improve P2MP transport efficiency in general, and to achieve maximum traffic optimization in certain types of topologies.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174].

3. Applicability

The P2MP transport mechanism in this document is generally applicable to all networks. However, it benefits more for certain types of topologies than others. These topologies include ring topologies, linear topologies, topologies with leaf nodes concentrated in geographical sites which can be modeled as leaf groups, etc.

The mechanism is transparent to all transit routers. Leaf nodes intended to take advantage of the mechanism will need to support the new forwarding behavior specified in this document. For other leaf nodes, the mechanism has a backward compatibility to allow them to be reached by P2P tunnels using ingress replication. Path computation and P2MP chain construction will need to be supported by a controller or root nodes, depending on where they are performed.

The mechanism is applicable to both SR-MPLS [RFC8660] and SRv6 [SRv6-SRH], [SRv6-Programming].

The mechanism does not create any state of P2MP tunnel or P2MP tree on routers. Therefore, if leaf nodes need to know the service level context (e.g. source, VPN) of a P2MP stream, they must rely on the information contained in an inner header. In SR-MPLS, service labels may be allocated from a domain-wide common block (DCB) to serve as globally unique context indicators. In SRv6, a root node's IP address or an upstream-assigned context indicator may be encoded in the source address of IPv6 header, or a downstream-assigned context indicator may be encoded in the ARG portion of a service SID.

4. P2MP Transport Using Chain Replication

In this document, a P2MP stream associated with a root node and a set of leaf nodes is denoted as {root node, leaf nodes}. It is achieved by using a bundle of P2MP chains covering all the leaf nodes. Each P2MP chain is a tunnel starting from the root node and reaching one or multiple leaf nodes along the path. The tail-end node of the P2MP chain is a leaf node, called a "tail-end" leaf node. Each leaf node traversed by the P2MP chain is called a "transit" leaf node. As a special case, a P2MP chain may have no transit leaf node, but only a tail-end leaf node, essentially becoming a P2P tunnel of ingress replication.

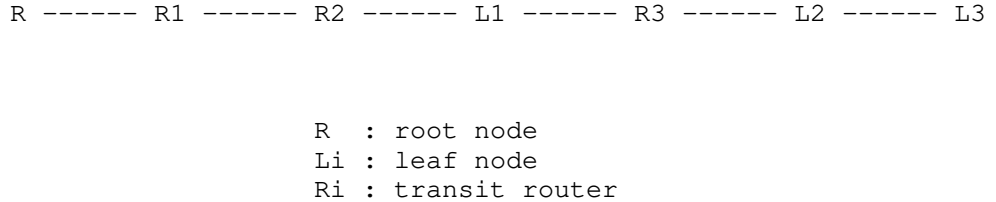


Figure 1

A tail-end leaf node and a transit leaf nodes have different behaviors when processing a received packet. In particular, a tail-end leaf node processes the packet as a normal receiver. A transit leaf node not only processes the packet as a receiver, but also

forwards it downstream along the P2MP chain, hence acting as a "bud node". To achieve this, the transit leaf node needs to replicate the packet, producing two packets, one for forwarding and the other for local processing. Such packet replication happens on every transit leaf node along a P2MP chain. Therefore, it is called "chain replication".

This document introduces a new type of segments, called "bud segments", to facilitate the above packet processing on transit leaf nodes. The segment ID (SID) of a bud segment is a "bud-SID".

4.1. Bud Segment

On a transit leaf node, a bud segment represents the following instructions for forwarding hardware to execute on a received packet P. They apply when the active SID of the packet P is the bud-SID of this bud segment.

[1] Replicate the packet P to generate a copy P1.

[2] For P, perform a NEXT operation on the bud-SID, make the next SID active, and forward the packet based on that SID.

[3] For P1, perform a sequence of NEXT operations on the bud-SID and all the subsequent SIDs of the P2MP chain, and process the packet locally. (The SIDs of the P2MP chain are not useful for processing P1 locally. Hence, they are removed before the processing.)

Bud segments are global segments of leaf nodes. They are routable segments via topological shortest-paths. Bud-SIDs are allocated from SRGB (SR global block). Only one bud segment is needed per leaf node, and per SR-MPLS or SRv6. It is used only when the leaf node is a transit leaf node on a P2MP chain.

In SR-MPLS, bud-SIDs are labels, and penultimate hop popping (PHP) MUST be disabled for bud-SID labels. In SRv6, bud-SIDs are IPv6 addresses explicitly associated with bud segments. Therefore, the above instructions [1] to [3] are achieved in different ways in SR-MPLS and SRv6:

(a) In SR-MPLS, the packet may have a service label(s) after P2MP chain labels in MPLS header, e.g. a VPN label, a bridge domain label, a source Ethernet segment label, etc. Therefore, the bud segment MUST have a way to identify the position of the last P2MP chain label, in order to execute [3] above. This document introduces an "end-of-chain" (EoC) label to facilitate the process. The EoC label is an extended special-purpose label (ESPL) [RFC 7274] with value TDB. When a root node constructs an MPLS header for a packet, if the packet has a service label(s), the root node MUST push the Extension Label (XL, value 15) and the EoC label, after pushing the service label(s) and before push P2MP chain labels. Hence, [XL, EoC] serves as a recognizable pattern to indicate the end of the P2MP chain labels. If the packet does not have a service label(s), the root node SHOULD NOT push [XL, EoC] to the MPLS header. In any case, in [3] above, the bud segment MUST pop labels until [XL, EoC] are popped or all labels have been popped.

(b) In SRv6, the packet is encapsulated with an outer IPv6 header corresponding to the P2MP chain, optionally followed by a segment routing header (SRH) containing the SIDs of the P2MP chain, and

followed by an inner header (of IPv4, IPv6, MPLS, layer-2, etc.) associated with a service. In [3] above, the bud segment SHOULD simply remove the outer IPv6 header and the SRH (if any), and leave the packet with the inner header to local processing.

Bud segments are shared by all P2MP streams, i.e. all combinations of {root node, leaf nodes}. A leaf node SHOULD advertise a bud segment for SR-MPLS, if its forwarding hardware supports the above SR-MPLS processing. Likewise, it SHOULD advertise a bud segment for SRv6, if its forwarding hardware supports the above SRv6 processing. The advertisement may be via IGP (ISIS, OSPF) or BGP-LS. The advertisement allows the leaf node to be considered as a transit leaf node on a P2MP chain. If a leaf node does not advertise a bud segment, it can only be considered as a tail-end leaf node on a P2MP chain, or reached via a P2P tunnel using ingress replication.

Bud segments are generic purpose segments. They may also be used in cases other than P2MP transport, such as traffic monitoring. These use cases are out of the scope of this document.

4.2. P2MP Chain

Construction of P2MP chains for a P2MP stream is performed by a controller or the root node based on path computation (Section 5). This decides the number of P2MP chains to use, and the set of leaf nodes that each P2MP chain reaches. In general, if the leaf nodes of the P2MP stream cannot be covered by using a single P2MP chain, multiple P2MP chains MUST be used, and the root node MUST replicate ingress packets over the P2MP chains.

The path of a P2MP chain is a single path traversing one or multiple transit leaf nodes and terminating at a tail-end leaf node. Between the root node and the first transit leaf node, and between two consecutive leaf nodes, there may be none, one, or multiple transit routers.

The path is then translated to a SID list to be programmed on the root node. In the SID list, each transit leaf node has its bud-SID in a corresponding position. Given a P2MP chain to a set of leaf nodes in the order of L1, L2, ..., Ln, the SID list may be represented as:

<SID_11, SID_12, ...>, bud-SID of L1, ..., <SID_i1, SID_i2, ...>, bud-SID of Li, ..., <SID_n1, SID_n2, ...>

Where:

- <SID_11, SID_12, ...> is the sub-path from the root node to L1.
- <SID_i1, SID_i2, ...> is the sub-path from Li-1 to Li.
- <SID_n1, SID_n2, ...> is the sub-path from Ln-1 to Ln. There is no need for Ln's bud-SID to be at the end of the SID list, because the tail-end leaf node does not perform a chain replication.

Each of the above sub-paths is a regular point-to-point path. The SIDs in the sub-path are regular SIDs, such as adjacency-SIDs, node-SIDs, binding-SIDs, etc. There is no SID specific to the given P2MP chain. A sub-path from Li-1 to Li may have an empty SID list, if the sub-path takes the shortest path indicated by the bud-SID of Li.

The root node then uses the SID list in packet encapsulation. Note

that in the SR-MPLS case where the EoC label is needed, [XL, EoC] MUST be pushed to an MPLS header, before the SID list is pushed.

4.3. Example

In the following example, P2MP transport is needed from the root node R, to leaf nodes L1, L2, L3 and L4.

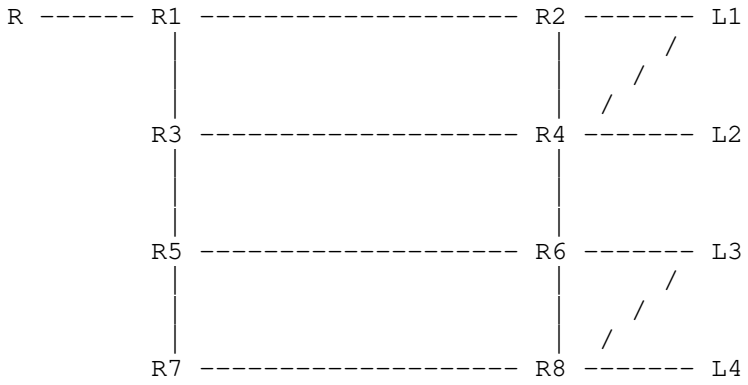


Figure 2

Path computation results in two P2MP chains:

P2MP chain 1:

Path: R -> R1 -> R2 -> L1 -> R4 -> L2, where L1 is a transit leaf node, and L2 is the tail-end leaf node.

Assuming that the sub-path R -> R1 -> R2 -> L1 is not the shortest path from R to L1, so that an explicit sub-path must be used. Also assuming that the sub-path L1 -> R4 -> L2 is the shortest path from L1 to L2, so that the node-SID of L2 can be used to represent this sub-path. The segment list applied to packets on R is:

adj-SID 100 - link from R to R1

adj-SID 200 - link from R1 to R2

adj-SID 300 - link from R2 to L1

bud-SID 1000 - L1

node-SID 2000 - L2

P2MP chain 2:

Path: R -> R1 -> R3 -> R5 -> R6 -> L3 -> R8 -> L4, where L3 is a transit leaf node, and L4 is the tail-end leaf node.

Assuming that the sub-path R -> R1 -> R3 -> R5 -> R6 -> L3 is the shortest path from R to L3, so that the bud-SID of L3 can be used to represent this sub-path. Also assuming that the sub-path L3 -> R8 -> L4 is not the shortest path from L3 to L4, so that an explicit sub-path must be used. The segment list applied to packets on R is:

bud-SID 3000 - L3

adj-SID 600 - link from L3 to R8

adj-SID 700 - link from R8 to L4

bud-SID 4000 - L4

5. Path Computation for P2MP Chains

Path computation for the P2MP chains of a P2MP stream {root node, leaf nodes} lies in the responsibility of a controller or the root node. This document does not enforce a particular computation algorithm. In general, any P2P path computation algorithm may be extended to serve the purpose.

The path computation may consider general metric for shortest paths, or traffic engineering (TE) constraints for TE paths. This document recommends the following constraints to be considered as well:

- The maximum hop count of path. This SHOULD be based on the maximum delay allowed for a packet to accumulate before reaching a tail-end leaf node. It may be used to restrict the length of each P2MP chain.

- The maximum length of SID list. This SHOULD be based on the maximum header size which a root node may apply to a packet. This is typically a limit of forwarding hardware. Note that a SID list is translated from a computed path. Hence, the length of the SID list and the hop count of the path are generally not the same.

- Maximum leaf nodes per P2MP chain. This may be used to restrict the length of each P2MP chain.

- Maximum hops between consecutive leaf nodes on a P2MP chain. This may be used prevent a P2MP chain from attempting leaf nodes which should ideally be reached by separate P2MP chains.

- Maximum times that a node or link may be traversed by a P2MP chain. This may be used to prevent a P2MP chain from congesting a node or link.

As an example, the path computation may start with forming a path from the root node to the closest leaf node, and extend the path to a second leaf node, a third leaf node, and so on. When any of the above limits is hit, the current computation SHOULD end, the path SHOULD be saved as a completed P2MP chain, and a new computation SHOULD be performed for the rest leaf nodes. This process SHOULD repeat until all the leaf nodes are covered, where a set of paths have been computed.

The path computation is generally deterministic in a ring or linear topology. In an arbitrary topology, deterministic path computation may be achieved by dividing leaf nodes into groups based on their location, and computing a separate path for each group. A group may even define its leaf nodes as an ordered list of loose hops, so that a path will traverse the leaf nodes in the specified order. During the computation of a group, if any of the above limits is hit, the computation SHOULD end, the path SHOULD be saved as a completed P2MP chain, and a new computation SHOULD be performed for the rest leaf nodes of the group. This process SHOULD repeat until all the leaf nodes of the group are covered. In this case, the group will end up

using multiple P2MP chains.

6. IGP and BGP-LS Extensions for Bud Segment

The protocol extensions of IGP (ISIS and OSPF) and BGP-LS for bud segment advertisement will be specified in the next version of this document.

7. Bud Segments for Special Processing

So far, the discussion in this document has been focusing on bud segments that are created on a per SR-MPLS or SRv6 basis on each leaf node. These bud segments indicate generic local processing which is based on the inner header of a packet. They are applicable to most of the common cases of P2MP transport, and hence are viewed as the default bud segments of leaf nodes.

The concept of bud segment can also be extended to other cases, where a transit leaf node needs to perform a special kind of local processing for packets, but cannot derive the context of the processing from their inner headers. For example, the node may need to forward the packets over one or more interfaces or tunnels to downstream device(s), or to process the packets based on a particular forwarding table or policy, and so on. In such cases, a dedicated bud segment SHOULD be created for the special kind of local processing. It will serve the general purpose of a bud segment, and additionally indicate the context of the special processing. Note that scaling of such bud segments per leaf node SHOULD be a consideration in network design, as well as the requirement for a controller or ingress router to have the knowledge of various special processing scenarios on leaf nodes and use the corresponding bud segments in P2MP chain construction.

8. IANA Considerations

This document requires IANA to allocate a value from the "Extended Special-Purpose MPLS Label Values" registry for the EoC label.

The document also requires IANA registration and allocation for the ISIS, OSPF and BGP-LS extensions for bud segment advertisement. The details will be provided in the next version of this document.

9. Security Considerations

This document introduces bud segments for leaf nodes to act as both packet receivers and transit routers. A security attack may target on a leaf node by constructing malicious packets with the node's bud-SID. Such kind of attacks can be defeated by restricting bud segment distribution and P2MP chain construction within the scope of a controller and a given network.

10. Acknowledgements

This document leverages work done by Alexander Arseniev and Ron Bonica.

11. Contributors

Alexander Arseniev

Juniper networks

Email: aarseniev@juniper.net

Ron Bonica

Juniper networks

Virginia

USA

Email: rbonica@juniper.net

12. References

12.1. Normative References

[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>.

[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>.

[SRv6-SRH] Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header", draft-ietf-6man-segment-routing-header (work in progress), 2019.

[SRv6-Programming] Filsfils, C., Garvia, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming (work in progress), 2019.

12.2. Informative References

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

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

Authors' Addresses

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

Email: yshen@juniper.net

Zhaohui Zhang
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

Email: zzhang@juniper.net

Rishabh Parekh
Cisco Systems
San Jose, CA
USA

Email: riparekh@cisco.com

Hooman Bidgoli
Nokia
Ottawa
Canada

Email: hooman.bidgoli@nokia.com

Yuji Kamite
NTT Communications
Tokyo
Japan

Email: y.kamite@ntt.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 28, 2020

T. Graf
Swisscom

April 26, 2020

Export of MPLS Segment Routing Label Type Information in IP Flow Information Export (IPFIX) draft-tgraf-ipfix-mpls-sr-label-type-04

Abstract

This document introduces additional code points in the mplsTopLabelType Information Element for IS-IS, OSPFv2, OSPFv3 MPLS Segment Routing (SR) extensions and a new SID type element to enable Segment Routing label and segment type information in IP Flow Information Export (IPFIX).

Status of This Memo

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

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

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

This Internet-Draft will expire on October 28, 2020.

Table of Contents

1. Introduction

Besides BGP-4 [RFC8277], LDP [RFC5036] and BGP VPN [RFC4364], three new routing-protocols, OSPFv2 Extensions [RFC8665], OSPFv3 Extensions [RFC8666] and IS-IS Extensions [RFC8667] have been added to the list of routing-protocols able to propagate Segment Routing labels for the MPLS dataplane [RFC8660].

Traffic Accounting in Segment Routing Networks [I-D.ali-spring-sr-traffic-accounting] describes how IPFIX can be leveraged to account traffic to MPLS Segment Routing label dimensions within a Segment Routing domain.

In the Information Model for IP Flow Information Export IPFIX [RFC7012], the information element #46 mplsTopLabelType describes which MPLS control plane protocol allocated the top-of-stack label in the MPLS label stack. RFC 7012 section 7.2 [RFC7012] describes the IANA Information Element #46 SubRegistry [IANA-IPFIX-IE46] where new code points should be added.

2. MPLS Segment Routing Top Label Type

By introducing three new code points to information element #46 mplsTopLabelType for IS-IS, OSPFv2 and OSPFv3, when Segment Routing with one of these three routing protocols is deployed, we get insight into which traffic is being forwarded based on which MPLS control plane protocol.

A typical use case scenario is to monitor MPLS control plane migrations from LDP to IS-IS or OSPF. By looking at the MPLS label value itself, it is not always clear as to which label protocol it belongs, since they could potentially share the same label allocation range. This is the case for IGP-Adjacency SID's and LDP as an example.

3. Segment Routing Segment Identifier Type

The introduction of a new Information Element called SrSidType, which contains the Segment Routing Segment Identifier type according to Segment Routing Architecture [RFC8402], allows the Segment Routing forwarding behaviour to be exported in IPFIX.

A typical use case scenario is to monitor the forwarding behaviour when Topology Independent Fast Reroute [I-D.ietf-rtgwg-segment-routing-ti-lfa] or micro loop avoidance [I-D.bashandy-rtgwg-segment-routing-uloop] tunnel traffic with IGP-Adjacency Segment SID's or when ECMP load balancing should occur with Anycast-SID's.

4. IANA Considerations

This document specifies three additional code points for IS-IS, OSPv2 and OSPFv3 Segment Routing extension in the existing sub-registry "IPFIX MPLS label type (Value 46)" of the "IPFIX Information Elements" and one new "IPFIX Information Element" with a new sub-registry in the "IP Flow Information Export (IPFIX) Entities" name space.

Value	Description	Reference
TBD1	OSPFv2 Segment Routing	RFC8665
TBD2	OSPFv3 Segment Routing	RFC8666
TBD3	IS-IS Segment Routing	RFC8667

Figure 1: Updates to "IPFIX Information Element #46" SubRegistry

ElementID	Name	Abstract Data Type	Data Type Semantics	Description	Reference
TBD4	SrSidType	unsigned8	identifier	This field identifies the Segment Routing Identifier Type of the MPLS top-of-stack label. SID types for this field are listed in the SR SID type registry.	RFC8402

Figure 2: New "IPFIX Information Element #TBD4"

Value	Description	Reference
TBD5	Unknown SID Type	RFC8402
TBD6	Prefix-SID	RFC8402
TBD7	Node-SID	RFC8402
TBD8	Anycast-SID	RFC8402
TBD9	Adjacency-SID	RFC8402
TBD10	LAN-Adjacency-SID	RFC8402
TBD11	PeerNode-SID	RFC8402
TBD12	PeerAdj-SID	RFC8402
TBD13	PeerSet-SID	RFC8402
TBD14	Binding-SID	RFC8402

Figure 3: New "IPFIX Information Element #TBD4" SubRegistry

5. Security Considerations

The same security considerations apply as for the IPFIX Protocol RFC7012 [RFC7012].

6. Acknowledgements

I would like to thank Paul Aitken, Loa Andersson, Tianran Zhou, Pierre Francois, Bruno Decreane and Paolo Lucente for their review and valuable comments.

7. References

7.1. Normative References

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <https://www.rfc-editor.org/info/rfc7012>.

7.2. Informative References

[I-D.ali-spring-sr-traffic-accounting]
Filsfils, C., Talaulikar, K., Sivabalan, S., Horneffer, M., Raszuk, R., Litkowski, S., Voyer, D., and R. Morton, "Traffic Accounting in Segment Routing Networks", draft-ali-spring-sr-traffic-accounting-04 (work in progress), February 2020.

[I-D.bashandy-rtgwg-segment-routing-uloop]
Bashandy, A., Filsfils, C., Litkowski, S., Decraene, B., Francois, P., and P. Psenak, "Loop avoidance using Segment Routing", draft-bashandy-rtgwg-segment-routing-uloop-08 (work in progress), January 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-03 (work in progress), March 2020.

[IANA-IPFIX-IE46]
"IANA IP Flow Information Export (IPFIX) Information Element #46 SubRegistry", <https://www.iana.org/assignments/ipfix/ipfix.xhtml#ipfix-mpls-label-type>.

[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>.

[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>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[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>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

Author's Address

Thomas Graf
Swisscom
Binzring 17
Zurich 8045
Switzerland

Email: thomas.graf@swisscom.com

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

T. Graf
Swisscom

September 12, 2020

Export of MPLS Segment Routing Label Type Information in IP Flow Information Export (IPFIX) draft-tgraf-ipfix-mpls-sr-label-type-05

Abstract

This document introduces additional code points in the mplsTopLabelType Information Element for IS-IS, OSPFv2, OSPFv3 and BGP MPLS Segment Routing (SR) extensions to enable Segment Routing label protocol type information in IP Flow Information Export (IPFIX).

Status of This Memo

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

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

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

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

Table of Contents

1. Introduction

Besides BGP-4 [RFC8277], LDP [RFC5036] and BGP VPN [RFC4364], four new routing-protocols, OSPFv2 Extensions [RFC8665], OSPFv3 Extensions [RFC8666], IS-IS Extensions [RFC8667] and BGP Prefix-SID [RFC8669] have been added to the list of routing-protocols able to propagate Segment Routing labels for the MPLS dataplane [RFC8660].

Traffic Accounting in Segment Routing Networks [I-D.ali-spring-sr-traffic-accounting] describes how IPFIX can be leveraged to account traffic to MPLS Segment Routing label dimensions within a Segment Routing domain.

In the Information Model for IP Flow Information Export IPFIX [RFC7012], the information element #46 mplsTopLabelType describes which MPLS control plane protocol allocated the top-of-stack label in the MPLS label stack. RFC 7012 section 7.2 [RFC7012] describes the IANA Information Element #46 SubRegistry [IANA-IPFIX-IE46] where new code points should be added.

2. MPLS Segment Routing Top Label Type

By introducing four new code points to information element #46 mplsTopLabelType for IS-IS, OSPFv2, OSPFv3 and BGP Prefix-SID, when Segment Routing with one of these four routing protocols is deployed, we get insight into which traffic is being forwarded based on which MPLS control plane protocol.

A typical use case scenario is to monitor MPLS control plane migrations from LDP to IS-IS or OSPF Segment Routing. Such a migration can be done node by node as described in RFC8661 [RFC8661]

Another use case is the monitoring of a migration to a Seamless MPLS SR [I-D.hegde-spring-mpls-seamless-sr] architecture. Where prefixes are propagated with dynamic BGP labels according to RFC8277

[RFC8277], BGP Prefix-SID according to RFC8669 [RFC8669] and used for the forwarding between IGP domains. Adding an additional layer into the MPLS dataplane to above discribed use case.

Both use cases can be verified by looking at IE46, mplsTopLabelType, IE47, mplsTopLabelIPv4Address, IE70, mplsTopLabelStackSection and IE89 forwardingStatus dimensions. Giving insights into the MPLS dataplane for which MPLS provider edge loopback address, which label protocol has been used and how many packets are forwarded or dropped and when dropped why they have been dropped.

By looking at the MPLS label value itself, it is not always clear as to which label protocol it belongs, since they could potentially share the same label allocation range. This is the case for IGP-Adjacency SID's, LDP and dynamic BGP labels as an example.

3. IANA Considerations

This document specifies four additional code points for IS-IS, OSPFv2, OSPFv3 and BGP Prefix-SID Segment Routing extension in the existing sub-registry "IPFIX MPLS label type (Value 46)" of the "IPFIX Information Elements" and one new "IPFIX Information Element" with a new sub-registry in the "IP Flow Information Export (IPFIX) Entities" name space.

Value	Description	Reference
TBD1	OSPFv2 Segment Routing	RFC8665
TBD2	OSPFv3 Segment Routing	RFC8666
TBD3	IS-IS Segment Routing	RFC8667
TBD4	BGP Segment Routing Prefix-SID	RFC8669

Figure 1: Updates to "IPFIX Information Element #46" SubRegistry

4. Security Considerations

The same security considerations apply as for the IPFIX Protocol RFC7012 [RFC7012].

5. Acknowledgements

I would like to thank Paul Aitken, Loa Andersson, Tianran Zhou, Pierre Francois, Bruno Decreane, Paolo Lucente, Hannes Gredler, Ketan Talaulikar, Sabrina Tanamal, Erik Auerswald and Sergey Fomin for their review and valuable comments.

6. References

6.1. Normative References

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <https://www.rfc-editor.org/info/rfc7012>.

6.2. Informative References

[I-D.ali-spring-sr-traffic-accounting]
Filsfils, C., Talaulikar, K., Sivabalan, S., Horneffer, M., Raszuk, R., Litkowski, S., Voyer, D., and R. Morton, "Traffic Accounting in Segment Routing Networks", draft-ali-spring-sr-traffic-accounting-04 (work in progress), February 2020.

[I-D.hegde-spring-mpls-seamless-sr]
Hegde, S., Bowers, C., Xu, X., Gulko, A., Bogdanov, A., and J. Uttaro, "Seamless Segment Routing", draft-hegde-spring-mpls-seamless-sr-01 (work in progress), July 2020.

[IANA-IPFIX-IE46]
"IANA IP Flow Information Export (IPFIX) Information Element #46 SubRegistry", <https://www.iana.org/assignments/ipfix/ipfix.xhtml#ipfix-mpls-label-type>.

[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>.

[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>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[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>.

[RFC8661] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing MPLS Interworking with LDP", RFC 8661, DOI 10.17487/RFC8661, December 2019, <https://www.rfc-editor.org/info/rfc8661>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

Author's Address

Thomas Graf
Swisscom
Binzring 17
Zurich 8045
Switzerland

Email: thomas.graf@swisscom.com