SPRING
Internet-Draft
Intended status: Informational
Expires: April 25, 2019

Z. Ali
K. Talaulikar
C. Filsfils
N. Nainar
C. Pignataro
Cisco Systems

October 22, 2018

Bidirectional Forwarding Detection (BFD) for Segment Routing Policies for Traffic Engineering
draft-ali-spring-bfd-sr-policy-02

Abstract

Segment Routing (SR) allows a headend node to steer a packet flow along any path using a segment list which is referred to as a SR Policy. Intermediate per-flow states are eliminated thanks to source routing. The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy. Bidirectional Forwarding Detection (BFD) is used to monitor different kinds of paths between node. BFD mechanisms can be also used to monitor the availability of the path indicated by a SR Policy and to detect any failures. Seamless BFD (S-BFD) extensions provide a simplified mechanism which is suitable for monitoring of paths that are setup dynamically and on a large scale.

This document describes the use of Seamless BFD (S-BFD) mechanism to monitor the SR Policies that are used for Traffic Engineering (TE) in SR deployments.

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 April 25, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) ([RFC8402]) allows a headend node to steer a packet flow along any path for specific objectives like Traffic Engineering (TE) and to provide it treatment according to the specific established service level agreement (SLA) for it. 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 the ordered list of segments associated with that SR Policy. SR Policy [I-D.ietf-spring-segment-routing-policy] specifies the concepts of SR Policy and steering into an SR Policy.

SR Policy state is instantiated only on the head-end node and any intermediate node or the endpoint node does not require any state to be maintained or instantiated for it. SR Policies are not signaled through the network nodes except the signaling required to instantiate them on the head-end in the case of a controller based deployment. This enables SR Policies to scale far better than previous TE mechanisms. This also enables SR Policies to be instantiated dynamically and on demand basis for steering specific traffic flows corresponding to service routes as they are signaled. These automatic steering and signaling mechanisms for SR Policies are described in SR Policy [I-D.ietf-spring-segment-routing-policy].

There is a requirement to continuously monitor the availability of the path corresponding to the SR Policy along the nodes in the network to rapidly detect any failures in the forwarding path so that it could take corrective action to restore service. The corrective actions may be either to invalidate the candidate path that has experienced failure and to switch to another candidate path within the same SR Policy OR to activate another backup SR Policy or candidate path for end-to-end path protection. These mechanisms are beyond the scope of this document.

Bidirectional Forwarding Detection (BFD) mechanisms have been specified for use for monitoring of unidirectional MPLS LSPs via BFD MPLS [RFC5884]. Seamless BFD [RFC7880] defines a simplified mechanism for using BFD by eliminating the negotiation aspect and the need to maintain per session state entries on the tail end of the policy, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating path monitoring. When BFD or S-BFD is used for verification of such unidirectional LSP paths, the reverse path is via the shortest path from the tail-end router back to the head-end router as determined by routing.

The SR Policy is essentially a unidirectional path through the network. This document describes the use of BFD and more specifically S-BFD for monitoring of SR Policy paths through the network. SR can be instantiated using both MPLS and IPv6 dataplanes. The mechanism described in this document applies to both these instantiations of SR Policy.

2. Choice of S-BFD over BFD

BFD MPLS [RFC5884] describes a mechanism where LSP Ping [RFC8029] is used to bootstrap the BFD session over an MPLS TE LSP path. The LSP Ping mechanism was extended to support SR LSPs via SR LSP Ping [RFC8287] and a similar mechanism could have been considered for BFD monitoring of SR Policies on MPLS data-plane. However, this document proposes instead to use S-BFD mechanism as it is more suitable for SR Policies.

Some of the key aspects of SR Policies that are considered in arriving at this decision are as follows:

- SR Policies do not require any signaling to be performed through the network nodes in order to be setup. They are simply instantiated on the head-end node via provisioning or even dynamically by a controller via BGP SR-TE [I-D.ietf-idr-segment-routing-te-policy] or using PCEP (PCEP SR [I-D.ietf-pce-segment-routing], PCE Initiated [RFC8281], PCEP Stateful [RFC8231]).

- SR Policies result in state being instantiated only on the head-end node and no other node in the network.

- In many deployments, SR Policies are instantiated dynamically and on-demand or in the case of automated steering for BGP routes, when routes are learnt with specific color communities (refer SR Policy [I-D.ietf-spring-segment-routing-policy] for details).

- SR Policies are expected to be deployed in much higher scale.

- SR Policies can be instantiated both for MPLS and IPv6 data-planes and hence a monitoring mechanism which works for both is desirable.

In view of the above, the BFD mechanism to be used for monitoring them needs to be simple, lightweight, one that does not result in instantiation of per SR Policy state anywhere but the head-end and which can be setup and deleted dynamically and on-demand. The S-BFD extensions provide this support as described in Seamless BFD [RFC7880]. Furthermore, S-BFD Use-Cases [RFC7882] clarifies the applicability in the Centralized TE and SR scenarios.

3. Procedures

The general procedures and mechanisms for S-BFD operations are specified in Seamless BFD [RFC7880]. This section describes the specifics related to S-BFD use for SR Policies.

SR Policies are represented on a head-end router as <color,endpoint IP address> tuple. The SRTE process on the head-end determines the tail-end node of a SR Policy on the basis of the endpoint IP address. In the cases where the SR Policy endpoint is outside the domain of the head-end node, this information is available with the centralized controller that computed the multi-domain SR Policy path for the head-end.

3.1. S-BFD Discriminator

In order to enable S-BFD monitoring for a given SR Policy, the S-BFD Discriminator for the tail-end node (i.e. one with the endpoint IP address) which is going to be the S-BFD Reflector is required. ISIS S-BFD [RFC7883] and OSPF S-BFD [RFC7884] describe the extensions to the ISIS and OSPF link state routing protocols that allow all nodes to advertise their S-BFD Discriminators across the network. BGP-LS S-BFD [I-D.li-idr-bgp-ls-sbfd-extensions] describes extensions for advertising the S-BFD discriminators via BGP-LS across domains and to a controller. Thus, either the SRTE head-end node or the controller, as the case may be, have the S-BFD Discriminator of the tail-end node of the SR Policy available.

When the end point IP address configured in the SR policy is IPv4, an implementation may support the use of end point address as the S-BFD Discriminator if SBFDReflector is enabled to associate the end point address as Discriminator for the target identifier.

The selection of S-BFD Discriminator from IGP or end point address is a local implementation matter and can be controlled by configuration knob.

3.2. S-BFD session Initiation by SBFDInitiator

The SRTE Process can straightaway instantiate the S-BFD mechanism on the SR Policy as soon as it is provisioned in the forwarding to start verification of the path to the endpoint. No signaling or provisioning is required for the tail-end node on a per SR Policy basis and it just performs its role as a stateless S-BFD Reflector. The return path used by S-BFD is via the normal IP routing back to the head-end node. Once the specific SR Policy path is verified via S-BFD, then it is considered as active and may be used for traffic steering.

The S-BFD monitoring continues for the SR Policy and any failure is notified to the SRTE process. In response to the failure of a specific candidate path, the SRTE process may trigger any of the following based on local policy or implementation specific aspects which are outside the scope of this document:

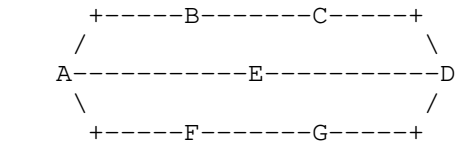
- Trigger path-protection for the SR Policy

- Declare the specific candidate path as invalid and switch to using the next valid candidate path based on preference

- If no alternate candidate path is available, then handle the steering over that SR Policy based on its invalidation policy (e.g. drop or switch to best effort routing).

3.3. Controlled Return Path

S-BFD response from SBFDResponder is IP routed and so the procedure defined in the above sections will receive the response through uncontrolled return path. S-BFD echo packets with relevant stack of segment ID can be used to control the return path.



Forward Paths: A-B-C-D
IP Return Paths: D-E-A

Figure 1: S-BFD Echo Example

Node A sending S-BFD control packets with segment stack {B, C, D} will cause S-BFD control packets to traverse the paths A-B-C-D in the forward direction. The response S-BFD control packets from node D back to node A will be IP routed and will traverse the paths D-E-A. The SBFDInitiator sending such packets can also send S-BFD echo packets with segment stack {B, C, D, C, A}. S-BFD echo packets will u-turn on node D and traverse the paths D-C-B-A. If required, the SBFDInitiator can possess multiple types of S-BFD echo packets, with each having varying return paths. In this particular example, the SBFDInitiator can be sending two types of S-BFD echo packets in addition to S-BFD control packets.

- S-BFD Control Packets
  - Segment Stack: {B, C, D}
  - Return Path: D->E->A
- S-BFD Echo packets #1

* Segment Stack: {B, C, D, C, A}

* Return Path: D->C->B->A

o S-BFD Echo packets #2

* Segment Stack: {B, C, D, G, A}

* Return Path: D->G->F->A

The SBFDInitiator can correlate the result of each packet type to determine the nature of the failure. One such example of failure correlation is described in the figure below.

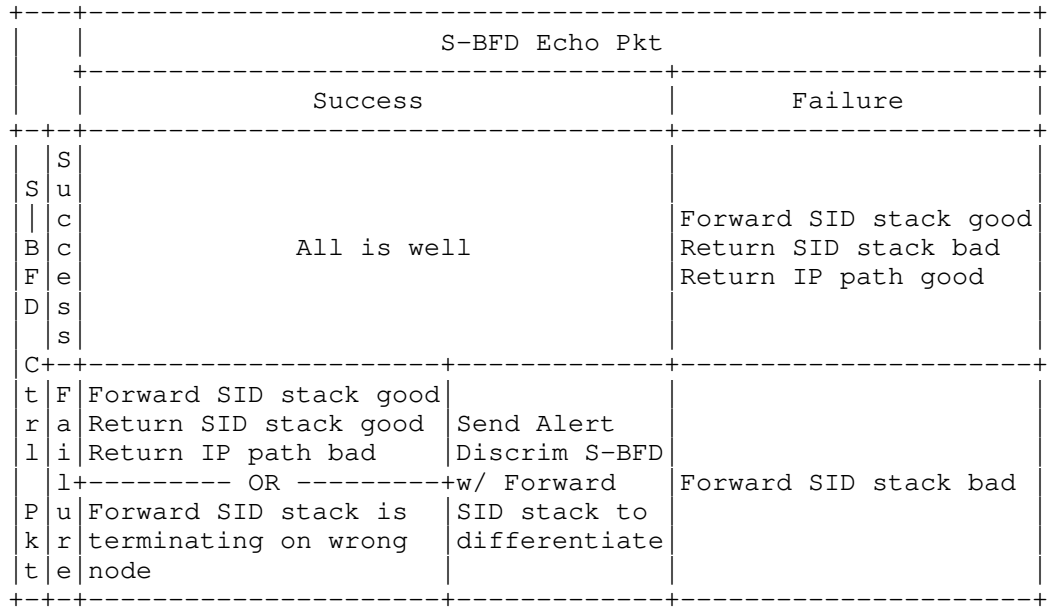


Figure 2: SBFDInitiator Failure Correlation Example

3.4. S-BFD Echo Recommendation

o It is RECOMMENDED to compute and use smallest number of segment stack to describe the return path of S-BFD echo packets to prevent the segment stack being too large. How SBFDInitiator determines when to use S-BFD echo packets and how to identify corresponding

segment stack for the return paths are outside the scope of this document.

- It is RECOMMENDED that SBFDInitiator does not send only S-BFD echo packets. S-BFD echo packets are crafted to traverse the network and to come back to self, thus there is no guarantee that S-BFD echo are u-turning on the intended remote target. On the other hand, S-BFD control packets can verify that segment stack of the forward direction reaches the intended remote target. Therefore, an SBFDInitiator SHOULD send S-BFD control packets when sending S-BFD echo packets.

4. IANA Considerations

None

5. Security Considerations

Procedures described in this document do not affect the BFD or Segment Routing security model. See the 'Security Considerations' section of [RFC7880] for a discussion of S-BFD security and to [RFC8402] for analysis of security in SR deployments.

6. Contributors

Mallik Mudigonda
Cisco Systems Inc.

Email: mmudigon@cisco.com

7. Acknowledgements

8. References

8.1. Normative References

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-02 (work in progress), October 2018.

[I-D.li-idr-bgp-ls-sbfd-extensions]
Li, Z., Aldrin, S., Tantsura, J., Mirsky, G., Zhuang, S., and K. Talaulikar, "BGP Link-State Extensions for Seamless BFD", draft-li-idr-bgp-ls-sbfd-extensions-02 (work in progress), June 2018.

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

[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>.

[RFC7882] Aldrin, S., Pignataro, C., Mirsky, G., and N. Kumar, "Seamless Bidirectional Forwarding Detection (S-BFD) Use Cases", RFC 7882, DOI 10.17487/RFC7882, July 2016, <https://www.rfc-editor.org/info/rfc7882>.

[RFC7883] Ginsberg, L., Akiya, N., and M. Chen, "Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS", RFC 7883, DOI 10.17487/RFC7883, July 2016, <https://www.rfc-editor.org/info/rfc7883>.

[RFC7884] Pignataro, C., Bhatia, M., Aldrin, S., and T. Ranganath, "OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators", RFC 7884, DOI 10.17487/RFC7884, July 2016, <https://www.rfc-editor.org/info/rfc7884>.

[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>.

8.2. Informative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Jain, D., Mattes, P., Rosen, E., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-04 (work in progress), July 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-14 (work in progress), October 2018.

[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>.

[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>.

[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>.

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

Authors' Addresses

Zafar Ali
Cisco Systems

Email: zali@cisco.com

Ketan Talaulikar
Cisco Systems

Email: ketant@cisco.com

Clarence Filsfils
Cisco Systems

Email: cfilsfil@cisco.com

Nagendra Kumar Nainar
Cisco Systems

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

SPRING
Internet-Draft
Intended status: Informational
Expires: May 15, 2021

Z. Ali
K. Talaulikar
C. Filsfils
N. Nainar
C. Pignataro
Cisco Systems

November 16, 2020

Bidirectional Forwarding Detection (BFD) for Segment Routing Policies for Traffic Engineering
draft-ali-spring-bfd-sr-policy-06

Abstract

Segment Routing (SR) allows a headend node to steer a packet flow along any path using a segment list which is referred to as a SR Policy. Intermediate per-flow states are eliminated thanks to source routing. The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy. Bidirectional Forwarding Detection (BFD) is used to monitor different kinds of paths between node. BFD mechanisms can be also used to monitor the availability of the path indicated by a SR Policy and to detect any failures. Seamless BFD (S-BFD) extensions provide a simplified mechanism which is suitable for monitoring of paths that are setup dynamically and on a large scale.

This document describes the use of Seamless BFD (S-BFD) mechanism to monitor the SR Policies that are used for Traffic Engineering (TE) in SR deployments.

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 May 15, 2021.

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) ([RFC8402]) allows a headend node to steer a packet flow along any path for specific objectives like Traffic Engineering (TE) and to provide it treatment according to the specific established service level agreement (SLA) for it. 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 the ordered list of segments associated with that SR Policy. SR Policy [I-D.ietf-spring-segment-routing-policy] specifies the concepts of SR Policy and steering into an SR Policy.

SR Policy state is instantiated only on the head-end node and any intermediate node or the endpoint node does not require any state to be maintained or instantiated for it. SR Policies are not signaled through the network nodes except the signaling required to instantiate them on the head-end in the case of a controller based deployment. This enables SR Policies to scale far better than previous TE mechanisms. This also enables SR Policies to be instantiated dynamically and on demand basis for steering specific traffic flows corresponding to service routes as they are signaled. These automatic steering and signaling mechanisms for SR Policies are described in SR Policy [I-D.ietf-spring-segment-routing-policy].

There is a requirement to continuously monitor the availability of the path corresponding to the SR Policy along the nodes in the network to rapidly detect any failures in the forwarding path so that it could take corrective action to restore service. The corrective actions may be either to invalidate the candidate path that has experienced failure and to switch to another candidate path within the same SR Policy OR to activate another backup SR Policy or candidate path for end-to-end path protection. These mechanisms are beyond the scope of this document.

Bidirectional Forwarding Detection (BFD) mechanisms have been specified for use for monitoring of unidirectional MPLS LSPs via BFD MPLS [RFC5884]. Seamless BFD [RFC7880] defines a simplified mechanism for using BFD by eliminating the negotiation aspect and the need to maintain per session state entries on the tail end of the policy, thus providing benefits such as quick provisioning, as well as improved control and flexibility for network nodes initiating path monitoring. When BFD or S-BFD is used for verification of such unidirectional LSP paths, the reverse path is via the shortest path from the tail-end router back to the head-end router as determined by routing.

The SR Policy is essentially a unidirectional path through the network. This document describes the use of BFD and more specifically S-BFD for monitoring of SR Policy paths through the network. SR can be instantiated using both MPLS and IPv6 dataplanes. The mechanism described in this document applies to both these instantiations of SR Policy.

2. Choice of S-BFD over BFD

BFD MPLS [RFC5884] describes a mechanism where LSP Ping [RFC8029] is used to bootstrap the BFD session over an MPLS TE LSP path. The LSP Ping mechanism was extended to support SR LSPs via SR LSP Ping [RFC8287] and a similar mechanism could have been considered for BFD monitoring of SR Policies on MPLS data-plane. However, this document proposes instead to use S-BFD mechanism as it is more suitable for SR Policies.

Some of the key aspects of SR Policies that are considered in arriving at this decision are as follows:

- SR Policies do not require any signaling to be performed through the network nodes in order to be setup. They are simply instantiated on the head-end node via provisioning or even dynamically by a controller via BGP SR-TE [I-D.ietf-idr-segment-routing-te-policy] or using PCEP (PCEP SR [I-D.ietf-pce-segment-routing], PCE Initiated [RFC8281], PCEP Stateful [RFC8231]).

- SR Policies result in state being instantiated only on the head-end node and no other node in the network.

- In many deployments, SR Policies are instantiated dynamically and on-demand or in the case of automated steering for BGP routes, when routes are learnt with specific color communities (refer SR Policy [I-D.ietf-spring-segment-routing-policy] for details).

- SR Policies are expected to be deployed in much higher scale.

- SR Policies can be instantiated both for MPLS and IPv6 data-planes and hence a monitoring mechanism which works for both is desirable.

In view of the above, the BFD mechanism to be used for monitoring them needs to be simple, lightweight, one that does not result in instantiation of per SR Policy state anywhere but the head-end and which can be setup and deleted dynamically and on-demand. The S-BFD extensions provide this support as described in Seamless BFD [RFC7880]. Furthermore, S-BFD Use-Cases [RFC7882] clarifies the applicability in the Centralized TE and SR scenarios.

3. Procedures

The general procedures and mechanisms for S-BFD operations are specified in Seamless BFD [RFC7880]. This section describes the specifics related to S-BFD use for SR Policies.

SR Policies are represented on a head-end router as <color,endpoint IP address> tuple. The SRTE process on the head-end determines the tail-end node of a SR Policy on the basis of the endpoint IP address. In the cases where the SR Policy endpoint is outside the domain of the head-end node, this information is available with the centralized controller that computed the multi-domain SR Policy path for the head-end.

3.1. S-BFD Discriminator

In order to enable S-BFD monitoring for a given SR Policy, the S-BFD Discriminator for the tail-end node (i.e. one with the endpoint IP address) which is going to be the S-BFD Reflector is required. ISIS S-BFD [RFC7883] and OSPF S-BFD [RFC7884] describe the extensions to the ISIS and OSPF link state routing protocols that allow all nodes to advertise their S-BFD Discriminators across the network. BGP-LS S-BFD [I-D.ietf-idr-bgp-ls-sbfd-extensions] describes extensions for advertising the S-BFD discriminators via BGP-LS across domains and to a controller. Thus, either the SRTE head-end node or the controller, as the case may be, have the S-BFD Discriminator of the tail-end node of the SR Policy available.

When the end point IP address configured in the SR policy is IPv4, an implementation may support the use of end point address as the S-BFD Discriminator if SBFDReflector is enabled to associate the end point address as Discriminator for the target identifier.

The selection of S-BFD Discriminator from IGP or end point address is a local implementation matter and can be controlled by configuration knob.

3.2. S-BFD session Initiation by SBFDInitiator

The SRTE Process can straightaway instantiate the S-BFD mechanism on the SR Policy as soon as it is provisioned in the forwarding to start verification of the path to the endpoint. No signaling or provisioning is required for the tail-end node on a per SR Policy basis and it just performs its role as a stateless S-BFD Reflector. The return path used by S-BFD is via the normal IP routing back to the head-end node. Once the specific SR Policy path is verified via S-BFD, then it is considered as active and may be used for traffic steering.

The S-BFD monitoring continues for the SR Policy and any failure is notified to the SRTE process. In response to the failure of a specific candidate path, the SRTE process may trigger any of the following based on local policy or implementation specific aspects which are outside the scope of this document:

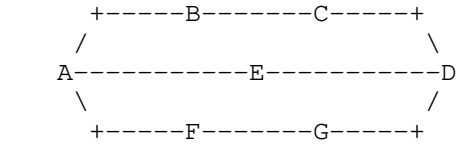
- Trigger path-protection for the SR Policy

- Declare the specific candidate path as invalid and switch to using the next valid candidate path based on preference

- If no alternate candidate path is available, then handle the steering over that SR Policy based on its invalidation policy (e.g. drop or switch to best effort routing).

3.3. Controlled Return Path

S-BFD response from SBFDResponder is IP routed and so the procedure defined in the above sections will receive the response through uncontrolled return path. S-BFD echo packets with relevant stack of segment ID can be used to control the return path.



Forward Paths: A-B-C-D
IP Return Paths: D-E-A

Figure 1: S-BFD Echo Example

Node A sending S-BFD control packets with segment stack {B, C, D} will cause S-BFD control packets to traverse the paths A-B-C-D in the forward direction. The response S-BFD control packets from node D back to node A will be IP routed and will traverse the paths D-E-A. The SBFDInitiator sending such packets can also send S-BFD echo packets with segment stack {B, C, D, C, A}. S-BFD echo packets will u-turn on node D and traverse the paths D-C-B-A. If required, the SBFDInitiator can possess multiple types of S-BFD echo packets, with each having varying return paths. In this particular example, the SBFDInitiator can be sending two types of S-BFD echo packets in addition to S-BFD control packets.

- S-BFD Control Packets

  * Segment Stack: {B, C, D}

  * Return Path: D->E->A

- S-BFD Echo packets #1

* Segment Stack: {B, C, D, C, A}

* Return Path: D->C->B->A

o S-BFD Echo packets #2

* Segment Stack: {B, C, D, G, A}

* Return Path: D->G->F->A

The SBFDInitiator can correlate the result of each packet type to determine the nature of the failure. One such example of failure correlation is described in the figure below.

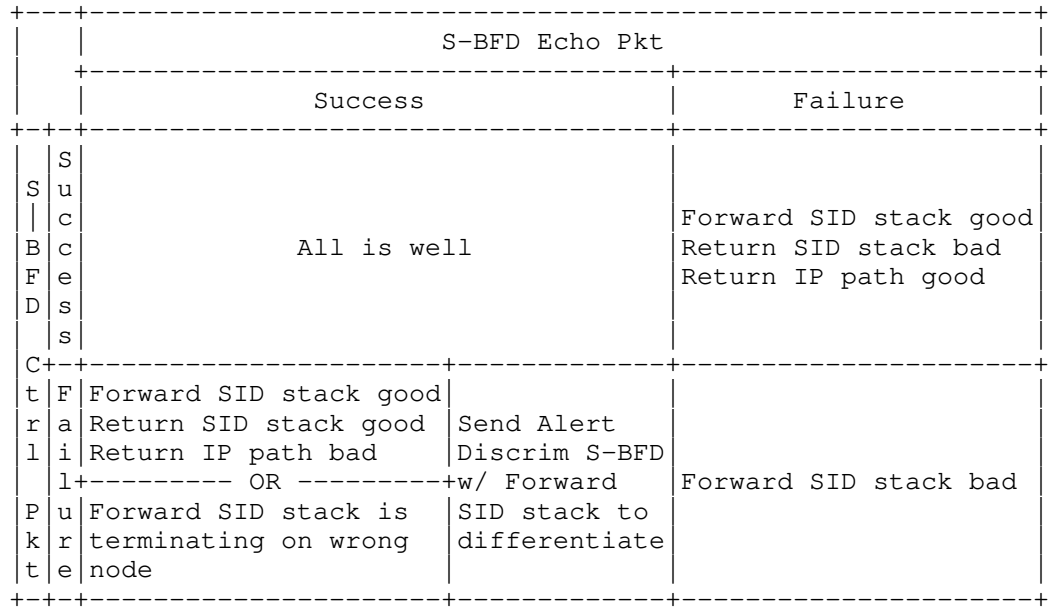


Figure 2: SBFDInitiator Failure Correlation Example

3.4. S-BFD Echo Recommendation

o It is RECOMMENDED to compute and use smallest number of segment stack to describe the return path of S-BFD echo packets to prevent the segment stack being too large. How SBFDInitiator determines when to use S-BFD echo packets and how to identify corresponding

segment stack for the return paths are outside the scope of this document.

- It is RECOMMENDED that SBFDInitiator does not send only S-BFD echo packets. S-BFD echo packets are crafted to traverse the network and to come back to self, thus there is no guarantee that S-BFD echo are u-turning on the intended remote target. On the other hand, S-BFD control packets can verify that segment stack of the forward direction reaches the intended remote target. Therefore, an SBFDInitiator SHOULD send S-BFD control packets when sending S-BFD echo packets.

4. IANA Considerations

None

5. Security Considerations

Procedures described in this document do not affect the BFD or Segment Routing security model. See the 'Security Considerations' section of [RFC7880] for a discussion of S-BFD security and to [RFC8402] for analysis of security in SR deployments.

6. Contributors

Mallik Mudigonda
Cisco Systems Inc.

Email: mmudigon@cisco.com

7. Acknowledgements

8. References

8.1. Normative References

[I-D.ietf-idr-bgp-ls-sbfd-extensions]
Li, Z., Zhuang, S., Talaulikar, K., Aldrin, S., Tantsura, J., and G. Mirsky, "BGP Link-State Extensions for Seamless BFD", draft-ietf-idr-bgp-ls-sbfd-extensions-02 (work in progress).

[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).

[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>.

[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>.

[RFC7882] Aldrin, S., Pignataro, C., Mirsky, G., and N. Kumar, "Seamless Bidirectional Forwarding Detection (S-BFD) Use Cases", RFC 7882, DOI 10.17487/RFC7882, July 2016, <https://www.rfc-editor.org/info/rfc7882>.

[RFC7883] Ginsberg, L., Akiya, N., and M. Chen, "Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS", RFC 7883, DOI 10.17487/RFC7883, July 2016, <https://www.rfc-editor.org/info/rfc7883>.

[RFC7884] Pignataro, C., Bhatia, M., Aldrin, S., and T. Ranganath, "OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators", RFC 7884, DOI 10.17487/RFC7884, July 2016, <https://www.rfc-editor.org/info/rfc7884>.

[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>.

8.2. Informative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-08 (work in progress)

[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).

[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>.

[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>.

[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>.

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

Authors' Addresses

Zafar Ali
Cisco Systems

Email: zali@cisco.com

Ketan Talaulikar
Cisco Systems

Email: ketant@cisco.com

Clarence Filsfils
Cisco Systems

Email: cfilsfil@cisco.com

Nagendra Kumar Nainar
Cisco Systems

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

Z. Ali
R. Gandhi
C. Filsfils
F. Brockners
N. Nainar
C. Pignataro
Cisco Systems, Inc.
C. Li
M. Chen
Huawei
G. Dawra
LinkedIn

October 22, 2018

Segment Routing Header encapsulation for In-situ OAM Data draft-ali-spring-ioam-srv6-00

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the data packet while the packet traverses a path between two points in the network. This document defines how IOAM data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header.

Status of This Memo

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

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

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

Copyright Notice

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

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

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

Table of Contents

1. Introduction

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

This document defines how IOAM data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header [I-D.6man-segment-routing-header].

The IOAM data fields carried are defined in [I-D.ietf-ippm-ioam-data], and can be used for various use-cases including Performance Measurement (PM) and Proof-of-Transit (PoT).

2. Conventions

2.1. Requirement Language

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

2.2. Abbreviations

Abbreviations used in this document:

IOAM	In-situ Operations, Administration, and Maintenance
OAM	Operations, Administration, and Maintenance
PM	Performance Measurement
PoT	Proof-of-Transit
SR	Segment Routing
SRH	SRv6 Header
SRv6	Segment Routing with IPv6 Data plane

3. IOAM Data Field Encapsulation in SRH

The SRv6 encapsulation header (SRH) is defined in [I-D.6man-segment-routing-header]. IOAM data fields are carried in the SRH, using a single SRH TLV. The different IOAM data fields defined in [I-D.ietf-ippm-ioam-data] are added as sub-TLVs.

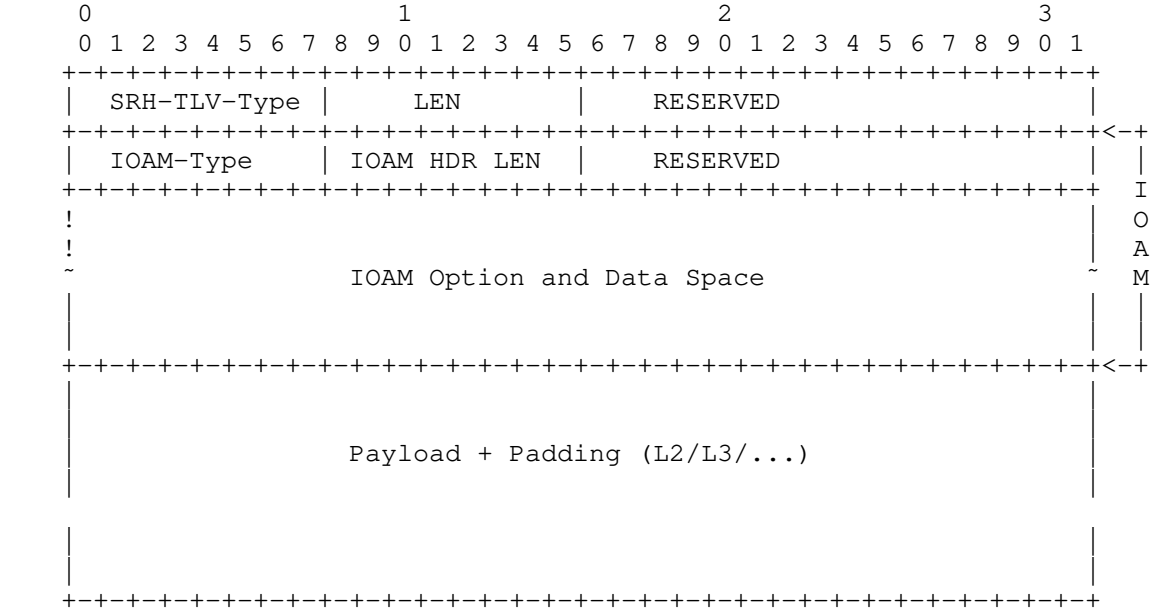


Figure 1: IOAM data encapsulation in SRH

SRH-TLV-Type: IOAM TLV Type for SRH is defined as TBA1.

The fields related to the encapsulation of IOAM data fields in the SRH are defined as follows:

IOAM-Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

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

RESERVED: 8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.

IOAM Option and Data Space: IOAM option header and data is present as defined by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

The IOAM TLVs MAY change en route [I-D.ietf-ippm-ioam-data]. For the IOAM TLVs carried in SRH that can change en route, the most significant bit of the SRH-TLV-Type is set [I-D.6man-segment-routing-header]. Furthermore, such IOAM TLV in SRH is considered mutable for ICV computation, the Type Length, and Variable Length Data is ignored for ICV Computation as defined in [RFC4302].

4. Procedure

This section summarizes the procedure for IOAM data encapsulation in SRv6 SRH. The SR nodes implementing the IOAM functionality follows the MTU and other considerations outlined in [I-D.6man-extension-header-insertion].

4.1. Ingress Node

The ingress node of an SR domain or an SR Policy [I-D.spring-segment-routing-policy] may insert the IOAM TLV in the SRH of the data packet. The ingress node may also insert the IOAM data about the local information in the IOAM TLV in the SRH. When IOAM data from the last node in the segment-list (Egress node) is desired, the ingress uses an Ultimate Segment Pop (USP) SID at the Egress node.

4.2. SR Segment Endpoint Node

The SR segment endpoint node is any node receiving an IPv6 packet where the destination address of that packet is a local SID or a local interface address. As part of the SR Header processing as described in [I-D.6man-segment-routing-header] and [I-D.spring-srv6-network-programming], the SR Segment Endpoint node performs the following IOAM operations. The description borrows the terminology used in [I-D.6man-segment-routing-header]. Specifically, n refers to the number of segments encoded in the SRH, "Hdr Ext Len" refers to the length of the SRH. The "SRH Header Len" is the length of the SRH header, which is 8 octets [I-D.6man-segment-routing-header].

The SR Segment Endpoint node compares the "Hdr Ext Len" of the SRH with the length of the "segment-list" in the SRH. Specifically, if the SRH.Hdr_Ext_Len > n*16 + 8, the node looks for the presence of the IOAM TLV in the SRH. If an IOAM TLV is present in the SRH and is supported by the Segment Endpoint Node, the SR segment endpoint node MAY modify the IOAM TLV in SRH with local IOAM data as per IOAM draft [I-D.ietf-ippm-ioam-data].

4.3. Egress Node

The Egress node is the last node in the segment-list of the SRH. When IOAM data from the Egress node is desired, a USP SID advertised by the Egress node is used.

The processing of IOAM TLV at the Egress node is similar to the processing of IOAM TLV at the SR Segment Endpoint Node. The only difference is that the Egress node also performs the functionality required by the Egress node in an IOAM domain. E.g., the Egress node may telemeter the IOAM data to a controller.

5. IANA Considerations

IANA is requested to allocate SRH TLV Type for IOAM TLV data fields under registry name "Segment Routing Header TLVs" requested by %[I-D.6man-segment-routing-header].

SRH TLV Type	Description	Reference
TBA1	TLV for IOAM Data Fields	This document

6. Security Considerations

The security considerations of SRv6 are discussed in [I-D.spring-srv6-network-programming] and [I-D.6man-segment-routing-header], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

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

7. Acknowledgements

The authors would like to thank Shwetha Bhandari and Vengada Prasad Govindan for the discussions on IOAM.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

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

[I-D.spring-srv6-network-programming] Filsfils, C. et al. "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming, work in progress.

[I-D.6man-segment-routing-header] Previdi, S., Filsfils, C. et al, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and Bernier, D., "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data, work in progress.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.

8.2. Informative References

[I-D.6man-extension-header-insertion] D. Voyer, et al., "Insertion of IPv6 Segment Routing Headers in a Controlled Domain", draft-voyer-6man-extension-header-insertion, work in progress.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Rakesh Gandhi
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Frank Brockners
Cisco Systems, Inc.
Germany

Email: fbrockne@cisco.com

Nagendra Kumar Nainar
Cisco Systems, Inc.

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.

Email: cpignata@cisco.com

Cheng Li
Huawei

Email: chengli13@huawei.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Gaurav Dawra
LinkedIn

Email: gdawra.ietf@gmail.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 14, 2021

Z. Ali
R. Gandhi
C. Filsfils
F. Brockners
N. Nainar
C. Pignataro
Cisco Systems, Inc.
C. Li
M. Chen
Huawei
G. Dawra
LinkedIn

November 15, 2020

Segment Routing Header encapsulation for In-situ OAM Data
draft-ali-spring-ioam-srv6-03

Abstract

OAM and PM information from the SR endpoints can be piggybacked in the data packet. The OAM and PM information piggybacking in the data packets is also known as In-situ OAM (IOAM). IOAM records operational and telemetry information in the data packet while the packet traverses a path between two points in the network. This document defines how IOAM data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header.

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 May 14, 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction

OAM and PM information from the SR endpoints can be piggybacked in the data packet. The OAM and PM information piggybacking in the data packets is also known as In-situ OAM (IOAM). IOAM records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than being sent within probe packets specifically dedicated to OAM.

This document defines how IOAM data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header [I-D.6man-segment-routing-header].

The IOAM data fields carried are defined in [I-D.ietf-ippm-ioam-data], and can be used for various use-cases including Performance Measurement (PM) and Proof-of-Transit (PoT).

2. Conventions

2.1. Requirement Language

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

2.2. Abbreviations

Abbreviations used in this document:

IOAM	In-situ Operations, Administration, and Maintenance
OAM	Operations, Administration, and Maintenance
PM	Performance Measurement
PoT	Proof-of-Transit
SR	Segment Routing
SRH	SRv6 Header
SRv6	Segment Routing with IPv6 Data plane

3. OAM Metadata Piggybacked in Data Packets

OAM and PM information from the SR endpoints can be piggybacked in the data packet. The OAM and PM information piggybacking in the data packets is also known as In-situ OAM (IOAM). This section describes iOAM functionality in SRv6 network.

The IOAM data is carried in SRH.TLV. This enables the IOAM mechanism to build on the network programmability capability of SRv6. Specifically, the ability for an SRv6 endpoint to determine whether to process or ignore some specific SRH TLVs is based on the SID function. This enables collection of the IOAM information hardware friendly based on the intermediate endpoint capability. The nodes that are not capable of supporting the IOAM functionality does not have to look or process SRH TLV (i.e., such nodes can simply ignore the SRH IOAM TLV). This also enable collection of IOAM data only from segment endpoint.

3.1 IOAM Data Field Encapsulation in SRH

The SRv6 encapsulation header (SRH) is defined in [I-D.ietf-6man-segment-routing-header]. IOAM data fields are carried in the SRH, using a single pre-allocated SRH TLV. The different IOAM data fields defined in [I-D.ietf-ippm-ioam-data] are added as sub-TLVs.

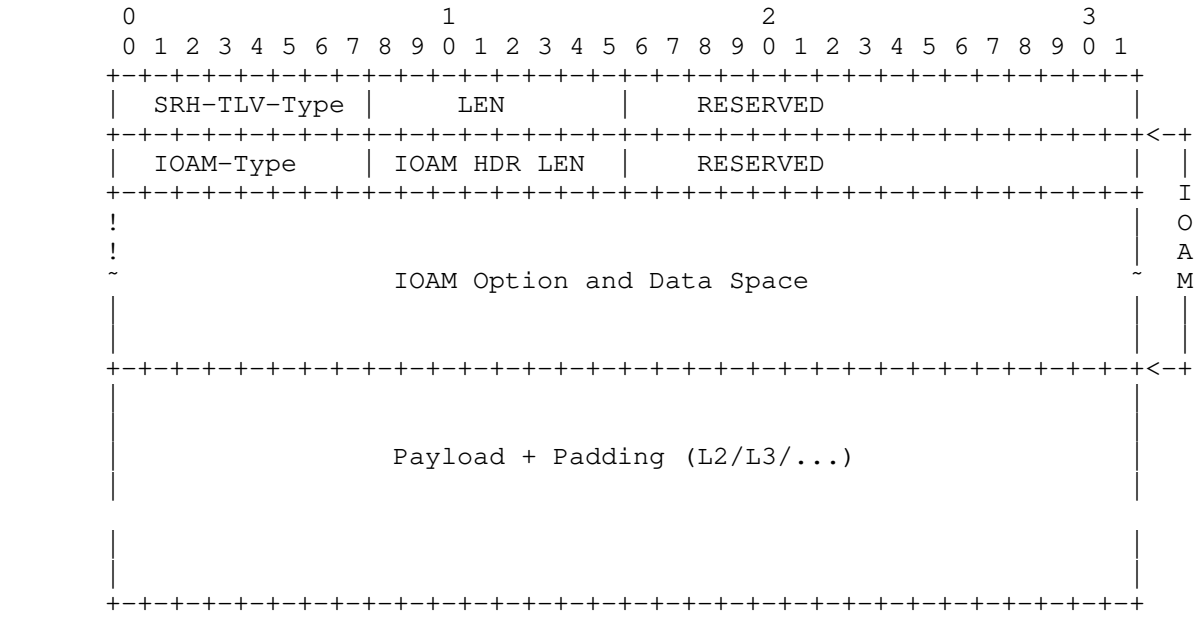


Figure 1: IOAM data encapsulation in SRH

SRH-TLV-Type: IOAM TLV Type for SRH is defined as TBA1.

The fields related to the encapsulation of IOAM data fields in the SRH are defined as follows:

IOAM-Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

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

RESERVED: 8-bit reserved field MUST be set to zero upon transmission and ignored upon receipt.

IOAM Option and Data Space: IOAM option header and data is present as defined by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

4. Procedure

This section summarizes the procedure for IOAM data encapsulation in SRv6 SRH. The SR nodes implementing the IOAM functionality follows the MTU and other considerations outlined in [I-D.6man-extension-header-insertion].

4.1. Ingress Node

As part of the SRH encapsulation, the ingress node of an SR domain or an SR Policy [I-D.ietf-spring-segment-routing-policy] MAY add the IOAM TLV in the SRH of the data packet. If an ingress node supports IOAM functionality and, based on a local configuration, wants to collect IOAM data, it adds IOAM TLV in the SRH. Based on the size of the segment list (SL), the ingress node preallocates space in the IOAM TLV.

If IOAM data from the last node in the segment-list (Egress node) is desired, the ingress uses an Ultimate Segment Pop (USP) SID advertised by the Egress node.

The ingress node MAY also insert the IOAM data about the local information in the IOAM TLV in the SRH at index 0 of the preallocated IOAM TLV.

4.2. Intermediate SR Segment Endpoint Node

The SR segment endpoint node is any node receiving an IPv6 packet where the destination address of that packet is a local SID. As part of the SR Header processing as described in [I-D.ietf-6man-segment-routing-header] and [I-D.ietf-spring-srv6-network-programming], the SR Segment Endpoint node performs the following IOAM operations.

If an intermediate SR segment endpoint node is not capable of processing IOAM TLV, it simply ignores it. I.e., it does not have to look or process SRH TLV.

If an intermediate SR segment endpoint node is capable of processing IOAM TLV and the local SID supports IOAM data recording, it checks if any SRH TLV is present in the packet using procedures defined in [I-D.ietf-6man-segment-routing-header]. If the node finds IOAM TLV in the SRH it finds the local index at which it is expected to record the IOAM data. The local index is found using the SRH.SL field. The node records the IOAM data at the desired preallocated space.

4.3. Egress Node

The Egress node is the last node in the segment-list of the SRH. When IOAM data from the Egress node is desired, a USP SID advertised by the Egress node is used by the Ingress node.

The processing of IOAM TLV at the Egress node is similar to the processing of IOAM TLV at the SR Segment Endpoint Node. The only difference is that the Egress node may telemeter the IOAM data to an external entity.

5. IANA Considerations

IANA is requested to allocate a mutable SRH TLV Type for IOAM TLV data fields under registry name "Segment Routing Header TLVs" requested by [I-D.6man-segment-routing-header].

SRH TLV Type	Description	Reference
TBA1 Greater than 128	TLV for IOAM Data Fields	This document

6. Security Considerations

The security considerations of SRv6 are discussed in [I-D.spring-srv6-network-programming] and [I-D.6man-segment-routing-header], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

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

7. Acknowledgements

The authors would like to thank Shwetha Bhandari and Vengada Prasad Govindan for the discussions on IOAM.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

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

[I-D.spring-srv6-network-programming] Filsfils, C. et al. "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming, work in progress.

[I-D.6man-segment-routing-header] Previdi, S., Filsfils, C. et al, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.

[I-D.ietf-ippm-ioam-data] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and Bernier, D., "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data, work in progress.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.

8.2. Informative References

[I-D.6man-extension-header-insertion] D. Voyer, et al., "Insertion of IPv6 Segment Routing Headers in a Controlled Domain", draft-voyer-6man-extension-header-insertion, work in progress.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Rakesh Gandhi
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Frank Brockners
Cisco Systems, Inc.
Germany

Email: fbrockne@cisco.com

Nagendra Kumar Nainar
Cisco Systems, Inc.

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.

Email: cpignata@cisco.com

Cheng Li
Huawei

Email: chengli13@huawei.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Gaurav Dawra
LinkedIn

Email: gdawra.ietf@gmail.com

spring
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2019

Z. Ali
C. Filsfils
N. Nainar
C. Pignataro
F. Clad
F. Iqbal
Cisco Systems, Inc.
X. Xu
Alibaba Inc.

October 22, 2018

OAM for Service Programming with Segment Routing
draft-ali-spring-sr-service-programming-oam-00

Abstract

This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction

[I-D.draft-xuclad-spring-sr-service-programming] defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture. This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

2. Requirements notation

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

3. Terminology

This document uses the terminologies defined in [I-D.ietf-spring-segment-routing], [I-D.filsfils-spring-srv6-network-programming] [I-D.xuclad-spring-sr-service-programming] and so the readers are expected to be familiar with the same.

4. Document Scope

The initial focus of this document to define and document the machinery required to apply OAM mechanisms on SRv6 based service programming.

Future version of this document will include the required details to apply OAM mechanism on other data planes.

5. Programmable OAM

Section 4 of [I-D.xuclad-spring-sr-service-programming] introduces Service segments and the procedure of service programming when the services are SR-aware and SR-unaware. By integrating the OAM functionality in the services, versatile OAM tool kits can be used to execute programmable OAM for service programming with Segment Routing.

This section describes the procedure to perform basic OAM mechanisms such as path validation and path tracing of Service Programming environment in Segment Routing network.

5.1. Service Programming OAM Packet Marker

Any services upon receiving OAM packet may apply the service treatment if it cannot differentiate the OAM packet from normal data packet. Depending on the service type, service treatment on OAM packet may result in dropping the OAM probe packet that may cause uncertainty in OAM mechanism.

To avoid such uncertainty, any node that is originating the OAM probe for Service Programming OAM MUST mark the packet as OAM packet so that the services can differentiate the OAM packet from data traffic.

5.2. OAM with SR-aware services

As defined in section 4.1 of [I-D.xuclad-spring-sr-service-programming], an SR-aware service can process the SR information in the packet header such as performing lookup or executing the next segment etc. An SR-aware service may

need to identify the packet payload and/or interpret SR information to apply the right policy to the received packet. While processing SR information in the packet header, it can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet.

An SR-aware service SHOULD skip applying the service on the OAM packet while forwarding the packet to the next segment or IP address. As defined in section 9, a local policy may be used to control any malicious use of OAM marker.

5.3. OAM with SR-unaware services

As defined in section 4.2 of [I-D.xuclad-spring-sr-service-programming], an SR-unaware service may be a legacy service that is not able to process the SR information in the packet header. SR Proxy, an entity that is external to the service is used to handle the SR information processing on behalf of the service. SR Proxy will remove the SR header before forwarding the packet to SR-unaware services to avoid any erroneous decision due to the presence of SR header that the service cannot recognize.

While processing SR information in the packet header, SR proxy can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet. SR Proxy MUST skip forwarding the packets with OAM marker to the service while forwarding the packet to the next segment or IP address. As defined in section 9, a local policy may be used to control any malicious use of OAM marker.

5.4. Controlling OAM packet processing in Services

As mentioned in the above sections, SR-aware service or the SR proxy can use the OAM marker to differentiate the OAM packet from data packet to skip the service treatment. Any intentional or unintentional use of OAM marker in data traffic may result in skipping service treatment for data traffic. To avoid such condition, a local policy will be used in the SR-aware service or SR Proxy that SHOULD rate limit or MAY drop the packets received with OAM marker.

6. OAM for Service Programming with SRv6

[I-D.draft-ietf-6man-segment-routing-header] defines SRH.Flags.O-bit in Segment Routing header. When service programming is implemented with SRv6 dataplane, SRH.Flags.O-bit is used as OAM marker. An IPv6 packet received with a local END.OP or END.OTP SID is also considered as an OAM packet.

Any node that is originating OAM probe to a service in SRv6 data plane MUST set SRH.Flags.O-bit in the SRH.

6.1. OAM Tool Kit

This section describes the availability of different tool kits that can be used to perform OAM functionality for Service Programming with SRv6 dataplane.

6.1.3. OAM Flag Processing

An SR-aware service or SR proxy MUST implement the SRH.Flags.O bit.
An SR-aware service SHOULD skip applying the service to the packet when SRH.Flags.O-bit is set and SHOULD forward the packet based on the next header.
SR Service Proxy MUST skip applying the service to the packet when SRH.Flags.O-bit is set and SHOULD forward the packet based on the next header.

An SR-aware service and SR proxy may choose to time-stamp and punt the packet with SRH.Flags.O-bit set for further processing and this is a local implementation matter.

6.1.4. OAM Segment ID

Section 3.2 of [[I-D.ali-spring-srv6-oam]] defines OAM segment ID and the associated forwarding semantics to implement the punt behavior for OAM packets. Specifically, the draft defines END.OP and END.OTP SIDs. An IPv6 packet received with DA set to a local END.OP or END.OTP SID is considered as an OAM packet.

Any service policy head end MAY include OAM segment ID in the desired segment list position of SRH. The inclusion of OAM SID in SRH can be used to control the services that are required to punt the OAM packet for processing.

6.1.3. ICMP

There is no hardware or software changes required to use ICMP for Ping operation. It can be triggered from the service policy head end or from any classical IPv6 nodes by sending ICMPv6 Echo Request. The existing ICMP Ping mechanism works seamlessly in SRv6 dataplane with no protocol changes required to the standard ICMPv6 [[RFC4443]], [[RFC4884]] or the standard ICMPv4 [[RFC0792]].

An SR-aware service SHOULD skip the service and forward to next segment based on the SR information in the packet header.
An SR Service Proxy MUST skip the service and forward to next segment based on the SR information in the packet header.

6.1.3.1. Pinging Service SID Function

When a remote node pings a service segment, it MUST set SRH.Flags.O = 1. If the target service segment is implemented with USP behavior, the ICMP packet can be constructed without adding END.OP or END.OTP SIDs defined in [I-D.draft-ali-spring-srv6-oam]. However, if the target service SID observes a PSP behavior, the sender needs to insert END.OP/ END.OTP SIDs before the target service SID in the segment-list. In either case, the target SR-aware service or SR proxy receives the ICMP echo request with either SRH.Flags.O-bit set or with the local END.OP or END.OTP SID. In both cases, the packet is punted for slow-path processing and service is skipped.

The Egress node process the packet as per procedure defined in [I-D.draft-ali-spring-srv6-oam]. The Egress checks if the target SID is locally programmed or not.

If the target SID is not locally programmed, the Egress responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned [I-D.draft-ali-spring-srv6-oam].

6.1.4. UDP Probes

A classic traceroute mechanism relies on UDP probes by sending packets with sequentially incrementing the TTL. More details are available in section 4.3.1 of [I-D.ali-spring-srv6-oam].

An SR-aware service or SR proxy upon receiving the probe with TTL=1, may follow the traditional behavior of replying with ICMPv6 Time Exceeded Message (Type 3) as defined in [[RFC4443]], without applying the service.

Use of SRH.Flags.O bit and END.OP/ END.OTP SIDs as OAM marker in the UDP probe for trace route is same as discussed for ICMPv6 ping discussed in the last section.

6.1.5. In-band OAM

To be Updated.

7. OAM for Service Programming with SR-MPLS

To be updated.

8. IANA Considerations

None.

9. Security Considerations

A local policy may be used to control any malicious use of OAM marker. More details are to be added in a future revision of the document.

10. Acknowledgement

Author would like to thank Bruno Decraene for his review and useful comments.

11. Normative References

[I-D.ali-spring-srv6-oam]
Ali, Z., Filsfils, C., Kumar, N., Pignataro, C., faiqbal@cisco.com, f., Gandhi, R., Leddy, J., Matsushima, S., Raszuk, R., daniel.voyer@bell.ca, d., Dawra, G., Peirens, B., Chen, M., and G. Naik, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ali-spring-srv6-oam-01 (work in progress), July 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-05 (work in progress), July 2018.

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

[I-D.xuclad-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-xuclad-spring-sr-service-programming-00 (work in progress), July 2018.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[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>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

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

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cfilsfils@cisco.com

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

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Francois Clad (editor)
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Faisal Iqbal
Cisco Systems, Inc.
2000 Innovation Dr
Ottawa, ON 3E8
Canada

Email: faiqbal@cisco.com

Xiaohu Xu (editor)
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

spring
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2020

Z. Ali
C. Filsfils
N. Nainar
C. Pignataro
F. Clad
F. Iqbal
Cisco Systems, Inc.
X. Xu
Alibaba

November 1, 2019

OAM for Service Programming with Segment Routing
draft-ali-spring-sr-service-programming-oam-02

Abstract

This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 4, 2020.

Copyright Notice

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

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

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

Table of Contents

1. Introduction

[I-D.ietf-spring-sr-service-programming] defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture. This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

2. Requirements notation

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

3. Terminology

This document uses the terminologies defined in [RFC8402], [I-D.filsfils-spring-srv6-network-programming] [I-D.ietf-spring-sr-service-programming] and so the readers are expected to be familiar with the same.

4. Document Scope

The initial focus of this document to define and document the machinery required to apply OAM mechanisms on SRv6 based service programming.

Future version of this document will include the required details to apply OAM mechanism on other data planes.

5. Programmable OAM

Section 4 of [I-D.ietf-spring-sr-service-programming] introduces Service segments and the procedure of service programming when the services are SR-aware and SR-unaware. By integrating the OAM functionality in the services, versatile OAM tool kits can be used to execute programmable OAM for service programming with Segment Routing.

This section describes the procedure to perform basic OAM mechanisms such as path validation and path tracing of Service Programming environment in Segment Routing network.

5.1. Service Programming OAM Packet Marker

Any services upon receiving OAM packet may apply the service treatment if it cannot differentiate the OAM packet from normal data packet. Depending on the service type, service treatment on OAM packet may result in dropping the OAM probe packet that may cause uncertainty in OAM mechanism.

To avoid such uncertainty, any node that is originating the OAM probe for Service Programming OAM MUST mark the packet as OAM packet so that the services can differentiate the OAM packet from data traffic.

5.2. OAM with SR-aware services

As defined in section 4.1 of [I-D.ietf-spring-sr-service-programming], an SR-aware service can process the SR information in the packet header such as performing lookup or executing the next segment etc. An SR-aware service may need to identify the packet payload and/or interpret SR information

to apply the right policy to the received packet. While processing SR information in the packet header, it can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet.

An SR-aware service SHOULD skip applying the service on the OAM packet while forwarding the packet to the next segment or IP address. As defined in section 9, a local policy may be used to control any malicious use of OAM marker.

5.3. OAM with SR-unaware services

As defined in section 4.2 of [I-D.ietf-spring-sr-service-programming], an SR-unaware service may be a legacy service that is not able to process the SR information in the packet header. SR Proxy, an entity that is external to the service is used to handle the SR information processing on behalf of the service. SR Proxy will remove the SR header before forwarding the packet to SR-unaware services to avoid any erroneous decision due to the presence of SR header that the service cannot recognize.

While processing SR information in the packet header, SR proxy can process the OAM packet marker in the SR header to differentiate the OAM packet from normal data packet. SR Proxy MUST skip forwarding the packets with OAM marker to the service while forwarding the packet to the next segment or IP address. As defined in section 9, a local policy may be used to control any malicious use of OAM marker.

5.4. Controlling OAM packet processing in Services

As mentioned in the above sections, SR-aware service or the SR proxy can use the OAM marker to differentiate the OAM packet from data packet to skip the service treatment. Any intentional or unintentional use of OAM marker in data traffic may result in skipping service treatment for data traffic. To avoid such condition, a local policy will be used in the SR-aware service or SR Proxy that SHOULD rate limit or MAY drop the packets received with OAM marker.

6. OAM for Service Programming with SRv6

[I-D.ietf-6man-segment-routing-header] defines SRH.Flags.O-bit in SRH header. When service programming is implemented with SRv6 dataplane, SRH.Flags.O-bit is used as OAM marker. An IPv6 packet received with a local END.OP or END.OTP SID is also considered as an OAM packet.

Any node that is originating OAM probe to a service in SRv6 dataplane MUST set SRH.Flags.O-bit in the SRH.

6.1. OAM Tool Kit

This section describes the availability of different tool kits that can be used to perform OAM functionality for Service Programming with SRv6 dataplane.

6.1.1. OAM Flag Processing

An SR-aware service or SR proxy MUST implement the SRH.Flags.O bit. An SR-aware service SHOULD skip applying the service to the packet when SRH.Flags.O-bit is set and SHOULD forward the packet based on the next header. SR Service Proxy MUST skip applying the service to the packet when SRH.Flags.O-bit is set and SHOULD forward the packet based on the next header.

An SR-aware service and SR proxy may choose to time-stamp and punt the packet with SRH.Flags.O-bit set for further processing and this is a local implementation matter.

6.1.2. OAM Segment ID

Section 3.2 of [[I-D.ali-spring-srv6-oam]] defines OAM segment ID and the associated forwarding semantics to implement the punt behavior for OAM packets. Specifically, the draft defines END.OP and END.OTP SIDs. An IPv6 packet received with DA set to a local END.OP or END.OTP SID is considered as an OAM packet.

Any service policy head end MAY include OAM segment ID in the desired segment list position of SRH. The inclusion of OAM SID in SRH can be used to control the services that are required to punt the OAM packet for processing.

6.1.3. ICMP

There is no hardware or software changes required to use ICMP for Ping operation. It can be triggered from the service policy head end or from any classical IPv6 nodes by sending ICMPv6 Echo Request. The existing ICMP Ping mechanism works seamlessly in SRv6 dataplane with no protocol changes required to the standard ICMPv6 [[RFC4443]], [[RFC4884]] or the standard ICMPv4 [[RFC0792]].

An SR-aware service SHOULD skip the service and forward to next segment based on the SR information in the packet header. An SR Service Proxy MUST skip the service and forward to next segment based on the SR information in the packet header.

6.1.3.1. Pinging Service SID Function

When a remote node pings a service segment, it MUST set SRH.Flags.O = 1. If the target service segment is implemented with USP behavior, the ICMP packet can be constructed without adding END.OP or END.OTP SIDs defined in [I-D.ali-spring-srv6-oam]. However, if the target service SID observes a PSP behavior, the sender needs to insert END.OP/ END.OTP SIDs before the target service SID in the segment-list. In either case, the target SR-aware service or SR proxy receives the ICMP echo request with either SRH.Flags.O-bit set or with the local END.OP or END.OTP SID. In both cases, the packet is punted for slow-path processing and service is skipped.

The Egress node process the packet as per procedure defined in [I-D.ali-spring-srv6-oam]. The Egress checks if the target SID is locally programmed or not.

If the target SID is not locally programmed, the Egress responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned [I-D.ali-spring-srv6-oam].

6.1.4. UDP Probes

A classic traceroute mechanism relies on UDP probes by sending packets with sequentially incrementing the TTL. More details are available in section 4.3.1 of [I-D.ali-spring-srv6-oam].

An SR-aware service or SR proxy upon receiving the probe with TTL=1, may follow the traditional behavior of replying with ICMPv6 Time Exceeded Message (Type 3) as defined in [RFC4443] without applying the service.

Use of SRH.Flags.O bit and END.OP/ END.OTP SIDs as OAM marker in the UDP probe for trace route is same as discussed for ICMPv6 ping discussed in the last section.

6.1.5. In-band OAM

To be Updated.

7. OAM for Service Programming with SR-MPLS

To be updated.

8. IANA Considerations

None.

9. Security Considerations

A local policy may be used to control any malicious use of OAM marker. More details are to be added in a future revision of the document.

10. Acknowledgement

Authors would like to thank Bruno Decraene for his review and useful comments.

11. Normative References

[I-D.ali-spring-srv6-oam]
Ali, Z., Filsfils, C., Kumar, N., Pignataro, C., faiqbal@cisco.com, f., Gandhi, R., Leddy, J., Matsushima, S., Raszuk, R., daniel.voyer@bell.ca, d., Dawra, G., Peirens, B., Chen, M., and G. Naik, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ali-spring-srv6-oam-02 (work in progress), October 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-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-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-00 (work in progress), October 2019.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[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>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

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

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

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cfilsfils@cisco.com

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

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Francois Clad
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Faisal Iqbal
Cisco Systems, Inc.
2000 Innovation Dr
Ottawa, ON 3E8
Canada

Email: faiqbal@cisco.com

Xiaohu Xu
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2019

Z. Ali
C. Filsfils
N. Kumar
C. Pignataro
F. Iqbal
R. Gandhi
Cisco Systems, Inc.
J. Leddy
Comcast
S. Matsushima
SoftBank
R. Raszuk
Bloomberg LP
D. Voyer
Bell Canada
G. Dawra
LinkedIn
B. Peirens
Proximus
M. Chen
Huawei
G. Naik
Drexel University
October 22, 2018

Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)
draft-ali-spring-srv6-oam-02.txt

Abstract

This document defines building blocks that can be used for Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Dataplane (SRv6). The document also describes some SRv6 OAM mechanisms that can be realized using these building blocks.

Status of This Memo

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

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

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

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the

document authors. All rights reserved.

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

Table of Contents

1. Introduction

This document defines building blocks that can be used for Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Dataplane (SRv6). The document also describes some SRv6 OAM mechanisms that can be implemented using these building blocks.

Additional OAM mechanisms will be added in a future revision of the document.

2. Conventions Used in This Document

2.1. Abbreviations

ECMP: Equal Cost Multi-Path.

SID: Segment ID.

SL: Segment Left.

SR: Segment Routing.

SRH: Segment Routing Header.

SRv6: Segment Routing with IPv6 Data plane.

TC: Traffic Class.

UCMP: Unequal Cost Multi-Path.

2.2. Terminology and Reference Topology

This document uses the terminology defined in [I-D.draft-filsfils-spring-srv6-network-programming]. The readers are expected to be familiar with the same.

Throughout the document, the following simple topology is used for illustration.

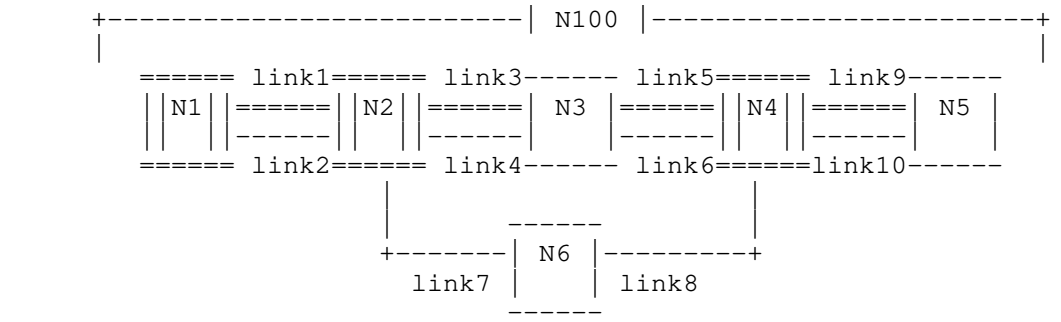


Figure 1 Reference Topology

In the reference topology:

Nodes N1, N2, and N4 are SRv6 capable nodes.

Nodes N3, N5 and N6 are classic IPv6 nodes.

Node N100 is a controller.

Node k has a classic IPv6 loopback address A:k::/128.
A SID at node k with locator block B and function F is represented by B:k:F::

The IPv6 address of the nth Link between node X and Y at the X side is represented as 2001:DB8:X:Y:Xn::, e.g., the IPv6 address of link6 (the 2nd link) between N3 and N4 at N3 in Figure 1 is 2001:DB8:3:4:32::. Similarly, the IPv6 address of link5 (the 1st link between N3 and N4) at node 3 is 2001:DB8:3:4:31::.

B:k:1:: is explicitly allocated as the END function at Node k.

B:k::Cij is explicitly allocated as the END.X function at node k towards neighbor node i via jth Link between node i and node j. e.g., B:2:C31 represents END.X at N2 towards N3 via link3 (the 1st link between N2 and N3). Similarly, B:4:C52 represents the END.X at N4 towards N5 via link10.

<S1, S2, S3> represents a SID list where S1 is the first SID and S3 is the last SID. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID (S1) in the SRH is the first SID and the leftmost SID (S3) in the SRH is the last SID.

(SA, DA) (S3, S2, S1; SL) represents an IPv6 packet, SA is the IPv6 Source Address, DA the IPv6 Destination Address, (S3, S2, S1; SL) is the SRH header that includes the SID list <S1, S2, S3>.

3. OAM Building Blocks

This section defines the various building blocks that can be used to implement OAM mechanisms in SRv6 networks. The following section describes some SRv6 OAM mechanisms that can be implemented using these building blocks.

3.1. O-flag in Segment Routing Header

[I-D. draft-ietf-6man-segment-routing-header] describes the Segment Routing Header (SRH) and how SR capable nodes use it. The draft [I-D. draft-ietf-6man-segment-routing-header] also define an OAM flag (SRH.Flags.O), which indicates that this packet is an operations and management (OAM) packet. The SRH draft also defines the processing rules for the O-flag in the SRH.Flags. The O-flag is one of the OAM building blocks considered in this document.

3.2. OAM Segments

OAM Segment IDs (SIDs) is another components of the building blocks needed to implement SRv6 OAM mechanisms. This document defines a couple of OAM SIDs. Additional SIDs will be added in the later version of the document.

3.2.1. End.OP: OAM Endpoint with Punt

Many scenarios require punting of SRv6 OAM packets at the desired nodes in the network. The "OAM Endpoint with Punt" function (End.OP for short) represents a particular OAM function to implement the punt behavior for an OAM packet. It is described using the pseudocode as follows:

When N receives a packet destined to S and S is a local End.OP SID, N does:

1. Punt the packet to CPU for SW processing (slow-path) ;; Ref1

Ref1: Hardware (microcode) only punts the packet. There is no requirement for the hardware to manipulate any TLV in the SRH (or elsewhere). Software (slow path) implements the required OAM mechanisms.

Please note that in an SRH containing END.OP SID, it is RECOMMENDED to set the SRH.Flags.O-flag = 0.

3.2.2. End.OTP: OAM Endpoint with Timestamp and Punt

Scenarios demanding performance management of an SR policy/ path requires hardware timestamping before hardware punts the packet to the software for OAM processing. The "OAM Endpoint with Timestamp and Punt" function (End.OTP for short) represents an OAM SID function to implement the timestamp and punt behavior for an OAM packet. It is described using the pseudocode as follows:

When N receives a packet destined to S and S is a local End.OTP SID, N does:

1. Timestamp the packet ;; Ref1

2. Punt the packet to CPU for SW processing (slow-path) ;; Ref2

Ref1: Timestamping is done in hardware, as soon as possible during the packet processing.

Ref2: Hardware (microcode) only punts the packet. There is no requirement for the hardware to manipulate any TLV in the SRH (or elsewhere). Software (slow path) implements the required OAM mechanisms.

Please note that in an SRH containing END.OTP SID, it is RECOMMENDED to set the SRH.Flags.O-flag = 0.

4. OAM Mechanisms

This section describes how OAM mechanisms can be implemented using the OAM building blocks described in the previous section. Additional OAM mechanisms will be added in a future revision of the document.

[RFC4443] describes Internet Control Message Protocol for IPv6 (ICMPv6) that is used by IPv6 devices for network diagnostic and error reporting purposes. As Segment Routing with IPv6 data plane (SRv6) simply adds a new type of Routing Extension Header, existing ICMPv6 ping mechanisms can be used in an SRv6 network. This section describes the applicability of ICMPv6 in the SRv6 network and how the existing ICMPv6 mechanisms can be used for providing OAM functionality.

Throughout this document, unless otherwise specified, the acronym ICMPv6 refers to multi-part ICMPv6 messages [RFC4884]. The document does not propose any changes to the standard ICMPv6 [RFC4443], [RFC4884] or standard ICMPv4 [RFC792].

4.1. Ping

There is no hardware or software change required for ping operation at the classic IPv6 nodes in an SRv6 network. That includes the classic IPv6 node with ingress, egress or transit roles. Furthermore, no protocol changes are required to the standard ICMPv6 [RFC4443], [RFC4884] or standard ICMPv4 [RFC792]. In other words, existing ICMP ping mechanisms work seamlessly in the SRv6 networks.

The following subsections outline some use cases of the ICMP ping in the SRv6 networks.

4.1.1. Classic Ping

The existing mechanism to ping a remote IP prefix, along the shortest path, continues to work without any modification. The initiator may be an SRv6 node or a classic IPv6 node. Similarly, the egress or transit may be an SRv6 capable node or a classic IPv6 node.

If an SRv6 capable ingress node wants to ping an IPv6 prefix via an arbitrary segment list <S1, S2, S3>, it needs to initiate ICMPv6 ping with an SR header containing the SID list <S1, S2, S3>. This is illustrated using the topology in Figure 1. Assume all the links have IGP metric 10 except both links between node2 and node3, which have IGP metric set to 100. User issues a ping from node N1 to a loopback of node 5, via segment list <B:2:C31, B:4:C52>.

Figure 2 contains sample output for a ping request initiated at node N1 to the loopback address of node N5 via a segment list <B:2:C31, B:4:C52>.

```
> ping A:5:: via segment-list B:2:C31, B:4:C52

Sending 5, 100-byte ICMP Echos to B5::, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0.625
/0.749/0.931 ms
```

Figure 2 A sample ping output at an SRv6 capable node

All transit nodes process the echo request message like any other data packet carrying SR header and hence do not require any change. Similarly, the egress node (IPv6 classic or SRv6 capable) does not require any change to process the ICMPv6 echo request. For example, in the ping example of Figure 2:

- Node N1 initiates an ICMPv6 ping packet with SRH as follows (A:1::, B:2:C31)(A:5::, B:4:C52, B:2:C31, SL=2, NH = ICMPv6)(ICMPv6 Echo Request).
- Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (B:2:C31) on the echo request packet.
- Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA B:4:C52 in the IPv6 header.
- Node N4, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it observes the END.X function (B:4:C52) with PSP (Penultimate Segment POP) on the echo request packet and removes the SRH and forwards the packet across link10 to N5.
- The echo request packet at N5 arrives as an IPv6 packet without a SRH. Node N5, which is a classic IPv6 node, performs the standard IPv6/ ICMPv6 processing on the echo request and responds, accordingly.

4.1.2. Pinging a SID Function

The classic ping described in the previous section cannot be used to ping a remote SID function, as explained using an example in the following.

Consider the case where the user wants to ping the remote SID function B:4:C52, via B:2:C31, from node N1. Node N1 constructs the ping packet (A:1::, B:2:C31)(B:4:C52, B:2:C31, SL=1; NH=ICMPv6)(ICMPv6 Echo Request). The ping fails because the node N4 receives the ICMPv6 echo request with DA set to B:4:C52 but the next header is ICMPv6, instead of SRH. To solve this problem, the initiator needs to mark the ICMPv6 echo request as an OAM packet.

The OAM packets are identified either by setting the O-flag in SRH or by inserting the END.OP/ END.OTP SIDs at an appropriate place in the SRH. The following illustration uses END.OTP SID but the procedures are equally applicable to the END.OP SID.

In an SRv6 network, the user can exercise two flavors of the ping: end-to-end ping or segment-by-segment ping, as outlined in the following.

4.1.2.1. End-to-end ping using END.OP/ END.OTP

The end-to-end ping illustration uses the END.OTP SID but the procedures are equally applicable to the END.OP SID.

Consider the same example where the user wants to ping a remote SID function B:4:C52 , via B:2:C31, from node N1. To force a punt of the ICMPv6 echo request at the node N4, node N1 inserts the END.OTP SID just before the target SID B:4:C52 in the SRH. The ICMPv6 echo request is processed at the individual nodes along the path as follows:

- Node N1 initiates an ICMPv6 ping packet with SRH as follows (A:1::, B:2:C31)(B:4:C52, B:4:OTP, B:2:C31; SL=2; NH=ICMPv6)(ICMPv6 Echo Request).
- Node N2, which is an SRv6 capable node, performs the standard SRH processing. Specifically, it executes the END.X function (B:2:C31) on the echo request packet.
- Node N3 receives the packet as follows (A:1::, B:4:OTP)(B:4:C52, B:4:OTP, B:2:C31 ; SL=1; NH=ICMPv6)(ICMPv6 Echo Request). Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA B:4:OTP in the IPv6 header.
- When node N4 receives the packet (A:1::, B:4:OTP)(B:4:C52, B:4:OTP, B:2:C31 ; SL=1; NH=ICMPv6)(ICMPv6 Echo Request), it processes the END.OTP SID, as described in the pseudocode in Section 3. The packet gets punted to the ICMPv6 process for processing. The ICMPv6 process checks if the next SID in SRH (the target SID B:4:C52) is locally programmed.
- If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned.

4.1.2.2. Segment-by-segment ping using O-flag (Proof of Transit)

Consider the same example where the user wants to ping a remote SID function B:4:C52, via B:2:C31, from node N1. However, in this ping, the node N1 wants to get a response from each segment node in the SRH as a "proof of transit". In other words, in the segment-by-segment ping case, the node N1 expects a response from node N2 and node N4 for their respective local SID function. When a response to O-bit is desired from the last SID in a SID-list, it is the responsibility of the ingress node to use USP as the last SID. E.g., in this example, the target SID B:4:C52 is a USP SID.

To force a punt of the ICMPv6 echo request at node N2 and node N4, node N1 sets the O-flag in SRH. The ICMPv6 echo request is processed at the individual nodes along the path as follows: and

- Node N1 initiates an ICMPv6 ping packet with SRH as follows (A:1::, B:2:C31)(B:4:C52, B:2:C31; SL=1, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request).
- When node N2 receives the packet (A:1::, B:2:C31)(B:4:C52, B:2:C31; SL=1, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request) packet, it processes the O-flag in SRH, as described in the pseudocode in Section 3. A time-stamped copy of the packet gets punted to the ICMPv6 process for processing. Node N2 continues to apply the B:2:C31 SID function on the original packet and forwards it, accordingly. As B:4:C52 is a USP SID, N2 does not remove the SRH.
  The ICMPv6 process at node N2 checks if its local SID (B:2:C31) is locally programmed or not and responds to the ICMPv6 Echo Request.
- If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned. Please note that, as mentioned in Section 3, if node N2 does not support the O-flag, it simply ignores it and process the local SID, B:2:C31.
- Node N3, which is a classic IPv6 node, performs the standard IPv6 processing. Specifically, it forwards the echo request based on DA B:4:C52 in the IPv6 header.
- When node N4 receives the packet (A:1::, B:4:C52)(B:4:C52, B:2:C31; SL=0, Flags.O=1; NH=ICMPv6)(ICMPv6 Echo Request), it processes the O-flag in SRH, as described in the pseudocode in Section 3. A time-stamped copy of the packet gets punted to the ICMPv6 process for processing. The ICMPv6 process at node N4 checks if its local SID (B:2:C31) is locally programmed or not and responds to the ICMPv6 Echo Request. If the target SID is not locally programmed, N4 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "SID not locally implemented (TBA)"); otherwise a success is returned.

Support for O-flag is part of node capability advertisement. That enables node N1 to know which segment nodes are capable of responding to the ICMPv6 echo request. Node N1 processes the echo responses and presents data to the user, accordingly.

Please note that segment-by-segment ping can be used to address proof of transit use-case.

4.2. Error Reporting

Any IPv6 node can use ICMPv6 control messages to report packet processing errors to the host that originated the datagram packet. To name a few such scenarios:

- If the router receives an undeliverable IP datagram, or
- If the router receives a packet with a Hop Limit of zero, or
- If the router receives a packet such that if the router decrements the packet's Hop Limit it becomes zero, or
- If the router receives a packet with problem with a field in the IPv6 header or the extension headers such that it cannot complete processing the packet, or
- If the router cannot forward a packet because the packet is larger than the MTU of the outgoing link.

In the scenarios listed above, the ICMPv6 response also contains the IP header, IP extension headers and leading payload octets of the "original datagram" to which the ICMPv6 message is a response. Specifically, the "Destination Unreachable Message", "Time Exceeded Message", "Packet Too Big Message" and "Parameter Problem Message" ICMPV6 messages can contain as much of the invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU [RFC4443], [RFC4884]. In an SRv6 network, the copy of the invoking packet contains the SR header. The packet originator can use this information for diagnostic purposes. For example, traceroute can use this information as detailed in the following.

4.3. Traceroute

There is no hardware or software change required for traceroute operation at the classic IPv6 nodes in an SRv6 network. That includes the classic IPv6 node with ingress, egress or transit roles. Furthermore, no protocol changes are required to the standard traceroute operations. In other words, existing traceroute mechanisms work seamlessly in the SRv6 networks.

The following subsections outline some use cases of the traceroute in the SRv6 networks.

4.3.1. Classic Traceroute

The existing mechanism to traceroute a remote IP prefix, along the shortest path, continues to work without any modification. The initiator may be an SRv6 node or a classic IPv6 node. Similarly, the egress or transit may be an SRv6 node or a classic IPv6 node.

If an SRv6 capable ingress node wants to traceroute to IPv6 prefix via an arbitrary segment list <S1, S2, S3>, it needs to initiate traceroute probe with an SR header containing the SID list <S1, S2, S3>. That is illustrated using the topology in Figure 1. Assume all the links have IGP metric 10 except both links between node2 and node3, which have IGP metric set to 100. User issues a traceroute from node N1 to a loopback of node 5, via segment list <B:2:C31, B:4:C52>. Figure 3 contains sample output for the traceroute request.

```
> traceroute A:5:: via segment-list B:2:C31, B:4:C52

Tracing the route to B5::

 1  2001:DB8:1:2:21:: 0.512 msec 0.425 msec 0.374 msec
    SRH: (A:5::, B:4:C52, B:2:C31, SL=2)

 2  2001:DB8:2:3:31:: 0.721 msec 0.810 msec 0.795 msec
    SRH: (A:5::, B:4:C52, B:2:C31, SL=1)

 3  2001:DB8:3:4::41:: 0.921 msec 0.816 msec 0.759 msec
    SRH: (A:5::, B:4:C52, B:2:C31, SL=1)

 4  2001:DB8:4:5::52:: 0.879 msec 0.916 msec 1.024 msec
```

Figure 3 A sample traceroute output at an SRv6 capable node

Please note that information for hop2 is returned by N3, which is a classic IPv6 node. Nonetheless, the ingress node is able to display SR header contents as the packet travels through the IPv6 classic node. This is because the "Time Exceeded Message" ICMPv6 message can contain as much of the invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU [RFC4443]. The SR header is also included in these ICMPv6 messages initiated by the classic IPv6 transit nodes that are not running SRv6 software. Specifically, a node generating ICMPv6 message containing a copy of the invoking packet does not need to understand the extension header(s) in the invoking packet.

The segment list information returned for hop1 is returned by N2, which is an SRv6 capable node. Just like for hop2, the ingress node is able to display SR header contents for hop1.

There is no difference in processing of the traceroute probe at an IPv6 classic node and an SRv6 capable node. Similarly, both IPv6 classic and SRv6 capable nodes use the address of the interface on which probe was received as the source address in the ICMPv6

response. ICMP extensions defined in [RFC5837] can be used to also display information about the IP interface through which the datagram would have been forwarded had it been forwardable, and the IP next hop to which the datagram would have been forwarded, the IP interface upon which a datagram arrived, the sub-IP component of an IP interface upon which a datagram arrived.

The information about the IP address of the incoming interface on which the traceroute probe was received by the reporting node is very useful. This information can also be used to verify if SID functions B:2:C31 and B:4:C52 are executed correctly by N2 and N4, respectively. Specifically, the information displayed for hop2 contains the incoming interface address 2001:DB8:2:3:31:: at N3. This matches with the expected interface bound to END.X function B:2:C31 (link3). Similarly, the information displayed for hop5 contains the incoming interface address 2001:DB8:4:5::52:: at N5. This matches with the expected interface bound to the END.X function B:4:C52 (link10).

4.3.2. Traceroute to a SID Function

The classic traceroute described in the previous section cannot be used to traceroute a remote SID function, as explained using an example in the following.

Consider the case where the user wants to traceroute the remote SID function B:4:C52, via B:2:C31, from node N1. The trace route fails at N4. This is because the node N4 trace route probe where next header is UDP or ICMPv6, instead of SRH (even though the hop limit is set to 1). To solve this problem, the
initiator needs to mark the ICMPv6 echo request as an OAM packet.

The OAM packets are identified either by setting the O-flag in SRH or by inserting the END.OTP SID at an appropriate place in the SRH.

In an SRv6 network, the user can exercise two flavors of the traceroute: hop-by-hop traceroute or overlay traceroute.

- In hop-by-hop traceroute, user gets responses from all nodes including classic IPv6 transit nodes, SRv6 capable transit nodes as well as SRv6 capable segment endpoints. E.g., consider the example where the user wants to traceroute to a remote SID function B:4:C52 , via B:2:C31, from node N1. The traceroute

output will also display information about node3, which is a transit (underlay) node.
- The overlay traceroute, on the other hand, does not trace the underlay nodes. In other words, the overlay traceroute only displays the nodes that acts as SRv6 segments along the route. I.e., in the example where the user wants to traceroute to a remote SID function B:4:C52 , via B:2:C31, from node N1, the overlay traceroute would only display the traceroute information from node N2 and node N2 and will not display information from node 3.

4.3.2.1. Hop-by-hop traceroute using END.OP/ END.OTP

In this section, hop-by-hop traceroute to a SID function is exemplified using UDP probes. However, the procedure is equally applicable to other implementation of traceroute mechanism. Furthermore, the illustration uses the END.OTP SID but the procedures are equally applicable to the END.OP SID

Consider the same example where the user wants to traceroute to a remote SID function B:4:C52 , via B:2:C31, from node N1. To force a punt of the traceroute probe only at the node N4, node N1 inserts the END.OTP SID just before the target SID B:4:C52 in the SRH. The traceroute probe is processed at the individual nodes along the path as follows.

- Node N1 initiates a traceroute probe packet with a monotonically increasing value of hop count and SRH as follows (A:1::, B:2:C31)(B:4:C52, B:4:OTP, B:2:C31; SL=2; NH=UDP)(Traceroute probe).
- When node N2 receives the packet with hop-count = 1, it processes the hop count expiry. Specifically, the node N2 responses with the ICMPv6 message (Type: "Time Exceeded", Code: "Time to Live exceeded in Transit").
- When Node N2 receives the packet with hop-count > 1, it performs the standard SRH processing. Specifically, it executes the END.X function (B:2:C31) on the traceroute probe.
- When node N3, which is a classic IPv6 node, receives the packet (A:1::, B:4:OTP)(B:4:C52, B:4:OTP, B:2:C31 ; HC=1, SL=1; NH=UDP)(Traceroute probe) with hop-count = 1, it processes the hop count expiry. Specifically, the node N3 responses with the ICMPv6 message (Type: "Time Exceeded", Code: "Time to Live exceeded in Transit").
- When node N3, which is a classic IPv6 node, receives the packet with hop-count > 1, it performs the standard IPv6 processing. Specifically, it forwards the traceroute probe based on DA B:4:OTP in the IPv6 header.

- When node N4 receives the packet (A:1::, B:4:OTP)(B:4:C52, B:4:OTP, B:2:C31 ; SL=1; HC=1, NH=UDP)(Traceroute probe), it processes the END.OTP SID, as described in the pseudocode in Section 3. The packet gets punted to the traceroute process for processing. The traceroute process checks if the next SID in SRH (the target SID B:4:C52) is locally programmed. If the target SID B:4:C52 is locally programmed, node N4 responses with the ICMPv6 message (Type: Destination unreachable, Code: Port Unreachable). If the target SID B:4:C52 is not a local SID, node N4 silently drops the traceroute probe.

Figure 4 displays a sample traceroute output for this example.

```
> traceroute srv6 B:4:C52 via segment-list B:2:C31

Tracing the route to SID function B:4:C52

 1  2001:DB8:1:2:21 0.512 msec 0.425 msec 0.374 msec
    SRH: (B:4:C52, B:4:OTP, B:2:C31; SL=2)

 2  2001:DB8:2:3:31 0.721 msec 0.810 msec 0.795 msec
    SRH: (B:4:C52, B:4:OTP, B:2:C31; SL=1)

 3  2001:DB8:3:4::41 0.921 msec 0.816 msec 0.759 msec
    SRH: (B:4:C52, B:4:OTP, B:2:C31; SL=1)
```

Figure 4 A sample output for hop-by-hop traceroute to a SID function

4.3.2.2. Tracing SRv6 Overlay

The overlay traceroute does not trace the underlay nodes, i.e., only displays the nodes that acts as SRv6 segments along the path. This is achieved by setting the SRH.Flags.O bit.

In this section, overlay traceroute to a SID function is exemplified using UDP probes. However, the procedure is equally applicable to other implementation of traceroute mechanism.

Consider the same example where the user wants to traceroute to a remote SID function B:4:C52 , via B:2:C31, from node N1.

- Node N1 initiates a traceroute probe with SRH as follows (A:1::, B:2:C31)(B:4:C52, B:2:C31; HC=64, SL=1, Flags.O=1; NH=UDP)(Traceroute Probe). Please note that the hop-count is

set to 64 to skip the underlay nodes from tracing. The O-flag in SRH is set to make the overlay nodes (nodes processing the SRH) respond.

- When node N2 receives the packet (A:1::, B:2:C31)(B:4:C52, B:2:C31; SL=1, HC=64, Flags.O=1; NH=UDP)(Traceroute Probe), it processes the O-flag in SRH, as described in the pseudocode in Section 3. A time-stamped copy of the packet gets punted to the traceroute process for processing. Node N2 continues to apply the B:2:C31 SID function on the original packet and forwards it, accordingly. As SRH.Flags.O=1, Node N2 also disables the PSP flavor, i.e., does not remove the SRH. The traceroute process at node N2 checks if its local SID (B:2:C31) is locally programmed. If the SID is not locally programmed, it silently drops the packet. Otherwise, it performs the egress check by looking at the SL value in SRH.
- As SL is not equal to zero (i.e., it's not egress node), node N2 responses with the ICMPv6 message (Type: "SRv6 OAM (TBA)", Code: "O-flag punt at Transit (TBA)"). Please note that, as mentioned in Section 3, if node N2 does not support the O-flag, it simply ignores it and processes the local SID, B:2:C31.
- When node N3 receives the packet (A:1::, B:4:C52)(B:4:C52, B:2:C31; SL=0, HC=63, Flags.O=1; NH=UDP)(Traceroute Probe), performs the standard IPv6 processing. Specifically, it forwards the traceroute probe based on DA B:4:C52 in the IPv6 header. Please note that there is no hop-count expiration at the transit nodes.
- When node N4 receives the packet (A:1::, B:4:C52)(B:4:C52, B:2:C31; SL=0, HC=62, Flags.O=1; NH=UDP)(Traceroute Probe), it processes the O-flag in SRH, as described in the pseudocode in Section 3. A time-stamped copy of the packet gets punted to the traceroute process for processing. The traceroute process at node N4 checks if its local SID (B:2:C31) is locally programmed. If the SID is not locally programmed, it silently drops the packet. Otherwise, it performs the egress check by looking at the SL value in SRH. As SL is equal to zero (i.e., N4 is the egress node), node N4 tries to consume the UDP probe. As UDP probe is set to access an invalid port, the node N4 responses with the ICMPv6 message (Type: Destination unreachable, Code: Port Unreachable).

Figure 5 displays a sample overlay traceroute output for this example. Please note that the underlay node N3 does not appear in the output.

```
> traceroute srv6 B:4:C52 via segment-list B:2:C31

Tracing the route to SID function B:4:C52
```

```
1  2001:DB8:1:2:21:: 0.512 msec 0.425 msec 0.374 msec
   SRH: (B:4:C52, B:4:OTP, B:2:C31; SL=2)

2  2001:DB8:3:4::41:: 0.921 msec 0.816 msec 0.759 msec
   SRH: (B:4:C52, B:4:OTP, B:2:C31; SL=1)
```

Figure 5 A sample output for overlay traceroute to a SID function

4.5. Monitoring of SRv6 Paths

In the recent past, network operators are interested in performing network OAM functions in a centralized manner. Various data models like YANG are available to collect data from the network and manage it from a centralized entity.

SR technology enables a centralized OAM entity to perform path monitoring from centralized OAM entity without control plane intervention on monitored nodes. [I.D-draft-ietf-spring-oam-usecase] describes such a centralized OAM mechanism. Specifically, the draft describes a procedure that can be used to perform path continuity check between any nodes within an SR domain from a centralized monitoring system, with minimal or no control plane intervene on the nodes. However, the draft focuses on SR networks with MPLS data plane. The same concept applies to the SRv6 networks. This document describes how the concept can be used to perform path monitoring in an SRv6 network. This document describes how the concept can be used to perform path monitoring in an SRv6 network as follows.

In the above reference topology, N100 is the centralized monitoring system implementing an END function B:100:1::. In order to verify a segment list <B:2:C31, B:4:C52>, N100 generates a probe packet with SRH set to (B:100:1::, B:4:C52, B:2:C31, SL=2). The controller routes the probe packet towards the first segment, which is B:2:C31. N2 performs the standard SRH processing and forward it over link3 with the DA of IPv6 packet set to B:4:C52. N4 also performs the normal SRH processing and forward it over link10 with the DA of IPv6 packet set to B:100:1::. This makes the probe loops back to the centralized monitoring system.

In the reference topology in Figure 1, N100 uses an IGP protocol like OSPF or ISIS to get the topology view within the IGP domain. N100 can also use BGP-LS to get the complete view of an inter-domain topology. In other words, the controller leverages the visibility of the topology to monitor the paths between the various endpoints without control plane intervention required at the monitored nodes.

5. Security Considerations

This document does not define any new protocol extensions and relies on existing procedures defined for ICMP. This document does not impose any additional security challenges to be considered beyond security considerations described in [RFC4884], [RFC4443], [RFC792] and RFCs that updates these RFCs.

6. IANA Considerations

6.1. ICMPv6 type Numbers Registry

This document defines one ICMPv6 Message, a type that has been allocated from the "ICMPv6 'type' Numbers" registry of [RFC4443].

Specifically, it requests to add the following to the "ICMPv6 Type Numbers" registry:

TBA (suggested value: 162) SRv6 OAM Message.

The document also requests the creation of a new IANA registry to the

"ICMPv6 'Code' Fields" against the "ICMPv6 Type Numbers TBA - SRv6 OAM Message" with the following codes:

Code	Name	Reference
0	No Error	This document
1	SID is not locally implemented	This document
2	O-flag punt at Transit	This document

6.3. SRv6 OAM Endpoint Types

This I-D requests to IANA to allocate, within the "SRv6 Endpoint Behaviors Registry" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry [I-D.filsfils-spring-srv6-network-programming], the following allocations:

Value (Suggested Value)	Endpoint Behavior	Reference
TBA (30)	End.OP	[This.ID]
TBA (31)	End.OTP	[This.ID]

7. References

7.1. Normative References

[RFC792] J. Postel, "Internet Control Message Protocol", RFC 792, September 1981.

[RFC4443] A. Conta, S. Deering, M. Gupta, Ed., "Internet Control

Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4884] R. Bonica, D. Gan, D. Tappan, C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.

[RFC5837] A. Atlas, Ed., R. Bonica, Ed., C. Pignataro, Ed., N. Shen, JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.

[I-D.filsfils-spring-srv6-network-programming] C. Filsfils, et al., "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming, work in progress.

[I-D.6man-segment-routing-header] Previdi, S., Filsfils, et al, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.

7.2. Informative References

[I-D.bashandy-isis-srv6-extensions] IS-IS Extensions to Support Routing over IPv6 Dataplane. L. Ginsberg, P. Psenak, C. Filsfils, A. Bashandy, B. Decraene, Z. Hu, draft-bashandy-isis-srv6-extensions, work in progress.

[I-D.dawra-idr-bgpls-srv6-ext] G. Dawra, C. Filsfils, K. Talaulikar, et al., BGP Link State extensions for IPv6 Segment Routing (SRv6), draft-dawra-idr-bgpls-srv6-ext, work in progress.

[I-D.ietf-spring-oam-usecase] A Scalable and Topology-Aware MPLS Dataplane Monitoring System. R. Geib, C. Filsfils, C. Pignataro, N. Kumar, draft-ietf-spring-oam-usecase, work in progress.

[I-D.brockners-inband-oam-data] F. Brockners, et al., "Data Formats for In-situ OAM", draft-brockners-inband-oam-data, work in progress.

[I-D.brockners-inband-oam-transport] F.Brockners, at al., "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport, work in progress.

[I-D.brockners-inband-oam-requirements] F.Brockners, et al., "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements, work in progress.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy for Traffic Engineering", draft-filsfils-spring-segment-routing-policy, work in progress.

8. Acknowledgments

To be added.

Authors' Addresses

Clarence Filsfils
Cisco Systems, Inc.
Email: cfilsfil@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Nagendra Kumar
Cisco Systems, Inc.
Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
Email: cpignata@cisco.com

Faisal Iqbal
Cisco Systems, Inc.
Email: faiqbal@cisco.com

Rakesh Gandhi
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

John Leddy
Comcast
Email: John_Leddy@cable.comcast.com

Robert Raszuk
Bloomberg LP
731 Lexington Ave
New York City, NY10022, USA
Email: robert@raszuk.net

Satoru Matsushima
SoftBank
Japan
Email: satoru.matsushima@g.softbank.co.jp

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

Gaurav Dawra
LinkedIn
Email: gdawra.ietf@gmail.com

Bart Peirens
Proximus
Email: bart.peirens@proximus.com

Mach Chen
Huawei
Email: mach.chen@huawei.com

Gaurav Naik
Drexel University
United States of America
Email: gn@drexel.edu

Routing area
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

K. Arora
S. Hegde
Juniper Networks Inc.

October 16, 2018

TTL Procedures for SR-TE Paths in Label Switched Path Traceroute Mechanisms
draft-arora-mpls-spring-ttl-procedures-srte-paths-00

Abstract

Segment routing supports the creation of explicit paths using adjacency-sids, node-sids, and anycast-sids. The SR-TE paths are built by stacking the labels that represent the nodes and links in the explicit path. A very useful Operations And Maintenance requirement is to be able to trace these paths as defined in [RFC8029]. This document specifies a uniform mechanism to support MPLS traceroute for the SR-TE paths when the nodes in the network are following uniform mode or short-pipe mode [RFC3443].

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 April 19, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

The mechanisms to handle TTL procedures for SR-TE paths are described in ([RFC8287]). Section 7.5 of ([RFC8287]) defines the TTL manipulation procedures for short pipe model as the LSR initiating the traceroute SHOULD start by setting the TTL to 1 for the tunnel in the LSP's label stack it wants to start the tracing from, the TTL of all outer labels in the stack to the max value, and the TTL of all the inner labels in the stack to zero. However this mechanism has issues when the constituent tunnels are penultimate-hop-popping(PHP). c

Section 2 describes problems tracing SR-TE paths and the need for a specialized mechanism to trace SR-TE paths. Section 3 describes the solution applied to mpls echo request/response to trace adjacency-sids and node-sids trace SR-TE path in uniform model and short pipe model.

2. Problem with SR-TE Paths

The topology shown in Figure 1. illustrates a example network topology with SPRING enabled on each node.

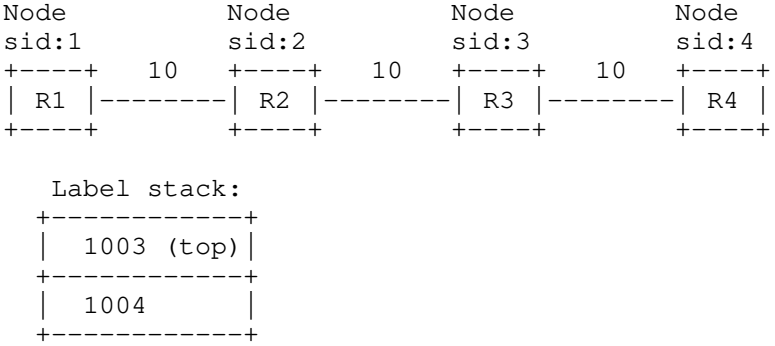


Figure 1: Example topology with SRGB 1000-2000

Consider an explicit path in the topology in Figure 1 from R1->R4 via R1->R2->R3->R4. The label stack to instantiate this path contains two node-sids 1003 and 1004. The 1003 label will take the packet from R1 to R3. The next label in the stack 1004 will take the packet from R3 to the destination R4. consider the mechanism below for the TTL procedures specified in RFC 8287 for short pipe model and uniform model for PHP LSPs.

Notation: ((X,Y>,(Z,W)) refers to a label stack whose top label stack entry has the label corresponding to the node-SID of X, with TTL Y, and whose second label stack entry has the label corresponding to the node-SID of Z, with TTL W.

According to the procedure in Section 7.5 of [RFC8287], the LSP traceroute is done as follows in short pipe model and uniform model:

2.1. Short Pipe model

Refer the diagram in Figure 1.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. R2 forwards packet to R3 as ((1004,0)) (i.e. R2 being PHP pops stack and does not propagate TTL)

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 will cause R3 to drop the packet.

RFC 8287 suggests that when R1's LSP Echo Request has reached the egress of the outer tunnel, R1 should being to trace the inner tunnel by sending a LSP Echo Request with label stack ((1003,2),(1004,1)). However there is no way for R1 to do that in this scenario, because R1 cannot tell when the egress of the outer tunnel has been reached.

2.2. Uniform Model

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. It is expected that R2 should propogate the TTL of outer label to inner label before forwarding the packet to R3. However most of the PFEs implementations generally do not increase a label stack entry's TTL when they do TTL propagation. So when (1003,2) is popped, we might still end up with (1004,0) at R3, even if we have TTL propagation configured. Increasing the TTL of a traveling packet may not be a good practice.

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 will cause R3 to drop the packet.

So in either case (uniform model or short pipe model) traceroute may not work for SR-TE paths with PHP Lsps.

3. Detailed Solution For TTL procedures for SR-TE paths

3.1. P bit in DDMT TLV

DS flags has 4 unused bits from position '0' to '3'. This document uses bit '3' in DS flags of downstream mapping TLV.

3.2. Procedures for a PHP router of the tunnel being traced

When a LSR receives an echo request it MUST validates the outermost FEC in the echo request. LSR must set the 'P' bit in the DS flags of downstream mapping TLV if its a PHP router for the outermost FEC. Other cases it should work as explained in RFC8287 and RFC 8209

3.3. Procedures for a egress router of the tunnel being traced

When a LSR receives an echo request it MUST validates the outermost FEC in the echo request. If LSR is egress for the outermost FEC Then it MUST look for the next label in the FEC stack if exists any. If the LSP is the PHP router for the next FEC (next to outermost FEC in FEC stack if any), Then LSR MUST set 'P' bit in the downstream mapping TLV. Other cases it should work as explained in RFC8287 and RFC 8209

3.4. Procedures for a ingress router of the SR-TE path

When an ingress LSR receives an echo response with 'P' bit set in the DS flags of downstream mapping TLV, Then while sending next echo request Ingress LSR MUST increase the TTL value of inner label also (if exists) in addition to increasing the TTL value of the tunnel its tracing. Other cases it should work as explained in RFC8287 and RFC 8209

3.5. Example describing the solution

This section provides a detailed description of how PHP router helps ingress in handling TTL procedures for SR-TE paths. Below are the procedures performed by PHP router and ingress router to perform TTL procedure for mpls traceroute for SR-TE paths. Below solution works for both uniform model and short pipe model.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet. R2's local software validates the outermost FEC and looking at the FEC R2 knows that its the PHP router for outermost FEC (Node-Sid R3).

3. R2 sets a bit in the DS flags in the DDMT TLV in echo response (P bit, One of the reserved bits).

4. When R1 looks at the echo response from R2 it sees P bit in DDMT TLV .

5. So R1 increment the TTL value of Node-R3 by 1 (make it 2) and TTL value of next element in the label stack also

6. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,1)).

7. R2 being PHP pops the ouetrmost label from the label stack and forward the packet to R3 with with label (1004, 1)

8. R3 receives mpls LSP Echo Request with TTL as 1 for outer most label, R3's local software process the echo request.

9. R3 validates the outermost FEC and knows that R3 is the egress for outermost FEC (Node-Sid R3).

10. Since R3 is the egress for outermost FEC so R3 should look at the next FEC in the FEC stack (Node-Sid-R4) and identify if R3 is the PHP router for next FEC in the label stack. Since R3 is the PHP router for next FEC (Node-Sid R4) R3 should set 'P' bit in the in the DS flags in the DDMT TLV in echo response with return code as 'Egress'.

11. When R1 receives 'P' in the DDMT TLV as well as return code as egress then R1 knows that the ouetrmost tunnel is traced.

11. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,2)) with FEC Node-Sid-R4 (Since its received Egress for ouetrmost FEC Node-Sid-R3).

12. R2 pops the first label from the label stack and R3 pops the second label from the label stack.

14. R4 receives an unlabelled packet with RA bit set in ip options. R4 delivers the packet to local software for processing.

15. R4's local software validates the ouetmost FEC as 'egress' and there is no more FEC in the FEC stack TLV.

16. R4 sends an echo reply with return code as egress.

17. R1 receives an echo reply with return code as egress for the last FEC in the FEC stack TLV and completes the traceroute.

4. Backward Compatibility

If the LSR with the proposed solution is the Ingress and all other LSR in the SR tunnel are not with the extension, Then no LSR is going to set 'P' bit so ingress LSR with new extension will work as per [RFC8029] and [RFC8287].If the LSR with the proposed extension is the one of the transit router and if its the PHP then it may set 'P' bit based on the section 3. Ingress may not react to the 'P' bit and traceroute will continue to work as per [RFC8029] and [RFC8287].

5. Security Considerations

TBD

6. IANA Considerations

IANA has created and now maintains a registry entitled "DS Flags". The registration policy for this registry is Standards Action [RFC5226]. IANA has made the following assignments: Bit Number Name Reference ---------- ---------- --------- 7 N: Treat as a Non-IP Packet [RFC8029] 6 I: Interface and Label Stack Object Request [RFC8029] 5 E: ELI/EL push indicator [RFC8012] 4 L: Label-based load balance indicator [RFC8012] 3 P: Penulimate Hop router 2-0 Unassigned

7. References

7.1. Normative References

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

[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>.

[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>.

7.2. Informative References

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <https://www.rfc-editor.org/info/rfc3443>.

Authors' Addresses

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

Email: kapilaro@juniper.net

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

Email: shraddha@juniper.net

Routing area
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2019

K. Arora
S. Hegde
Juniper Networks Inc.
S. Aldrin
Google
S. Litkowski
Orange Business Service
M. Durrani
Equinix

February 21, 2019

TTL Procedures for SR-TE Paths in Label Switched Path Traceroute Mechanisms
draft-arora-mpls-spring-ttl-procedures-srte-paths-01

Abstract

Segment routing supports the creation of explicit paths using adjacency-sids, node-sids, and anycast-sids. The SR-TE paths are built by stacking the labels that represent the nodes and links in the explicit path. A very useful Operations And Maintenance requirement is to be able to trace these paths as defined in [RFC8029]. This document specifies a uniform mechanism to support MPLS traceroute for the SR-TE paths when the nodes in the network are following uniform mode or short-pipe mode [RFC3443].

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 August 25, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

The mechanisms to handle TTL procedures for SR-TE paths are described in ([RFC8287]). Section 7.5 of ([RFC8287]) defines the TTL manipulation procedures for short pipe model as below.The LSR

initiating the traceroute SHOULD start by setting the TTL to 1 for the tunnel in the LSP's label stack it wants to start the tracing from, the TTL of all outer labels in the stack to the max value, and the TTL of all the inner labels in the stack to zero. However this mechanism has issues when the constituent tunnels are penultimate-hop-popping(PHP). This document does not propose any change to ([RFC8287]) if the constituent tunnels are ultimate-hop-popping (UHP) or Egress LSR advertizes explicit NULL.

Section 2 describes problems in tracing SR-TE paths and the need for a specialized mechanism to trace SR-TE paths. Section 3 describes the solution applied to mpls echo request/response to trace adjacency-sids and node-sids trace SR-TE path in uniform model and short pipe model.

2. Problem with SR-TE Paths

The topology shown in Figure 1. illustrates a example network topology with SPRING enabled on each node.

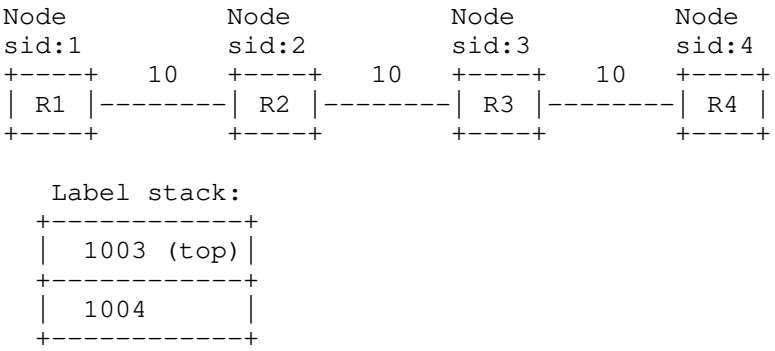


Figure 1: Example topology with SRGB 1000-2000

Consider an explicit path in the topology in Figure 1 from R1->R4 via R1->R2->R3->R4. The label stack to instantiate this path contains two node-sids 1003 and 1004. The 1003 label will take the packet from R1 to R3. The next label in the stack 1004 will take the packet from R3 to the destination R4. consider the mechanism below for the TTL procedures specified in RFC 8287 for short pipe model and uniform model for PHP LSPs.

Notation: ((X,Y),(Z,W)) refers to a label stack whose top label stack entry has the label corresponding to the node-SID of X, with TTL Y, and whose second label stack entry has the label corresponding to the node-SID of Z, with TTL W.

According to the procedure in Section 7.5 of [RFC8287], the LSP traceroute is done as follows in short pipe model and uniform model:

2.1. Short Pipe model

Refer the diagram in Figure 1.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp traceroute packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. R2 forwards packet to R3 as ((1004,0)) (i.e. R2 being PHP, pops the label 1003 and does not propagate TTL)

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 may cause R3 to drop the packet or rate limit it.

6. Even if R3's local software processes the packet and validates the FEC for 1003 and sends egress code in echo-reply, the next packet will have ((1003,255), (1004, 1)) which causes TTL to expire again on R3 as the 1003 label is popped at the penultimate.

RFC 8287 suggests that when R1's LSP Echo Request has reached the egress of the outer tunnel, R1 should begin to trace the inner tunnel by sending a LSP Echo Request with label stack ((1003,255),(1004,1)). However, as explained in step 6, the traceroute procedure does not work correctly.

2.2. Uniform Model

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. It is expected that R2 should propogate the TTL of outer label to inner label before forwarding the packet to R3. However most of the PFEs implementations generally do not increase a label stack entry's TTL when they do TTL propagation. So when (1003,2) is popped, we might still end up with (1004,0) at R3, even if we have TTL propagation configured. Increasing the TTL of a packet is not a good practice as it can result in forwarding loops.

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 will cause R3 to drop the packet or rate limit it.

6. Even if R3's local software processes the packet and validates the FEC for 1003 and sends egress code in echo-reply, the next packet will have ((1003,255), (1004, 1)) which causes TTL to expire again on R3 as the 1003 label is popped at the penultimate.

So in either case (uniform model or short pipe model) traceroute may not work for SR-TE paths with PHP Lsps.

3. Detailed Solution For TTL procedures for SR-TE paths

3.1. P bit in DDMT TLV

DS flags has 4 unused bits from position '0' to '3'. This document uses bit '3' in DS flags of downstream mapping TLV.

3.1.1. Procedures for a PHP router of the tunnel being traced

When a LSR receives an echo request it MUST validate the outermost FEC in the echo request. LSR SHOULD set the 'P' bit in the DS flags of downstream mapping TLV if its a PHP router for the outermost FEC. Other cases it should work as explained in [RFC8029] and [RFC8287].

3.1.2. Procedures for a egress router of the tunnel being traced

When a LSR receives an echo request it MUST validate the outermost FEC in the echo request. Egress cases should work as explained in [RFC8029] and [RFC8287].

3.1.3. Procedures for a ingress router of the SR-TE path

When an ingress LSR receives an echo response it MUST behave as defined below depending on the return code in the echo response.

1. When an ingress LSR receives an echo response with return code as 8 (Label switched at stack-depth), Ingress LSR MUST check if the LSR that sent the echo response is PHP for the outermost FEC in the FEC

stack. If the LSR that sent the echo response is PHP for the outermost FEC then while sending next echo request Ingress LSR MUST increase the TTL value of inner label also (if exists) in addition to increasing the TTL value of the tunnel it is tracing. Ingress LSR can detect that LSR that sent the echo response is a PHP router for the outermost FEC, either by looking at 'P' bit set in the DS flags of downstream mapping TLV or if Ingress LSR has received LABEL '3' in the label stack TLV of downstream detailed mapping TLV. For all other cases ingress should work as explained in [RFC8029] and [RFC8287].

2. When an Ingress LSR receives an echo response with return code as 3 (Replying router is an egress for the FEC at stack-depth) for the outermost FEC and this is not the only FEC in the FEC stack, then ingress LSR SHOULD remove the outermost FEC from the FEC stack and send the next traceroute request with the same TTL value for all the labels in the label stack as the previous echo request. This will ensure the egress of the tunnel is visited twice, once as egress for top label and again as a transit for next tunnel.

3.1.4. Example describing the solution

This section provides a detailed description of how PHP router helps ingress in handling TTL procedures for SR-TE paths. Below are the procedures performed by PHP router and ingress router to perform TTL procedure for mpls traceroute for SR-TE paths. Below solution works for both uniform model and short pipe model.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet. R2's local software validates the outermost FEC and looking at the FEC R2 knows that its the PHP router for outermost FEC (Node-Sid R3).

3. R2 sets a bit in the DS flags in the DDMT TLV in echo response (P bit, One of the reserved bits).

4. When R1 looks at the echo response from R2 it sees P bit in DDMT TLV.

5. So R1 increments the TTL value of Node-R3 by 1 (make it 2) and TTL value of next element in the label stack also

6. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,1)).

7. R2 being PHP pops the outermost label from the label stack and forwards the packet to R3 with with label (1004, 1)

8. R3 receives mpls LSP Echo Request with TTL as 1 for outer most label, R3's local software processes the echo request.

9. R3 validates the outermost FEC and sends echo response to R1 with return code as the egress for outermost FEC (Node-Sid R3).

10. When R1 receives echo response with return code as egress, R1 should remove outermost FEC (Node-Sid R3) from the FEC stack and send the next echo request with the same TTL value as the previous one i.e ((1003,2),(1004,1)).

11. Since R3 is the PHP router for FEC (Node-Sid R4) in the label stack. R3 should set 'P' bit in the in the DS flags in the DDMT TLV in echo response with return code as Transit.

12. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,2)) with FEC Node-Sid-R4 .

13. R2 pops the first label from the label stack and R3 pops the second label from the label stack.

14. R4 receives an unlabelled packet with RA bit set in ip options. R4 delivers the packet to local software for processing.

15. R4's local software validates the ouetmost FEC as 'egress' and sends an echo reply with return code as egress.

17. R1 receives an echo reply with return code as egress for the last FEC in the FEC stack TLV and completes the traceroute.

3.2. Procedures for handling binding-sids

Inorder to provide greater scalability, network opacity, and service independence, SR architecture [RFC8402] defines a Binding SID (BSID). A Binding SID is bound to an SR policy which typically involves a list of SIDs. These Binding SIDs may appear in another SR Policy or may be used to steer service traffic from the service origin. The TTL handling mechanisms for MPLS traceroute procedures involving Binding SIDs is described below.

3.2.1. Uniform Model

When the node advertising the Binding SID is operating in uniform mode [RFC3443], it SHOULD send FEC stack change sub-TLV as in sec 4.5.1 of [RFC8029]. The ingress node SHOULD increment the TTL of

Binding SID label at every step until "egress" return code is sent for all the new FECs included due to FEC stack change and all the Tunnels replaced by the Binding SID are completely traced. It is required that all the label popping nodes involved in these tunnels MUST support uniform model and copy the TTL to bottom label when the label is popped.

3.2.2. Shortpipe Model

When the node advertising the Binding SID is operating in short pipe model [RFC3443], it SHOULD not send FEC stack change sub-TLV. The Binding SID is treated as single hop and the nodes internal to the Tunnel represented by Binding SID SHOULD NOT be traced.

4. Backward Compatibility

The extension proposed in this document is backward compatible with procedures described in [RFC8029] and [RFC8287]. If the LSR with the proposed solution is the Ingress and all other LSR in the SR tunnel are not with the extension, Then no LSR is going to set 'P' bit so ingress LSR with new extension will work as per [RFC8029] and [RFC8287].If the LSR with the proposed extension is the one of the transit router and if its the PHP then it may set 'P' bit based on the section 3. Ingress may not react to the 'P' bit and traceroute will continue to work as per [RFC8029] and [RFC8287].

5. Security Considerations

TBD

6. IANA Considerations

IANA has created and now maintains a registry entitled "DS Flags". The registration policy for this registry is Standards Action [RFC5226]. IANA has made the following assignments:

Bit Number Name Reference

---------- ---------- ---------

7 N: Treat as a Non-IP Packet [RFC8029]

6 I: Interface and Label Stack Object Request [RFC8029]

5 E: ELI/EL push indicator [RFC8012]

4 L: Label-based load balance indicator [RFC8012]

3 P: Penulimate Hop router

2-0 Unassigned

7. Acknowledgements

Thanks to Przemyslaw Krol for careful review and comments.

8. References

8.1. Normative References

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

[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>.

[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>.

8.2. Informative References

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <https://www.rfc-editor.org/info/rfc3443>.

Authors' Addresses

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

Email: kapilaro@juniper.net

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

Email: shraddha@juniper.net

Sam Aldrin
Google

Email: aldrin.ietf@gmail.com

Stephane Litkowski
Orange Business Service

Email: stephane.litkowski@orange.com

Muhammad Durrani
Equinix

Email: mdurrani@equinix.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

W. Cheng
L. Wang
H. Li
China Mobile
M. Chen
Huawei
R. Gandhi
Cisco Systems, Inc.
R. Zigler
Broadcom
S. Zhan
ZTE

October 16, 2018

Path Segment in MPLS Based Segment Routing Network
draft-cheng-spring-mpls-path-segment-03

Abstract

A Segment Routing (SR) path is identified by an SR segment list, one or partial segments of the list cannot uniquely identify the SR path. Path identification is a pre-requisite for various use-cases such as performance measurement (PM) of an SR path.

This document defines a new type of segment that is referred to as Path Segment, which is used to identify an SR path. When used, it is inserted at the ingress node of the SR path and immediately follows the last segment of the SR path. The Path Segment will not be popped off until it reaches the egress node of the SR path.

Path Segment can be used by the egress node to implement path identification hence to support various use-cases including SR path PM, end-to-end 1+1 SR path protection and bidirectional SR paths correlation.

Status of This Memo

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

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

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

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

This Internet-Draft will expire on April 19, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] is a source routed forwarding method that allows to directly encode forwarding instructions (called segments) in each packet, hence it enables to steer traffic through a network without the per-flow states maintained on the transit nodes. Segment Routing can be instantiated on MPLS data plane or IPv6 data plane. The former is called SR-MPLS, the latter is called SRv6

[RFC8402]. SR-MPLS leverages the MPLS label stack to construct SR path, and SRv6 uses the a new IPv6 Extension Header (EH) called the IPv6 Segment Routing Header (SRH) [I-D.ietf-6man-segment-routing-header] to construct SR path.

In an SR-MPLS network, when a packet is transmitted along an SR path, the labels in the MPLS label stack will be swapped or popped. So that no label or only the last label may be left in the MPLS label stack when the packet reaches the egress node. Thus, the egress node cannot determine from which SR path the packet comes.

However, to support use cases like end-to-end 1+1 path protection (Live-Live case), bidirectional path correlation or performance measurement (PM), the ability to implement path identification is a pre-requisite.

Therefore, this document introduces a new segment that is referred to as Path Segment. A Path Segment is defined to uniquely identify an SR path in the context of the egress node. It is normally used by egress nodes for path identification or correlation.

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

1.2. Abbreviations

DM: Delay Measurement.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

PM: Performance Measurement.

PSID: Path Segment ID.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing instantiated on MPLS data plane.

SRv6: Segment Routing instantiated on IPv6 data plane

2. Path Segment

A Path Segment is a single label that is assigned from the Segment Routing Local Block (SRLB) or Segment Routing Global Block (SRGB) of the egress node of an SR path. It means that the Path Segment is unique in the context of the egress node of the SR path. When Path Segment is used, the Path Segment MUST be inserted at the ingress node and MUST immediately follow the last label of the SR path. The Path Segment may be used to identify an SR-MPLS Policy, its Candidate-Path (CP) or a SID List (SL) [I-D.ietf-spring-segment-routing-policy] terminating on an egress node depending on the use-case.

The value of the TTL field of the Path Segment MUST be set to the same value of the last segment label of the SR path. If the Path Segment is the bottom label, the S bit MUST be set.

Normally, the intermediate nodes will not see the Path Segment label and do not know how to process it. A Path Segment presenting to an intermediate node is an error condition.

The egress node MUST pop the Path Segment. The egress node MAY use the Path Segment for further processing. For example, when performance measurement is enabled on the SR path, it can trigger packet counting or timestamping.

The label stack with Path Segment is as below (Figure1):

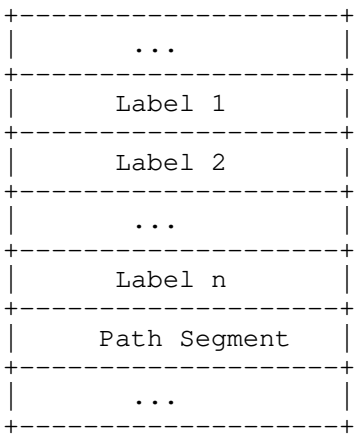


Figure 1: Label Stack with Path Segment

Where:

- The Labels 1 to n are the segment label stack used to direct how to steer the packets along the SR path.

- The Path Segment identifies the SR path in the context of the egress node of the SR path.

3. Nesting of Path Segments

Binding SID (BSID) [RFC8402] can be used for SID list compression. With BSID, an end-to-end SR path can be split into several sub-paths, each sub-path is identified by a BSID. Then an end-to-end SR path can be identified by a list of BSIDs, therefore, it can provide better scalability.

BSID and Path SID (PSID) can be combined to achieve both sub-path and end-to-end path monitoring. A reference model for such a combination in (Figure 2) shows an end-to-end path (A->D) that spans three domains (Access, Aggregation and Core domain) and consists of three sub-paths, one in each sub-domain (sub-path (A->B), sub-path (B->C) and sub-path (C->D)). Each sub-path is allocated a BSID. For nesting the sub-paths, each sub-path is allocated a PSID. Then, the SID list of the end-to-end path can be expressed as <BSID1, BSID2, ..., BSIDn, e-PSID>, where the e-PSID is the PSID of the end-to-end path. The SID list of a sub-path can be expressed as <SID1, SID2, ...SIDn, s-PSID>, where the s-PSID is the PSID of the sub-path.

Figure 2 shows the details of the label stacks when PSID and BSID are used to support both sub-path and end-to-end path monitoring in a multi-domain scenario.

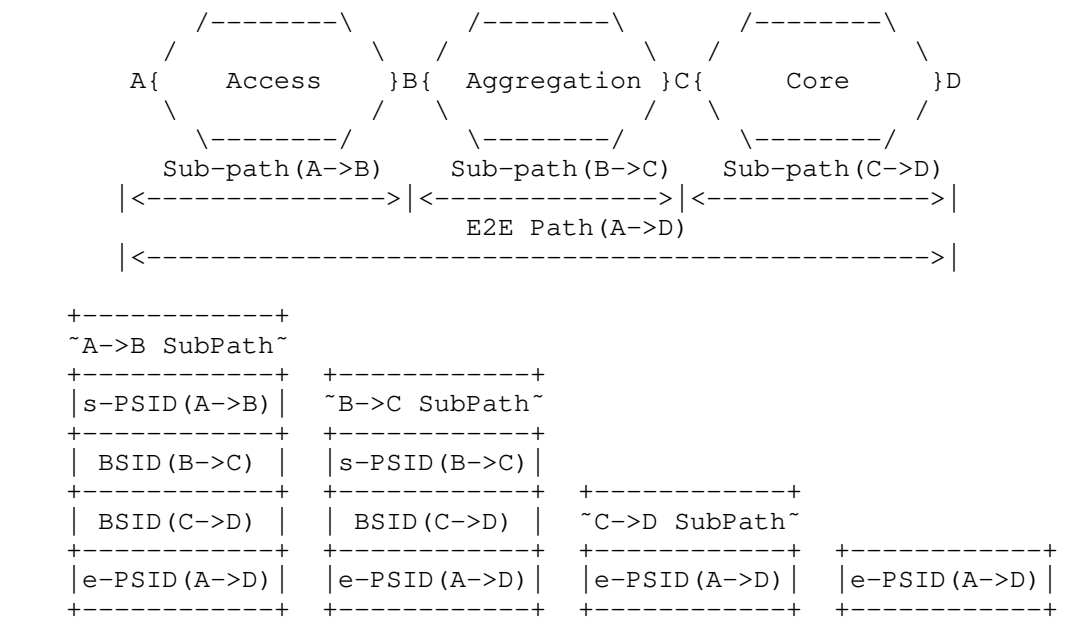


Figure 2: Nesting of Path Segments

4. Path Segment Allocation

Several ways can be used to allocate the Path Segment.

One way is to set up a communication channel (e.g., MPLS Generic Associated Channel (G-ACh)) between the ingress node and the egress node, and the ingress node of the SR path can directly send a request to the egress node to ask for a Path Segment.

Another way is to leverage a centralized controller (e.g., PCE, SDN controller) to assign the Path Segment. PCEP based Path Segment allocation is defined in [I-D.li-pce-sr-path-segment], and SR-policy based path segment allocation is defined in [I-D.li-idr-sr-policy-path-segment-distribution].

5. Path Segment for PM

As defined in [RFC7799], performance measurement can be classified into Active, Passive and Hybrid measurement. For Passive measurement, path identification at the measuring points is the pre-requisite. Path segment can be used by the measuring points (e.g., the ingress/egress nodes of an SR path) or a centralized controller to correlate the packets counts/timestamps that are from the ingress

and egress nodes to a specific SR path, then packet loss/delay can be calculated.

Performance Delay Measurement (DM) and Loss Measurements (LM) in SR networks with MPLS data plane can be found in [I-D.gandhi-spring-sr-mpls-pm] and [I-D.gandhi-spring-udp-pm].

6. Path Segment for Bi-directional SR Path

With the current SR architecture, an SR path is a unidirectional path. In some scenarios, for example, mobile backhaul transport network, there are requirements to support bidirectional path, and the path is normally treated as a single entity and both directions of the path have the same fate, for example, failure in one direction will result in switching at both directions.

MPLS supports this by introducing the concepts of co-routed bidirectional LSP and associated bidirectional LSP. With SR, to support bidirectional path, a straightforward way is to bind two unidirectional SR paths to a single bidirectional path. Path segments can be used to correlate the two unidirectional SR paths at both ends of the paths.

[I-D.li-pce-sr-bidir-path] defines how to use PCEP and Path segment to initiate a bidirectional SR path, and [I-D.li-idr-sr-policy-path-segment-distribution] defines how to use SR policy and Path segment to initiate a bidirectional SR path.

7. Path Segment for End-to-end Path Protection

For end-to-end 1+1 path protection (i.e., Live-Live case), the egress node of an SR path needs to know the set of paths that constitute the primary and the secondary(s), in order to select the primary packet for onward transmission, and to discard the packets from the secondary(s).

To do this, each path needs a path identifier that is unique at the egress node. Depending on the design, this is a single unique path segment label chosen by the egress PE.

There then needs to be a method of binding this path identifiers into equivalence groups such that the egress PE can determine the set of packets that represent a single path and its secondary.

It is obvious that this group can be instantiated in the network by an SDN controller.

8. IANA Considerations

This document does not require any IANA actions.

9. Security Considerations

This document does not introduce additional security requirements and mechanisms other than the ones described in [RFC8402].

10. Contributors

The following individuals also contribute to this document.

- Cheng Li, Huawei

11. Acknowledgements

The authors would like to thank Stewart Bryant, Alexander Vainshtein, Andrew G. Malis and Loa Andersson for their review, suggestions and comments to this document.

The authors would like to acknowledge the contribution from Alexander Vainshtein on "Nesting of Path Segments".

12. References

12.1. Normative References

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-14 (work in progress), June 2018.

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

[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>.

12.2. Informative References

[I-D.gandhi-spring-sr-mpls-pm]
Gandhi, R., Filsfils, C., daniel.voyer@bell.ca, d., Salsano, S., Ventre, P., and M. Chen, "Performance Measurement in Segment Routing Networks with MPLS Data Plane", draft-gandhi-spring-sr-mpls-pm-03 (work in progress), September 2018.

[I-D.gandhi-spring-udp-pm]
Gandhi, R., Filsfils, C., daniel.voyer@bell.ca, d., Salsano, S., Ventre, P., and M. Chen, "UDP Path for In-band Performance Measurement for Segment Routing Networks", draft-gandhi-spring-udp-pm-02 (work in progress), September 2018.

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-14 (work in progress), June 2018.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-01 (work in progress), June 2018.

[I-D.li-idr-sr-policy-path-segment-distribution]
Li, C., Chen, M., Dong, J., and Z. Li, "Segment Routing Policies for Path Segment and Bi-directional Path", draft-li-idr-sr-policy-path-segment-distribution-00 (work in progress), April 2018.

[I-D.li-pce-sr-bidir-path]
Li, C., Chen, M., Dhody, D., Cheng, W., Li, Z., Dong, J., and R. Gandhi, "PCEP Extension for Segment Routing (SR) Bi-directional Associated Paths", draft-li-pce-sr-bidir-path-01 (work in progress), September 2018.

[I-D.li-pce-sr-path-segment]
Li, C., Chen, M., Dhody, D., Cheng, W., Dong, J., Li, Z., and R. Gandhi, "Path Computation Element Communication Protocol (PCEP) Extension for Path Identification in Segment Routing (SR)", draft-li-pce-sr-path-segment-02 (work in progress), September 2018.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <https://www.rfc-editor.org/info/rfc6374>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[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>.

Authors' Addresses

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Lei Wang
China Mobile

Email: wangleiyj@chinamobile.com

Han Li
China Mobile

Email: lihan@chinamobile.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Rakesh Gandhi
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Royi Zigler
Broadcom

Email: royi.zigler@broadcom.com

Shuangping Zhan
ZTE

Email: zhan.shuangping@zte.com.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

J. Dong
S. Bryant
Huawei Technologies
Z. Li
China Mobile
T. Miyasaka
KDDI Corporation
October 22, 2018

Segment Routing for Enhanced VPN Service
draft-dong-spring-sr-for-enhanced-vpn-02

Abstract

Enhanced VPN (VPN+) is an enhancement to VPN technology to enable it to support the needs of new applications, particularly applications that are associated with 5G services. These applications require better isolation from both control and data plane's perspective and have more stringent performance requirements than can be provided with overlay VPNs. The characteristics of an enhanced VPN as perceived by its tenant needs to be comparable to those of a dedicated private network. This requires tight integration between the overlay VPN and the underlay network resources in a scalable manner. An enhanced VPN may form the underpinning of 5G network slicing, but will also be of use in its own right. This document describes the use of segment routing based mechanisms to provide the enhanced VPN service with dedicated network resources. The proposed mechanism is applicable to both SR with MPLS data plane and SR with IPv6 data plane (SRv6).

Status of This Memo

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

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

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

This Internet-Draft will expire on April 25, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

Driven largely by needs arising from the 5G mobile network design, the concept of network slicing has gained traction [NGMN-NS-Concept] [TS23501][TS28530] [BBF-SD406]. Network slicing requires the transport network to support partitioning the network resources to provide the client with dedicated (private) networking, computing,

and storage resources drawn from a shared pool. The slices may be seen as (and operated as) virtual networks.

Thus there is a need to create virtual networks with enhanced characteristics. The tenant of such a virtual network can require a degree of isolation and performance that previously could only be satisfied by dedicated networks. Additionally the tenant may ask for some level of control to their virtual network e.g. to customize the service paths in the network slice.

The enhanced VPN service (VPN+) as described in [I-D.dong-teas-enhanced-vpn] is targeted at new applications which require better isolation from both control plane and data plane's perspective and have more stringent performance requirements than can be provided with existing overlay VPNs. An enhanced VPN may form the underpinning of network slicing, but will also be of use in its own right.

Although each VPN can be associated with a set of dedicated RSVP-TE [RFC3209] LSPs with bandwidth reservation to provide some guarantee to service performance, such mechanisms would introduce per-VPN per-path states into the network, which is known to have scalability issues [RFC5439] and has not been widely adopted in production networks.

Segment Routing (SR) [I-D.ietf-spring-segment-routing] specifies a mechanism to steer packets through an ordered list of segments. It can achieve explicit source routing without introducing per-path state into the network. Like RSVP-TE, SR also supports source specification of the packet path. However, currently SR does not have the capability of reserving or identifying different network resources for different services or customers. Although the controller can have global view of network state and can provision different services onto different SR paths, in the data plane it still relies on traditional DiffServ QoS model to provide coarse-grained traffic differentiation in the network. While this may be sufficient for some traditional services, it cannot meet the requirement of the enhanced VPN service.

This document extends the SR paradigm by allocating different Segment Identifiers (SIDs) to represent the different subset of resources allocated on each network elements (links or nodes). The SIDs associated with particular network resources can be used to construct customized virtual networks for different services, the SID can also be used to steer the service traffic to be processed with the corresponding allocated resources. This mechanism can be used to provide the enhanced VPN service with dedicated network resources.

The proposed mechanism is applicable to both SR with MPLS data plane and SR with IPv6 data plane (SRv6).

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC8174 [RFC8174][when, and only when, they appear in all capitals, as shown here.

3. Segment Routing with Resource Awareness

In segment routing, several types of segments are defined to represent either topological elements or service instructions. A topological segment may be a node segment or an adjacency segment. Some other types of segments may be associated with specific service functions for service chaining purpose. However, so far non of the SR segments are associated with network resources for the QoS purpose.

In order to support the enhanced VPNs which require guaranteed performance and isolation from other services in the network, the overlay VPN needs to been integrated with underlay networks. Some dedicated network resources need to be allocated for enhanced VPN. When segment routing is used to build enhanced VPNs, it is necessary to associate the segments with network resources.

By extending the segment routing paradigm, different set of network resources are allocated by network elements, and associated with dedicated SIDs. On one particular link, multiple adjacency segment identifiers (Adj-SIDs) can be allocated, each of which is associated with a subset of the link resource allocated, such as bandwidth, queues, etc. For one particular node, multiple node-SIDs can be allocated, each of which may be associated with a subset of resource allocated from the node, such as the processing resources. Per-segment resource allocation complies to the SR paradigm, which avoids introducing per-path state into the network.

Different groups of SIDs associated with network resources can be used to build virtual networks for different enhanced VPNs, this provides the required isolation between enhanced VPNs. The adj-SIDs are used to steer traffic of different enhanced VPNs into different set of link resources. The node SIDs can be used to steer traffic of different enhanced VPNs into different node resources. The node SIDs can also be used to build loose SR paths for different enhanced VPNs. In this case, the node-SIDs are used by transit nodes to steer traffic into the local link resources allocated for the corresponding

enhanced VPN. Note in this case Penultimate Hop Popping (PHP) [RFC3031] MUST be disabled, as the node-SID is used to identify the SR virtual network and the corresponding network resources allocated to the enhanced VPN.

4. Control Plane

The architecture described in this document makes use of a centralized controller that collects the information about the network (configuration, state, routing databases, etc.) as well as the service information (traffic matrix, performance statistics, etc). The controller is also responsible for the centralized computation and optimization of the virtual network used for enhanced VPN. A distributed control plane is needed for the collection and distribution of the network topology and state information. Distributed routing computation for some services in the enhanced VPNs is also possible.

5. Procedures

This section describes the procedures of provisioning an enhanced VPN service based on segment routing with resource awareness.

According to the requirement of an enhanced VPN service, a centralized network controller calculates a subset of the underlay network topology to support this enhanced VPN. Within this topology, the network resources needed on each network element can also be determined. The network resources are allocated in a per-segment manner, and are associated with different node-SIDs and adj-SIDs. The group of the node-SIDs and adj-SIDs allocated for the enhanced VPN will be used by network nodes and the network controller to build a SR virtual topology, which is used as the logical underlay of the enhanced VPN service. The extensions to IGP protocol to distribute the SIDs and the associated resources allocated for a virtual network topology is specified in [I-D.dong-lsr-sr-enhanced-vpn].

Suppose that customer requests for an enhanced VPN service from the network operator. The fundamental requirement is that customer A's service does not experience interference from other services in the network, such as other customers' VPN services, or the non-VPN services in the network. The detailed requirements can be described with characteristics such as the following:

- Service topology: the service sites and the connectivity between them

- Service bandwidth: the bandwidth requirement between service sites

- Isolation: the level of isolation from other services in the network
- Reliability: whether fast repair or end-to-end protection is needed or not.
- Latency
- Jitter
- Visibility: the customer may want to have some form of visibility of the network deliversing the service.

5.1. Topology and Resource Computation

As described in section 4, a centralized network controller is responsible for the provisioning of enhanced VPNs. The controller needs to determine the information of network connectivity, network resources, network performance and other relevant network state of the underlay network. This is often done using either IGP [RFC5305] [RFC3630] [RFC7471] [RFC7810] or BGP-LS [RFC7752] [I-D.ietf-idr-te-pm-bgp].

Based on the network information collected from the underlay network, the controller computes the underlay topology (possibly using multiple algorithms) and knows the resources that are available and allocated. When a request is received from a tenant, the controller computes the subgraph of the underlay network, along with the resources to be allocated on each network element (e.g. links and nodes) in the topology to meet the tenant's requirements, whilst maintaining the needs of the existing tenants that are using the same network.

5.2. Network Resource and SID Allocation

According to the output of computation, the network controller instructs the network devices involved in the subgraph to allocate the required network resources for the enhanced VPN. This can be done with either PCEP [RFC5440] or Netconf/YANG [RFC6241] [RFC7950] with necessary extensions. The network resources are allocated in a per-segment manner. In addition, dedicated segment identifiers, e.g. node-SIDs and adj-SIDs are also allocated to represent the network resources allocated for the enhanced VPN on each network segment.

In the forwarding plane, there are multiple ways of allocating or reserving network resources to different enhanced VPNs. For example, FlexE may be used to partition the link resource into different sub-channels to achieve hard isolation between each other. The candidate

data plane technologies of enhanced VPN can be found in [I-D.dong-teas-enhanced-vpn]. The SR SIDs are used as a good abstraction of the various types of network resource reservation mechanisms in the forwarding plane.

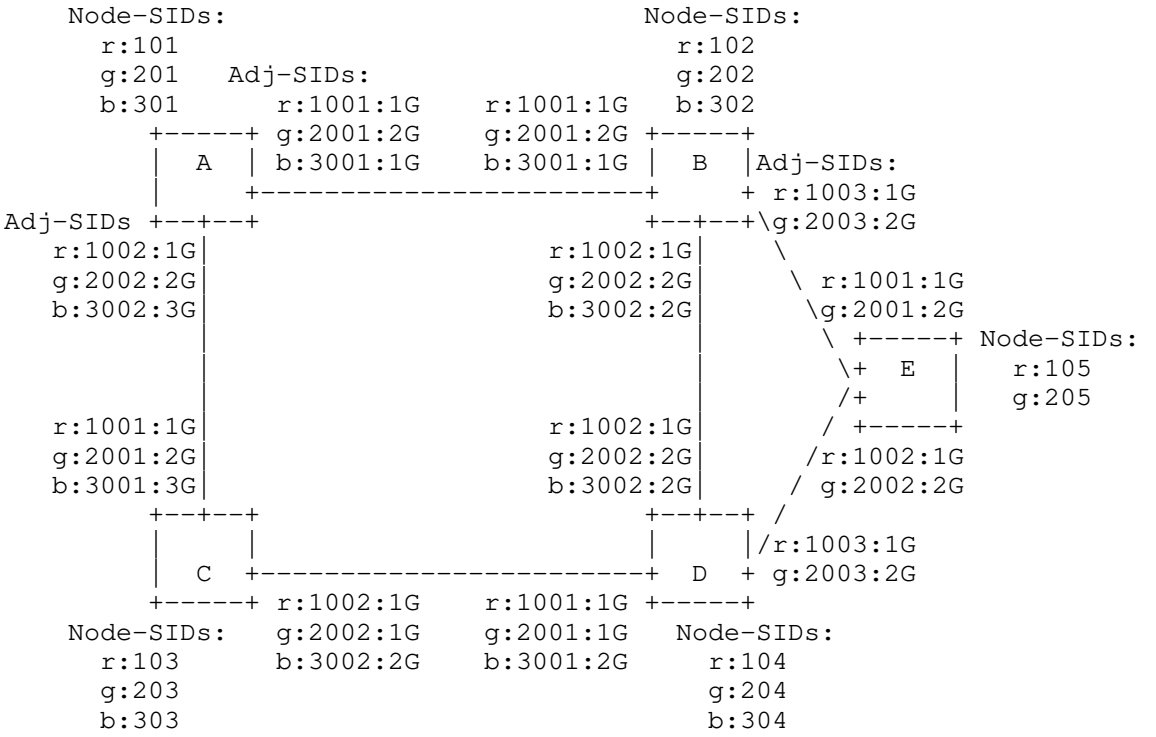


Figure 1. SIDs identify resources allocated to different virtual networks

Figure 1 shows a network fragment of enhanced VPN supported by SR. Note that the format of the SIDs in this figure are for illustration, both SR-MPLS and SRv6 can be utilized as the data plane. In this example, there are three virtual topologies created for enhanced VPNs red (r) , green (g) and blue (b). The red and green topologies consist of nodes A, B, C, D, and E with all their interconnecting links, whilst the blue topology only consists of nodes A, B, C and D with all their interconnecting links. Each node allocates a dedicated adjacency SID for each link participating in a particular topology. Each node is also allocated with a dedicated node SID for each topology it participates in. The adj-SIDs are associated with the link resources (e.g. bandwidth) allocated to each topology, so that the adj-SIDs can be used to steer service of different enhanced VPNs into different set of reserved resources in the data plane. The node-SIDs can be associated with dedicated nodal resources allocated

for each topology. In addition, the node-SIDs of different topologies can be used to build loose SR path within each virtual topology, and steer service of different enhanced VPNs into the different set of reserved resources in the data plane.

In Figure 1, the notation x:nnnn:y that in topology colour x, the adj-SID nnnn will steer the packet over that link which has a total bandwidth of y assigned to that topology. Thus the note r:1002:1G in link C->D says that the red topology over link C->D has a reserved bandwidth of 1Gb/s and will be used by packets arriving at node C with an adj-SID 1002 at the top of the label stack.

5.3. Construction of SR Virtual Networks

Each node MUST advertise its set of resources (allocated and available) and the associated SIDs both to the centralized controller and into the network. This can be achieve by many different means such as (non-exhaustive list) IGP extensions [I-D.dong-lsr-sr-enhanced-vpn], BGP-LS [RFC7752] with possible extensions, NETCONF/YANG [RFC6241] [RFC7950].

With the collected network resource and SIDs information, the controller and network nodes are able to construct the SR virtual topologies and forwarding entries using the node-SIDs and adj-SIDs allocated for each enhanced VPN. Unlike classic segment routing in which network resources are shared by all services and customers, the SR virtual networks are associated with dedicated resource allocated in the underlay, so that they can be used to meet the service requirement of enhanced VPN and provide the required isolation from other services in the same network.

Figure 2 shows the virtual SR topologies created from the underlay network in Figure 1.

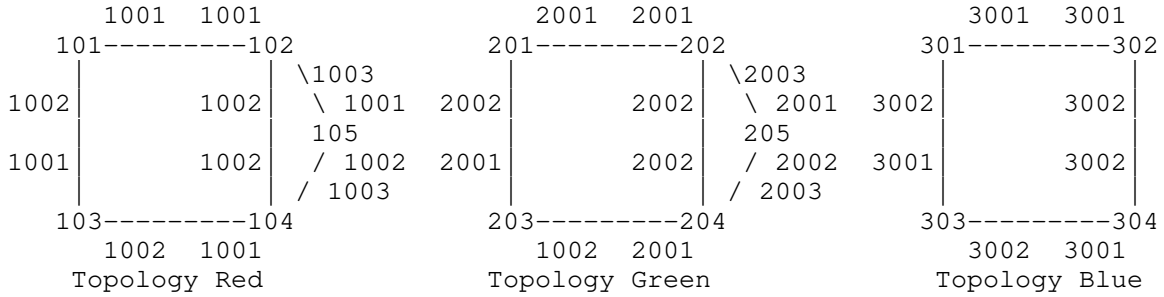


Figure 2. SR virtual topologies using different groups of SIDs

5.4. VPN Service to SR Virtual Network Mapping

The services of an enhanced VPN customer can be provisioned using the customized SR virtual network as the underlay. In this way, services of different enhanced VPNs will only use the network resources allocated and will not interfere with each other. For each enhanced VPN customer, the service paths can be customized for different services within the SR virtual topology, and the allocated network resources are shared by different services of the same enhanced VPN customer.

For example, to create a strict path along the path A-B-D-E in the red topology in Figure 2, the SR segment list in the service packet would be (1001, 1002, 1003). For the same strict path in green topology, the SR segment list would be (2001, 2002, 2003). In the case where we wish to construct a loose path A-D-E in the green topology, the service packet SHOULD be set with the SR segment list (201, 204, 205). At node A the packet is sent towards D via either node B or C using the link and node resources allocated for the green topology. At node D the packet is forwarded to E using the link and node resource allocated for the green topology. Similarly, a packet for the loose path A-D-E in the red topology would arrive at node A with the SID list (101, 104, 105).

5.5. Network Visibility to Customer

The tenants of enhanced VPNs may request different granularity of visibility to the network which deliver the service. Depending on the requirement, the network can be exposed to the tenant either as a virtual network topology, or a set of computed paths with transit nodes, or simply the connectivity between endpoints without any path information. The visibility can be delivered through different possible mechanisms, such as IGPs (e.g. IS-IS, OSPF) or BGP-LS. In addition, the network operator may want to restrict the visibility of the information it delivers to the tenant by either hiding the transit nodes between sites (and only delivering the endpoints connectivity) or by hiding portions of the transit nodes (summarizing the path into fewer nodes). Mechanisms such as BGP-LS allow the flexibility of the advertisement of aggregated network information.

6. Benefits of the Proposed Mechanism

The proposed mechanism provides several key characteristics:

- Flexibility
- Scalability

- Resource isolation

In addition to isolation, the proposed mechanism allows resource sharing between different services of the same enhanced VPN customer. This gives the customer more flexibility and control in service planning and provisioning, the experience would be similar to using a dedicated private network. The performance of critical services flows in a particular enhanced VPN can be further ensured using the mechanisms defined in [DetNet].

The detailed comparison with other candidates technologies are given in the following subsections.

6.1. MPLS-TP

MPLS-TP could be enhanced to include the allocation of specific resources along the path to a specific LSP. This would require that the SDN system set up and maintain every resource at every path for every customer, and map this to the LSP in the data plane, hence at every hop unique LSP label is needed for each path. Whilst this would be a way to produce a proof of concept for network slicing of an MPLS underlay, delegation would be difficult, resulting in a high overhead and a system needing too much administration. This leads to scaling concerns. The number of labels needed at any node would be the total number of services passing through that node. Experience with early pseudowire designs shows that this can lead to scaling issues.

6.2. RSVP-TE

RSVP-TE has the same scaling concern as MPLS-TP in terms of the number of LSPs that need to be maintained being equal to the number of services passing through any given node. It also has the two RSVP disadvantages that basic SR seeks to address:

- The use of RSVP for path establishment in addition to the routing protocol used to discover the topology and the network resources.

- The overhead of the soft-state maintenance associated with RSVP. The impact of this overhead would be exacerbated by the increased number of end to end paths requiring state maintenance.

6.3. Basic SR

Compared to RSVP, SR reduces the number of control protocols to only the routing protocol. It also attempts to minimize the core state by pushing state into the packet, although in some cases the binding SIDs are required to overcome the limitations in the ability of some

nodes to push large label stacks. Moreover, currently SR does not support resource allocation or identification below the level of link, and none at node level. This restricts the extent to which some particular tenant traffic can be isolated from other traffic in the network.

6.4. SR with Resource Awareness

The approach described in this document seeks to achieve a compromise between the state limitations of traditional TE systems and the lack of resource awareness in basic SR.

By segmenting the path and allocating network resources to each element of the virtual network topologies, the operator can choose the granularity of resource to path binding within a virtual topology. In network segments where resource is scarce such that the service requirement may not always be met, the SR approach can allocate specific resources to a particular high priority service. By contrast, in other parts of the network where resource is plentiful, the resource may be shared by a number of services. The decision to do this is in the hands of the operator. Because of the segmented nature of the path, resource aggregation is possible in a way that is more difficult with RSVP-TE and MPLS-TP due to the use of dedicated label to identify each end-to-end path.

7. Service Assurance

In order to provide service assurance it is necessary to instrument the network at multiple levels. The network operator needs to ascertain that the underlay is operating correctly. A tenant needs to ascertain that their services are correctly operating. In principle these can use existing techniques. These are well known problems and solutions either exist or are in development to address them.

New work is needed to instrument the virtual networks that are created. Such instrumentation needs to operate without causing disruption to other services using the network. Given the sensitivity of some applications, care needs to be taken to ensure that the instrumentation itself does not cause disruption either to the service being instrumented or to other services.

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

The normal security considerations of VPNs are applicable and it is assumed that industry best practise is applied to an enhanced VPN.

The security considerations of segment routing are applicable and it is assumed that these are applied to an enhanced VPN that uses SR.

Some applications of enhanced VPNs are sensitive to packet latency; the enhanced VPNs provisioned to carry their traffic have latency SLAs. By disrupting the latency of such traffic an attack can be directly targeted at the customer application, or can be targeted at the network operator by causing them to violate their SLA, triggering commercial consequences. Dynamic attacks of this sort are not something that networks have traditionally guarded against, and networking techniques need to be developed to defend against this type of attack. By rigorously policing ingress traffic and carefully provisioning the resources provided to critical services this type of attack can be prevented. However case needs to be taken when providing shared resources, and when the network needs to be reconfigured as part of ongoing maintenance or in response to a failure.

The details of the underlay MUST NOT be exposed to third parties, to prevent attacks aimed at exploiting a shared resource.

10. Acknowledgements

The authors would like to thank Mach Chen, Zhenbin Li, Stefano Previdi and Charlie Perkins for the discussion and suggestions to this document.

11. References

11.1. Normative References

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

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

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

11.2. Informative References

[BBF-SD406] "BBF SD-406: End-to-End Network Slicing", 2016, <https://wiki.broadband-forum.org/display/BBF/SD-406+End-to-End+Network+Slicing>.

[DetNet] "DetNet WG", 2016, <https://datatracker.ietf.org/wg/detnet>.

[I-D.dong-lsr-sr-enhanced-vpn] Dong, J. and S. Bryant, "IGP Extensions for Segment Routing based Enhanced VPN", draft-dong-lsr-sr-enhanced-vpn-00 (work in progress), June 2018.

[I-D.dong-teas-enhanced-vpn] Dong, J., Bryant, S., Li, Z., and T. Miyasaka, "A Framework for Enhanced Virtual Private Networks (VPN+)", draft-dong-teas-enhanced-vpn-02 (work in progress), October 2018.

[I-D.ietf-idr-te-pm-bgp] Ginsberg, L., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP-LS Advertisement of IGP Traffic Engineering Performance Metric Extensions", draft-ietf-idr-te-pm-bgp-14 (work in progress), October 2018.

[NGMN-NS-Concept] "NGMN NS Concept", 2016, <https://www.ngmn.org/fileadmin/user_upload/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf>.

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

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

[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>.

[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>.

[RFC5439] Yasukawa, S., Farrel, A., and O. Komolafe, "An Analysis of Scaling Issues in MPLS-TE Core Networks", RFC 5439, DOI 10.17487/RFC5439, February 2009, <https://www.rfc-editor.org/info/rfc5439>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

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

[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>.

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

[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>.

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

[TS28530] "3GPP TS28.530", 2016,
<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273>.

Authors' Addresses

Jie Dong
Huawei Technologies

Email: jie.dong@huawei.com

Stewart Bryant
Huawei Technologies

Email: stewart.bryant@gmail.com

Zhenqiang Li
China Mobile

Email: li_zhenqiang@hotmail.com

Takuya Miyasaka
KDDI Corporation

Email: ta-miyasaka@kddi.com

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: June 6, 2021

J. Dong
Huawei Technologies
S. Bryant
Futurewei Technologies
T. Miyasaka
KDDI Corporation
Y. Zhu
China Telecom
F. Qin
Z. Li
China Mobile
F. Clad
Cisco Systems

December 3, 2020

Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN
draft-dong-spring-sr-for-enhanced-vpn-12

Abstract

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called "segments". A segment can represent topological or service based instructions. A segment can further be associated with network resources allocated for executing the instruction. Such a segment is called resource-aware SID.

Resource-aware SIDs may be used to build SR paths with a set of reserved network resources. In addition, resource-aware SIDs may be used to build SR based virtual underlay networks, which can provide the customized network topology and resource attributes required by different customers and/or services. Such virtual networks are called SR based Virtual Transport Networks (VTNs). The SR based VTNs can be used as the underlay network to enable services with required topology and resource characteristics. This document describes a suggested use of resource-aware SIDs to build SR based VTNs.

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 6, 2021.

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) [RFC8402] specifies a mechanism to steer packets through an ordered list of segments. A segment is referred to by its Segment Identifier (SID). With SR, explicit source routing can be achieved without introducing per-path state into the network. When compared with RSVP-TE [RFC3209], SR currently does not have the capability to reserve network resources or identify different sets of network resources reserved for different customers and/or services. [I-D.ietf-spring-resource-aware-segments] proposes to extend SR by associating SIDs with network resource attributes, (e.g. bandwidth, processing or storage resources). On a network segment, multiple resource-aware SIDs may be allocated, each of which represents a subset of network resources assigned to meet the requirements of one or a group of customers and/or services.

Once allocated, Resource-aware SIDs can be used to build SR paths using a set of reserved network resources. In addition, a group of resource-aware SIDs can be used to build SR based virtual networks with customized network topology and resource attributes. In this document, such virtual networks are called SR based Virtual Transport Networks (VTNs), and can be used to enable services with required topology and resource characteristics, such as the enhanced VPN (VPN+) services as described in [I-D.ietf-teas-enhanced-vpn].

This document describes a suggested use of resource-aware SIDs to build SR based VTNs. Although the procedure is illustrated using SR-MPLS, the proposed mechanism is applicable to both segment routing over MPLS data plane (SR-MPLS) and segment routing over IPv6 data plane (SRv6).

2. Resource-Aware SIDs for VTN

When SR is used as the data plane to provide multiple VTNs in one network, it is necessary to compute and instantiate SR paths with the topology constraints of the VTN, and from the set of network resources allocated to the VTN.

With the mechanism defined in [I-D.ietf-spring-resource-aware-segments], multiple SR SIDs can be allocated for each network segment, with each SID used to identify both the network topological instruction, and the set of network resources allocated for a VTN. The mechanisms to identify the network topology or path with a SID as defined in [RFC8402] are reused.

The control plane mechanisms for advertising resource-aware SIDs for different VTNs may be based on [RFC4915], [RFC5120] and

[I-D.ietf-lsr-flex-algo] with necessary extensions. This is further described in section 3.3.

2.1. SR-MPLS based VTN

This section describes a mechanism of allocating resource-aware SIDs to SR-MPLS based VTNs.

For one IGP link, multiple Adj-SIDs are allocated, each of which is associated with a VTN that link participates in, and represents a subset of the link resources allocated to the VTN. Similarly, for one IGP node, multiple prefix-SIDs are allocated, each of which is associated with a VTN the node participates in, and represents a subset of the node level processing resources allocated to the VTN.

In the case of multi-domain VTNs, on an inter-domain link, multiple BGP peering SIDs [I-D.ietf-idr-bgpls-segment-routing-epe] are allocated, each of which is associated with a VTN which spans multiple domains, and represents a subset of resources allocated on the inter-domain link.

2.2. SRv6 based VTN

This section describes a mechanism of allocating resource-aware SIDs to VTN based on SRv6.

For a network node, multiple SRv6 Locators are allocated, each of which is associated with a VTN that node participates in, and represents a subset of the network resources allocated by the network node to the VTN. The SRv6 SIDs associated with a VTN are allocated from the SID space using the VTN-specific Locators as the prefix. These SRv6 SIDs can be used to represent VTN-specific SRv6 functions which are executed using the network resources allocated to the VTN.

2.3. Scalability Considerations

Note that the introduction of SR based VTNs increases the number of SIDs and SRv6 Locators needed in a network, there may be some concern, especially about the prefix-SIDs, which are allocated from the Segment Routing Global Block (SRGB). The amount of network state will also increase accordingly. However, based on the SR paradigm, resource-aware SIDs and the associated network state are allocated and maintained per VTN, and per-path network state is avoided in the SR network.

3. Procedures

This section describes possible procedures for creating SR based VTNs and the corresponding forwarding tables and entries. Although it is illustrated using SR-MPLS, the proposed mechanism is applicable to both SR-MPLS and SRv6.

Suppose a virtual network is requested by some customer or service. One of the basic requirement is that customer or service is allocated with some dedicated network resource, so that it does not experience unexpected interference from other services in the same network. Other possible requirements may include the required topology, bandwidth, latency, reliability, etc.

According to the received service requirement, a centralized network controller calculates a subset of the underlay network topology to support the service. Within this topology, the set of network resources required on each network element is also determined. The subset of network topology and network resources together constitute a VTN. Depending on the service requirement, the network topology and resource can be dedicated for an individual customer or service, or can be shared by a group of customers and/or services.

Based on the mechanisms defined in [I-D.ietf-spring-resource-aware-segments], the network topology and resources of a VTN can be represented by a group of resource-aware SIDs. With SR-MPLS, a group of prefix-SIDs and adj-SIDs will be used by network nodes and the network controller to construct an SR based VTN, which will be used as the virtual underlay network for the requested service. Control plane protocols such as IGP (e.g. IS-IS or OSPF) and BGP-LS can be used to distribute the SIDs and the associated resource information of each VTN. The detailed control plane mechanisms and possible extensions are out of the scope of this document.

3.1. VTN Topology and Resource Planning

A centralized network controller can be responsible for the planning of a VTN to meet the received service request. The controller needs to collect information on network connectivity, network resources, network performance and any other relevant network states from the underlay network. This can be done using either IGP TE extensions such as [RFC5305] [RFC3630] [RFC7471] [RFC8570], or BGP-LS [RFC7752] [RFC8571], or any other form of control plane signaling.

Based on the information collected from the underlay network, the controller obtains the underlay network topology and the information about the allocated and available network resources. When a service

request is received, the controller determines the subset of the network topology, along with the set of the resources needed on each network segment (e.g. links and nodes) in the topology to meet the service requirements, whilst maintaining the needs of the existing services that are using the same network. The subset of network topology and network resources constitute a VTN, which will be used as the virtual underlay network of the requested service.

3.2. VTN Network Resource and SID Allocation

According to the result of VTN planning, the network controller instructs the network nodes with the information of the VTN identifier and the required network resources to be allocated to the VTN, so that the involved network nodes could join the VTN and allocate the network resources for the VTN accordingly. This may be done with PCEP [RFC5440], Netconf/YANG [RFC6241] [RFC7950] or with any other control plane mechanism with necessary extensions. Thus, the controller not only allocates the resources to the newly computed VTN but also keeps track of the remaining available resources in order to cope with subsequent VTN requests.

On each network node involved in a VTN, a set of network resources are allocated to that VTN. Such set of network resources can be dedicated for the processing of traffic in that VTN, and cannot be used for traffic in other VTNs. Note it is also possible that a group of VTNs may share a set of network resources on some network segments. Resource-aware SIDs are allocated to represent the set of resources allocated on the network node and the attached links. Such group of resource-aware SIDs, e.g. prefix-SIDs and adj-SIDs are used as the data plane identifiers of the node and links in the VTN.

In the underlying forwarding plane, there can be multiple ways of allocating a subset of network resources to a VTN. The candidate data plane technologies to support resource partitioning or reservation can be found in [I-D.ietf-teas-enhanced-vpn]. The resource-aware SIDs are considered as a unified abstraction in the network layer, which can work with various network resource partition or reservation mechanisms in the underlying forwarding plane.

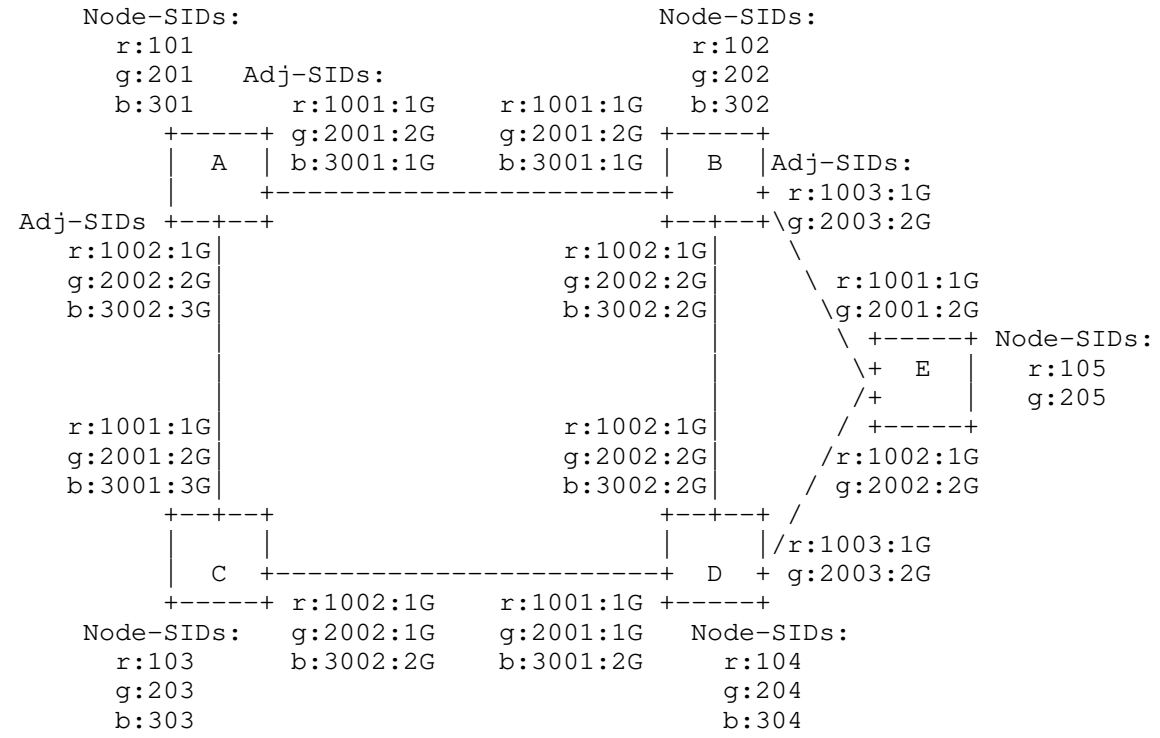


Figure 1. SID and resource allocation for multiple VTNs

Figure 1 shows an example of providing multiple VTNs in an SR based network. Note that the format of the SIDs in this figure is for illustration, both SR-MPLS and SRv6 can be used as the data plane. In this example, three VTNs: red (r) , green (g) and blue (b) are created to carry traffic of different customers or services. Both the red and green VTNs consist of nodes A, B, C, D, and E with all their interconnecting links, whilst the blue VTN only consists of nodes A, B, C and D with all their interconnecting links. Note that different VTNs may have a set of shared nodes and links. On each link, a resource-aware adj-SID is allocated for each VTN it participates in.

In Figure 1, the notation x:nnnn:y means that in VTN x, the adj-SID nnnn will steer the packet over a link which has bandwidth y reserved for that VTN. For example, r:1002:1G in link C->D says that the VTN red has a reserved bandwidth of 1Gb/s on link C->D, and will be used by packets arriving at node C with an adj-SID 1002 at the top of the label stack. Similarly, on each node, a resource-aware prefix-SID is allocated for each VTN it participates in. The adj-SIDs can be associated with different set of link resources (e.g. bandwidth)

allocated to different VTNs, so that the adj-SIDs can be used to steer service traffic into different set of link resources in packet forwarding. The prefix-SIDs can be associated with the nodal resources allocated to different VTNs. In addition, the prefix-SIDs can be used to build loose SR path within a VTN, in this case it can be used by the transit nodes to steer service traffic into the set of local network resources allocated to the VTN.

3.3. Construction of SR based VTNs

The network controller needs to obtain the information of all the VTNs in the network it oversees, and the network nodes need to obtain the information of the VTNs they participate in. To achieve this, each network node needs to advertise the identifiers of the VTNs it participates in, together with the group of SIDs and the associated resource attributes both to other network nodes and to the controller.

[I-D.dong-lsr-sr-enhanced-vpn] defines an IGP mechanism to advertise the customized topology and resource attributes of VTN, which allows flexible combination of the virtual network topology and the network resources attribute to provide a relatively large number of VTNs. The corresponding BGP-LS mechanism used to distribute the VTN information to the controller is described in [I-D.dong-idr-bgpls-sr-enhanced-vpn].

For network scenarios which require less flexibility or scalability, the simplified control plane mechanisms based on Multi-Topology [RFC5120] or Flex-Algo [I-D.ietf-lsr-flex-algo] are described in [I-D.xie-lsr-isis-sr-vtn-mt] and [I-D.zhu-lsr-isis-sr-vtn-flexalgo] respectively. The corresponding BGP-LS mechanisms used to distribute the VTN information to the controller are described in [I-D.xie-idr-bgpls-sr-vtn-mt] and [I-D.zhu-idr-bgpls-sr-vtn-flexalgo] respectively.

Based on the collected information of the topology, the allocated network resources and the associated SIDs of VTNs, both the controller and network nodes can construct the SR based VTNs and generate the forwarding tables and entries for each VTN based on the SIDs and SRv6 Locators of each VTN. Unlike classic segment routing in which network resources on a network segment are shared by all the SR traffic, different SR VTNs can be associated with different set of resources allocated in the underlay forwarding plane, so that they can be used to provide the required resource isolation between different customers and/or services in the same network.

Figure 2 shows the SR based VTNs created in the network in Figure 1.

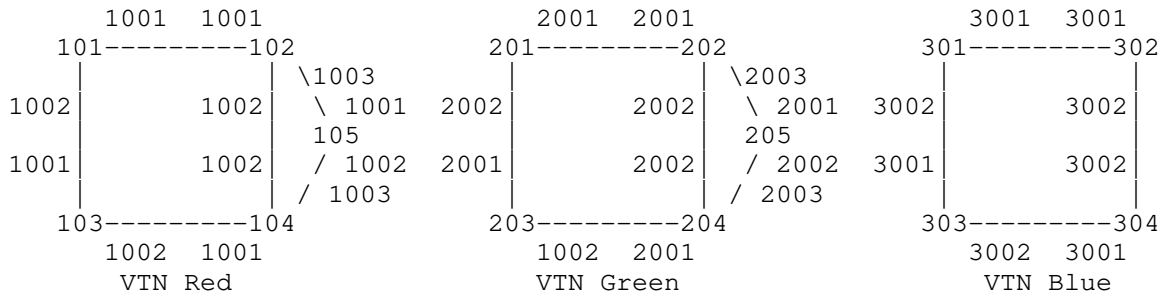


Figure 2. SR based VTNs with different groups of SIDs

For each SR based VTN, SR paths are computed within the VTN, taking the VTN topology and resources as constraints. The SR path can be an explicit path instantiated using SR policy [I-D.ietf-spring-segment-routing-policy], in which the SID-list is built only with the SIDs allocated to the VTN. The SR path can also be an IGP computed path associated with a prefix-SID or SRv6 End SID allocated by a node for the VTN, the IGP computation is also based on the VTN constraints. Different SR paths in the same VTN may use shared network resources when they use the same resource-aware SIDs allocated to the VTN, while SR paths in different VTNs can be steered to use different set of network resources over the shared network links or nodes. These VTN-specific SR paths need to be installed in the corresponding forwarding tables.

For example, to create an explicit path A-B-D-E in VTN red in Figure 2, the SR SID-list encapsulated in the service packet would be (1001, 1002, 1003). For the same explicit path A-B-D-E in VTN green, the SR segment list would be (2001, 2002, 2003). In the case where we wish to construct a loose path A-D-E in VTN green, the service packet SHOULD be encapsulated with the SR SID-list (201, 204, 205). At node A, the packet can be sent towards D via either node B or C using the link and node resources allocated for VTN green. At node D the packet is forwarded to E using the link and node resource allocated for VTN green. Similarly, a packet to be sent via loose path A-D-E in VTN red would be encapsulated with segment list (101, 104, 105). In the case where an IGP computed path can meet the service requirement, the packet can be simply encapsulated with the prefix-SID of egress node E in the corresponding VTN.

3.4. Mapping Service to SR based VTN

Network services can be provisioned using SR based VTNs as the virtual underlay networks. For example, different services may be provisioned in different SR based VTNs, each of which would use the network resources allocated to the VTN, so that they will not

interfere with each other. In another case, a group of services which have similar characteristics and requirements may be provisioned in the same VTN, in this case the network resources allocated to the VTN are only shared among this group of services, but will not be shared with other services in the network. The steering of service traffic to SR based VTNs can be based on either local policy or the mechanisms as defined in [I-D.ietf-spring-segment-routing-policy].

3.5. VTN Visibility to Customer

The customers may request different granularity of visibility to the VTN which deliver the service. Depending on the requirement, the network can be exposed to the customer either as a virtual network with both the edge nodes and the intermediate nodes, or a set of paths with some of the transit nodes, or simply a set of virtual connectivity between endpoints without any transit node information. The visibility may be delivered through different possible mechanisms, such as IGPs (e.g. IS-IS, OSPF), BGP-LS or Netconf/YANG. On the other hand, network operators may want to restrict the visibility of the network information it delivers to the customer by either hiding the transit nodes between sites (and only delivering the endpoints connectivity), or by hiding portions of the transit nodes (summarizing the path into fewer nodes). Mechanisms such as BGP-LS allow the flexibility of the advertisement of aggregated virtual network information.

4. Characteristics of SR based VTN

The proposed mechanism provides several key characteristics:

- Customization: Different customized VTNs can be created in a shared network to meet different customers' connectivity and service requirement. Each customer is only aware of the topology and attributes of his own VTN, and provision services on the VTN instead of the shared physical network. This provides an practical mechanism to support network slicing.

- Resource Isolation: The computation and instantiation of SR paths in one VTN can be independent from other VTNs or other services in the network. In addition, a VTN can be associated with a set of dedicated network resources, which can avoid resource competition and performance interference from other VTNs or other services in the network. The proposed mechanism also allows resource sharing between different service flows of the same customer, or between a group of services which are provisioned in the same VTN. This gives the operators and the customers the flexibility in network planning and service provisioning. In a VTN, the performance of

critical services can be further ensured using other mechanisms, e.g. those as defined in [DetNet].

- o Scalability: The introduction of resource aware SIDs for different VTNs would increase the amount of SIDs and state in the network. While the increased network state is considered an inevitable price in meeting the requirements of some customers or services, the SR based VTN mechanism seeks to achieve a balance between the state limitations of traditional end-to-end TE mechanism and the lack of resource awareness in classic segment routing. Following the segment routing paradigm, network resources are allocated on network segments in a per VTN manner and represented as SIDs, this ensures that there is no per-path state introduced in the network. In addition, operators can choose the granularity of resource allocation on different network segments. In network segments where resource is scarce such that the service requirement may not always be met, the proposed approach can be used to allocate a set of resources to a VTN which contains such network segment to avoid possible competition. By contrast, in other segment of the network where resource is considered plentiful, the resource may be shared between a number of VTNs. The decision to do this is in the hands of the operator. Because of the segmented nature of the SR based VTN, resource aggregation is easier and more flexible than RSVP-TE based approach.

5. Service Assurance of VTN

In order to provide assurance for services provisioned in the SR based VTNs, it is necessary to instrument the network at multiple levels, e.g. in both the underlay network level and the VTN level. The operator or the customer may also monitor and measure the performance of the services carried by the VTN. In principle these can be achieved using existing or in development techniques in IETF. The detailed mechanisms are out of the scope of this document.

In case of failure or service performance degradation happens in a VTN, it is necessary that some recovery mechanisms, e.g. local protection or end-to-end protection mechanism is used to switch the traffic to another path in the same VTN which could meet the service performance requirement. Care must be taken that the service or path recovery mechanism in one VTN does not impact other VTNs in the same network.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

The security considerations of segment routing and resource-aware SIDs are applicable to this document.

The SR VTNs may be used carry services with specific SLA parameters. An attack can be directly targeted at the customer application by disrupting the SLA, and can be targeted at the network operator by causing them to violate their SLA, triggering commercial consequences. By rigorously policing ingress traffic and carefully provisioning the resources provided to the VTN, this type of attack can be prevented. However care needs to be taken when shared resources are provided between VTNs at some point in the network, and when the network needs to be reconfigured as part of ongoing maintenance or in response to a failure.

The details of the underlying network should not be exposed to third parties, some abstraction would be needed, this is also to prevent attacks aimed at exploiting a shared resource between VTNs.

8. Contributors

Zhenbin Li
Email: lizhenbin@huawei.com

Zhibo Hu
Email: huzhibo@huawei.com

9. Acknowledgements

The authors would like to thank Mach Chen, Stefano Previdi, Charlie Perkins, Bruno Decraene, Loa Andersson, Alexander Vainshtein, Joel Halpern and James Guichard for the valuable discussion and suggestions to this document.

10. References

10.1. 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>.

10.2. Informative References

[DetNet] "DetNet WG", 2016, <https://datatracker.ietf.org/wg/detnet>.

[I-D.dong-idr-bgpls-sr-enhanced-vpn] Dong, J., Hu, Z., Li, Z., Tang, X., and R. Pang, "BGP-LS Extensions for Segment Routing based Enhanced VPN", draft-dong-idr-bgpls-sr-enhanced-vpn-02 (work in progress), June 2020.

[I-D.dong-lsr-sr-enhanced-vpn] Dong, J., Hu, Z., Li, Z., Tang, X., Pang, R., JooHeon, L., and S. Bryant, "IGP Extensions for Segment Routing based Enhanced VPN", draft-dong-lsr-sr-enhanced-vpn-04 (work in progress), June 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-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-spring-resource-aware-segments] Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", draft-ietf-spring-resource-aware-segments-00 (work in progress), July 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-09 (work in progress), November 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-26 (work in progress), November 2020.

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

[I-D.xie-idr-bgpls-sr-vtn-mt]
Xie, C., Li, C., Dong, J., and Z. Li, "BGP-LS with Multi-topology for Segment Routing based Virtual Transport Networks", draft-xie-idr-bgpls-sr-vtn-mt-01 (work in progress), July 2020.

[I-D.xie-lsr-isis-sr-vtn-mt]
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", draft-xie-lsr-isis-sr-vtn-mt-02 (work in progress), October 2020.

[I-D.zhu-idr-bgpls-sr-vtn-flexalgo]
Zhu, Y., Dong, J., and Z. Hu, "BGP-LS with Flex-Algo for Segment Routing based Virtual Transport Networks", draft-zhu-idr-bgpls-sr-vtn-flexalgo-00 (work in progress), March 2020.

[I-D.zhu-lsr-isis-sr-vtn-flexalgo]
Zhu, Y., Dong, J., and Z. Hu, "Using Flex-Algo for Segment Routing based VTN", draft-zhu-lsr-isis-sr-vtn-flexalgo-01 (work in progress), September 2020.

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

[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>.

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

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

[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>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

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

[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>.

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

[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>.

[RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <https://www.rfc-editor.org/info/rfc8571>.

Authors' Addresses

Jie Dong
Huawei Technologies

Email: jie.dong@huawei.com

Stewart Bryant
Futurewei Technologies

Email: stewart.bryant@gmail.com

Takuya Miyasaka
KDDI Corporation

Email: ta-miyasaka@kddi.com

Yongqing Zhu
China Telecom

Email: zhuyq8@chinatelecom.cn

Fengwei Qin
China Mobile

Email: qinfengwei@chinamobile.com

Zhenqiang Li
China Mobile

Email: li_zhenqiang@hotmail.com

Francois Clad
Cisco Systems

Email: fclad@cisco.com

SPRING Working Group
Internet-Draft
Intended Status: Informational
Expires: March 18, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
P. L. Ventre
CNIT
M. Chen
Huawei

September 14, 2018

Performance Measurement in Segment Routing Networks with MPLS Data Plane draft-gandhi-spring-sr-mpls-pm-03

Abstract

RFC 6374 specifies protocol mechanisms to enable the efficient and accurate measurement of packet loss, one-way and two-way delay, as well as related metrics such as delay variation in MPLS networks using probe messages. This document reviews how these mechanisms can be used for Delay and Loss Performance Measurements (PM) in Segment Routing (SR) networks with MPLS data plane (SR-MPLS), for both SR links and end-to-end SR Policies. The performance measurements for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and are advertised in the network using routing protocol extensions.

Status of This Memo

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

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

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

Copyright Notice

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

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

Table of Contents

1. Introduction

Service provider's ability to satisfy Service Level Agreements (SLAs) depend on the ability to measure and monitor performance metrics for packet loss and one-way and two-way delay, as well as related metrics such as delay variation. The ability to monitor these performance metrics also provides operators with greater visibility into the performance characteristics of their networks, thereby facilitating planning, troubleshooting, and network performance evaluation.

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics in MPLS networks using probe messages. 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. However, mechanisms defined in [RFC6374] are more suitable for Segment Routing (SR) when using MPLS data plane (SR-MPLS). The [RFC6374] also supports IEEE 1588 timestamps [IEEE1588] and "direct mode" Loss Measurement (LM), which are required in SR networks.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe replies over an UDP return path when receiving RFC 6374 based probe queries. These procedures can be used to send out-of-band PM replies for both SR links and SR Policies [I-D.spring-segment-routing-policy] for one-way measurement.

This document reviews how probe based mechanisms defined in [RFC6374] can be used for Delay and Loss Performance Measurements (PM) in SR networks with MPLS data plane, for both SR links and end-to-end SR Policies. The performance measurements for SR links are used to compute extended Traffic Engineering (TE) metrics for delay and loss and are advertised in the network using routing protocol extensions.

2. Conventions Used in This Document

2.1. Abbreviations

ACH: Associated Channel Header.

DFLag: Data Format Flag.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

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

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

PM: Performance Measurement.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

TC: Traffic Class.

TE: Traffic Engineering.

URO: UDP Return Object.

2.2. Reference Topology

In the reference topology shown in Figure 1, the querier node R1 initiates a performance measurement probe query and the responder node R5 sends a probe response for the query message received. The probe response is typically sent to the querier node R1. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.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.spring-sr-p2mp-policy].

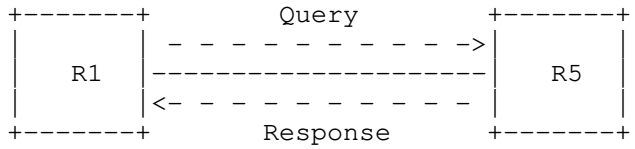


Figure 1: Reference Topology

Both delay and loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in Section 2.4 of [RFC6374]. Transmit and Receive packet loss measurements are defined in Section 2.2 and Section 2.6 of [RFC6374]. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

3. Probe Query and Response Packets

3.1. Probe Packet Header for SR-MPLS Policies

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for an end-to-end measurement for SR Policy contains SR-MPLS label stack [I-D.spring-segment-routing-policy], with the G-ACh Label (GAL) at the bottom of the stack. The GAL is followed by an Associated Channel Header (ACH), which identifies the message type and the message payload following the ACH as shown in Figure 2.

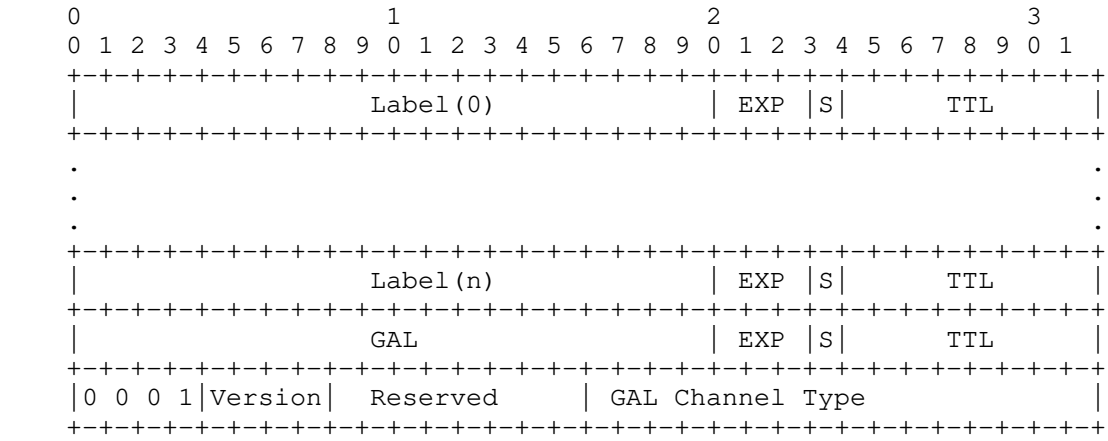


Figure 2: Probe Packet Header for an End-to-end SR-MPLS Policy

The SR-MPLS label stack can be empty to indicate Implicit NULL label case.

3.2. Probe Packet Header for SR-MPLS Links

As described in Section 2.9.1 of [RFC6374], MPLS PM probe query and

response messages flow over the MPLS Generic Associated Channel (G-ACh). A probe packet for SR-MPLS links contains G-ACh Label (GAL). The GAL is followed by an Associated Channel Header (ACH), which identifies the message type, and the message payload following the ACH as shown in Figure 3.

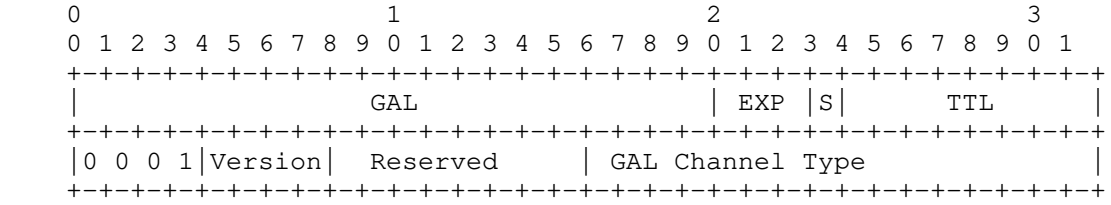


Figure 3: Probe Packet Header for an SR-MPLS Link

3.3. Probe Response Message for SR-MPLS Links and Policies

3.3.1. One-way Measurement Probe Response Message

For one-way performance measurement [RFC7679], the PM querier node can receive "out-of-band" probe replies by properly setting the UDP Return Object (URO) TLV in the probe query message. The URO TLV (Type=131) is defined in [RFC7876] and includes the UDP-Destination-Port and IP Address. In particular, if the querier sets its own IP address in the URO TLV, the probe response is sent back by the responder node to the querier node. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

3.3.2. Two-way Measurement Probe Response Message

For two-way performance measurement [RFC6374], when using a bidirectional channel, the probe response message is sent back to the querier node in-band on the reverse direction SR Link or SR Policy using a message with format similar to their probe query message. In this case, the "control code" in the probe query message is set to "in-band response requested".

A path segment identifier [I-D.spring-mpls-path-segment] [I-D.pce-sr-path-segment] of the forward SR Policy can be used to find the reverse SR Policy to send the probe response message.

4. Performance Delay Measurement

4.1. Delay Measurement Message Format

As defined in [RFC6374], MPLS DM probe query and response messages use Associated Channel Header (ACH) (value 0x000C for delay measurement) [RFC6374], which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR Policies, the same MPLS DM ACH value is used.

The DM message payload as defined in [RFC6374] is used for SR-MPLS delay measurement, for both SR links and end-to-end SR Policies. The DM message payload format is defined as following in [RFC6374]:

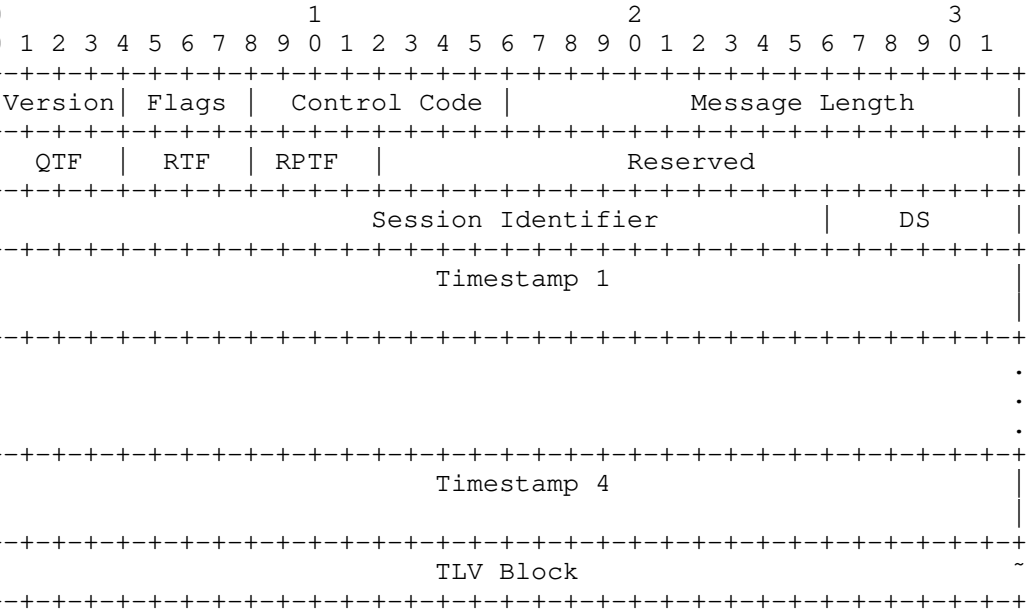


Figure 4: Delay Measurement Message Payload Format

The meanings of the fields are summarized in the following table, see [RFC6374] for details.

Field	Meaning
Version	Protocol version
Flags	Message control flags
Control Code	Code identifying the query or response type
QTF	Querier timestamp format

	(see Section 3.4 of [RFC6374])
RTF	Responder timestamp format (see Section 3.4 of [RFC6374])
RPTF	Responder's preferred timestamp format
Reserved	Reserved for future specification
Session Identifier	Set arbitrarily by the querier
Differentiated Services (DS) Field	Differentiated Services Code Point (DSCP) being measured
Timestamp 1-4	64-bit timestamp values (see Section 3.4 of [RFC6374])
TLV Block	Optional block of Type-Length-Value fields

4.2. Timestamps

The Section 3.4 of [RFC6374] defines timestamp format that can be used for delay measurement. The IEEE 1588 Precision Time Protocol (PTP) timestamp format [IEEE1588] is used by default as described in Appendix A of [RFC6374], but it may require hardware support. As an alternative, Network Time Protocol (NTP) timestamp format can also be used [RFC6374].

Note that for one-way delay measurement, clock synchronization between the querier and responder nodes using the methods detailed in [RFC6374] is required. The two-way delay measurement does not require clock synchronization between the querier and responder nodes.

5. Performance Loss Measurement

The LM protocol can perform two distinct kinds of loss measurement as described in Section 2.9.8 of [RFC6374].

- In inferred mode, LM will measure the loss of specially generated test messages in order to infer the approximate data plane loss level. Inferred mode LM provides only approximate loss accounting.

- In direct mode, LM will directly measure data plane packet loss. Direct mode LM provides perfect loss accounting, but may require

hardware support.

For both of these modes of LM, path segment identifier [I-D.spring-mpls-path-segment] [I-D.pce-sr-path-segment] is required for accounting received traffic on the egress node of the SR-MPLS Policy.

5.1. Loss Measurement Message Format

As defined in [RFC6374], MPLS LM probe query and response messages use Associated Channel Header (ACH) (value 0x000A for direct loss measurement or value 0x000B for inferred loss measurement), which identifies the message type, and the message payload following the ACH. For both SR links and end-to-end measurement for SR Policies, the same MPLS LM ACH value is used.

The LM message payload as defined in [RFC6374] is used for SR-MPLS loss measurement, for both SR links and end-to-end SR Policies. The LM message payload format is defined as following in [RFC6374]:

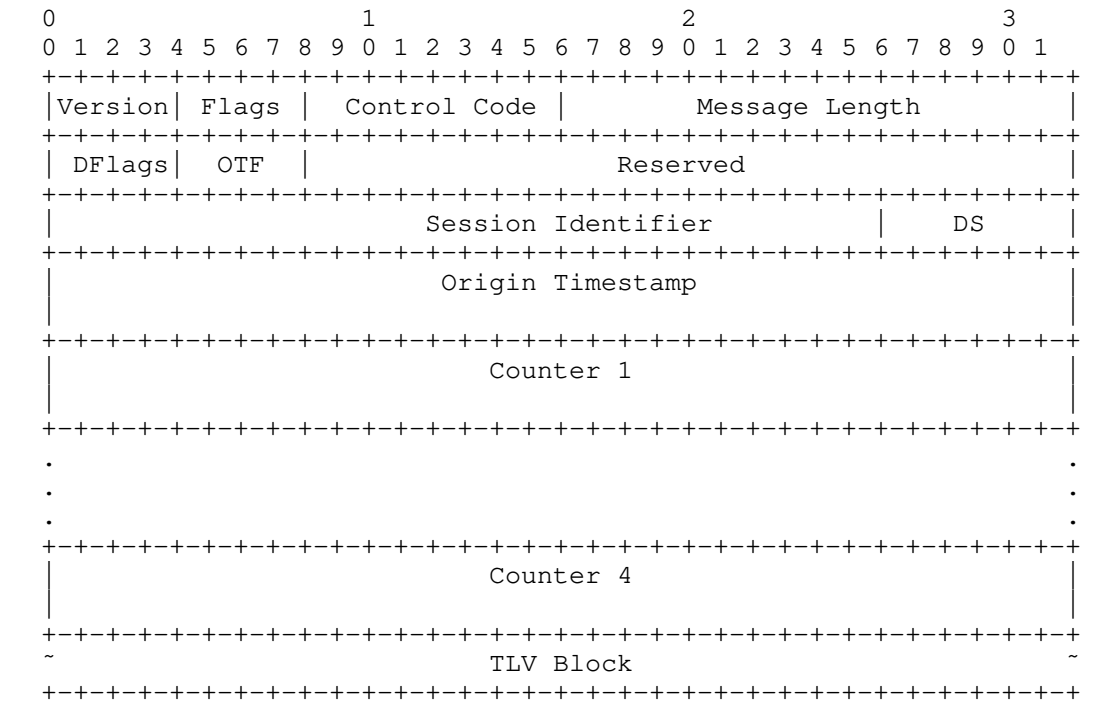


Figure 5: Loss Measurement Message Payload Format

The meanings of the fields are summarized in the following table, see

[RFC6374] for details.

Field	Meaning
Version	Protocol version
Flags	Message control flags
Control Code	Code identifying the query or response type
Message Length	Total length of this message in bytes
Data Format Flags (DFlags)	Flags specifying the format of message data
Origin Timestamp Format (OTF)	Format of the Origin Timestamp field
Reserved	Reserved for future specification
Session Identifier	Set arbitrarily by the querier
Differentiated Services (DS) Field	Differentiated Services Code Point (DSCP) being measured
Origin Timestamp	64-bit field for query message transmission timestamp
Counter 1-4	64-bit fields for LM counter values
TLV Block	Optional block of Type-Length-Value fields

6. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

The responder node may add the "Source Address" TLV (Type 130) [RFC6374] in the probe response message. This TLV allows the querier node to identify the responder node for the SR Policy.

7. SR Link Extended TE Metrics Advertisements

The extended TE metrics for SR link delay and loss computed using the performance measurement procedures reviewed in this document can be

advertised in the routing domain as follows:

- For OSPF, ISIS, and BGP-LS, protocol extensions defined in [RFC7471], [RFC7810] [I-D.lsr-isis-rfc7810bis], and [I-D.idr-te-pm-bgp] are used, respectively for advertising the extended TE link metrics in the network.

- The extended TE link delay metrics advertised are minimum-delay, maximum-delay, average-delay, and delay-variance for one-way.

- The delay-variance metric is computed as specified in Section 4.2 of [RFC5481].

- The one-way delay metrics can be computed using two-way measurement by dividing the measured delay values by 2.

- The extended TE link loss metric advertised is one-way percentage packet loss.

8. Security Considerations

This document reviews the procedures for performance delay and loss measurement for SR-MPLS networks, for both links and end-to-end SR Policies using the mechanisms defined in [RFC6374]. This document does not introduce any additional security considerations other than those covered in [RFC6374], [RFC7471], [RFC7810], and [RFC7876].

9. IANA Considerations

This document does not require any IANA actions.

10. References

10.1. Normative References

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks', RFC 6374, September 2011.

[RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.

10.2. Informative References

[IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock

Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.

[RFC7679] Almes, G., et al., "A One-Way Delay Metric for IP Performance Metrics (IPPM)', RFC 7679, January 2016.

[RFC7471] Giacalone, S., et al., "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.

[RFC7810] Previdi, S., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, May 2016.

[I-D.lsr-isis-rfc7810bis] Ginsberg, L., et al., "IS-IS Traffic Engineering (TE) Metric Extensions", draft-ietf-lsr-isis-rfc7810bis, work in progress.

[I-D.idr-te-pm-bgp] Ginsberg, L. Ed., et al., "BGP-LS Advertisement of IGP Traffic Engineering Performance Metric Extensions", draft-ietf-idr-te-pm-bgp, work in progress.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.

[I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.

[I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment, work in progress.

[I-D.pce-sr-path-segment] Li, C., et al., "Path Computation Element Communication Protocol (PCEP) Extension for Path Identification in Segment Routing (SR)", draft-li-pce-sr-path-segment, work in progress.

Acknowledgments

To be added.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Daniel Bernier
Bell Canada
Email: daniel.bernier@bell.ca

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

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

SPRING Working Group
Internet-Draft
Intended Status: Standards Track
Expires: March 18, 2019

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
S. Salsano
Universita di Roma "Tor Vergata"
P. L. Ventre
CNIT
M. Chen
Huawei

September 14, 2018

UDP Path for In-band Performance Measurement for Segment Routing Networks

draft-gandhi-spring-udp-pm-02

Abstract

Segment Routing (SR) is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document specifies procedures for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement. The procedure uses the RFC 6374 defined mechanisms for Delay and Loss performance measurement. The procedure specified is applicable to SR-MPLS and SRv6 data planes for both links and end-to-end measurement for SR Policies. This document also defines mechanisms for handling Equal Cost Multipaths (ECMPs) for SR Policies. In addition, this document defines Return Path Segment List TLV for two-way performance measurement and Block Number TLV for loss measurement.

Status of This Memo

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

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

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

Copyright Notice

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

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

Table of Contents

1. Introduction

Segment Routing (SR) technology 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, transit and destination nodes. SR Policies as defined in [I-D.spring-segment-routing-policy] are used to steer traffic through a specific, user-defined path 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. These protocols rely on control channel signaling to establish a test channel over an UDP path. These protocols lack support for IEEE 1588 timestamp [IEEE1588] format and direct-mode Loss Measurement (LM), which are required in SR networks [RFC6374]. The Simple Two-way Active Measurement Protocol (STAMP) [I-D.ippm-stamp] alleviates the control channel signaling by using configuration data model to provision test channels. In addition, the STAMP supports IEEE 1588 timestamp format for Delay Measurement (DM). The TWAMP Light from broadband forum [BBF.TR-390] provides simplified mechanisms for active performance measurement in Customer Edge IP networks.

[RFC6374] specifies protocol mechanisms to enable the efficient and accurate measurement of performance metrics and can be used in SR networks with MPLS data plane [I-D.spring-sr-mpls-pm]. [RFC6374] addresses the limitations of the IP based performance measurement protocols as specified in Section 1 of [RFC6374]. The [RFC6374] requires data plane to support MPLS Generic Associated Channel Label (GAL) and Generic Associated Channel (G-Ach), which may not be supported on all nodes in the network.

[RFC7876] specifies the procedures to be used when sending and processing out-of-band performance measurement probe response messages over an UDP return path for RFC 6374 based probe queries.

[RFC7876] can be used to send out-of-band PM probe responses in both SR-MPLS and SRv6 networks for one-way performance measurement.

For SR Policies, there are ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Existing PM protocols (e.g. RFC 6374) do not define handling for ECMP forwarding paths in SR networks.

For two-way measurements for SR Policies, there is a need to specify a return path in the form of a Segment List in PM probe query messages without requiring any SR Policy state on the destination node. Exiting protocols do not have such mechanisms to specify return path in the PM probe query messages.

This document specifies a procedure for using UDP path for sending and processing in-band probe query and response messages for Performance Measurement that does not require to bootstrap PM sessions. The procedure uses RFC 6374 defined mechanisms for Delay and Loss PM and unless otherwise specified, the procedures from RFC 6374 are not modified. The procedure specified is applicable to both SR-MPLS and SRv6 data planes. The procedure does not require to bootstrap PM sessions and can be used for both SR links and end-to-end performance measurement for SR Policies. This document also defines mechanisms for handling Equal Cost Multipaths (ECMPs) for SR Policies. In addition, this document defines Return Path Segment List (RPSL) TLV for two-way performance measurement and Block Number TLV for loss measurement.

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

ACH: Associated Channel Header.

BSID: Binding Segment ID.

DFLag: Data Format Flag.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

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

GAL: Generic Associated Channel (G-ACh) Label.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PTP: Precision Time Protocol.

RPSL: Return Path Segment List.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

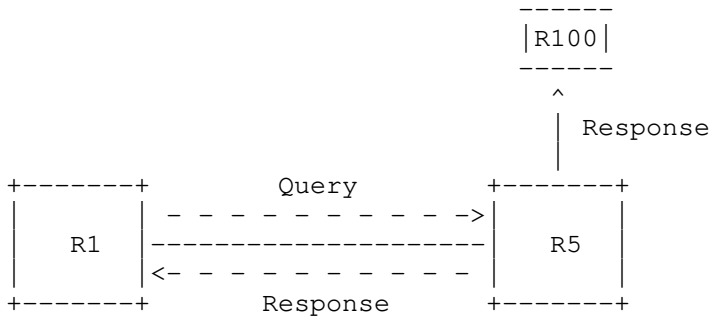
TWAMP: Two-Way Active Measurement Protocol.

URO: UDP Return Object.

2.3. Reference Topology

In the reference topology, the querier node R1 initiates a probe query for performance measurement and the responder node R5 sends a probe response for the query message received. The probe response may be sent to the querier node R1 or to a controller node R100. The nodes R1 and R5 may be directly connected via a link enabled with Segment Routing or there exists a Point-to-Point (P2P) SR Policy [I-D.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.spring-sr-p2mp-policy].



Reference Topology

Both Delay and Loss performance measurement is performed in-band for the traffic traversing between node R1 and node R5. One-way delay and two-way delay measurements are defined in Section 2.4 of [RFC6374]. Transmit and Receive packet loss measurements are defined in Section 2.2 and Section 2.6 of [RFC6374]. One-way loss measurement provides receive packet loss whereas two-way loss measurement provides both transmit and receive packet loss.

3. Probe Messages

3.1. Probe Query Message

In this document, UDP path is defined for sending and processing PM probe query messages for Delay and Loss measurements for SR links and end-to-end SR Policies as described in the following Sections. As well-known UDP port is used for identifying PM probe packets, bootstrapping of the PM session [RFC5357] is not required. The TTL / Hop Limit field of the IP header MUST be set to 1.

3.1.1. Delay Measurement Probe Query Message

The message content for Delay Measurement probe query message using UDP header [RFC768] is shown in Figure 1. As shown, the DM probe query message is sent with Destination UDP port number TBA1 defined in this document. The Source UDP port may optionally be set to TBA1 for two-way delay measurement. The DM probe query message contains the payload for delay measurement defined in Section 3.2 of [RFC6374].

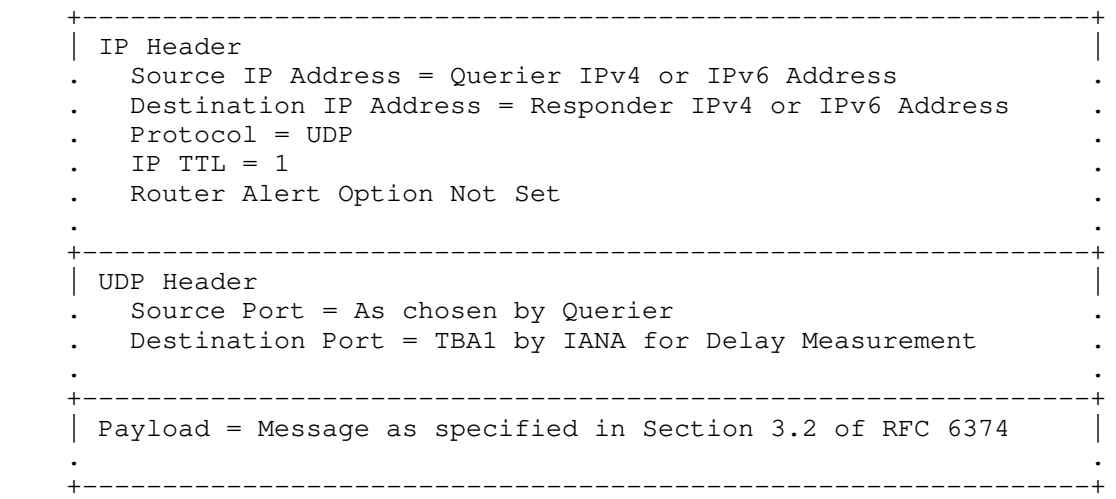


Figure 1: DM Probe Query Message

3.1.2. Loss Measurement Probe Query Message

The message content for Loss measurement probe query message using UDP header [RFC768] is shown in Figure 2. As shown, the LM probe query message is sent with Destination UDP port number TBA2 defined in this document. The Source UDP port may optionally be set to TBA2 for two-way loss measurement. The LM probe query message contains the payload for loss measurement defined in Section 3.1 of [RFC6374].

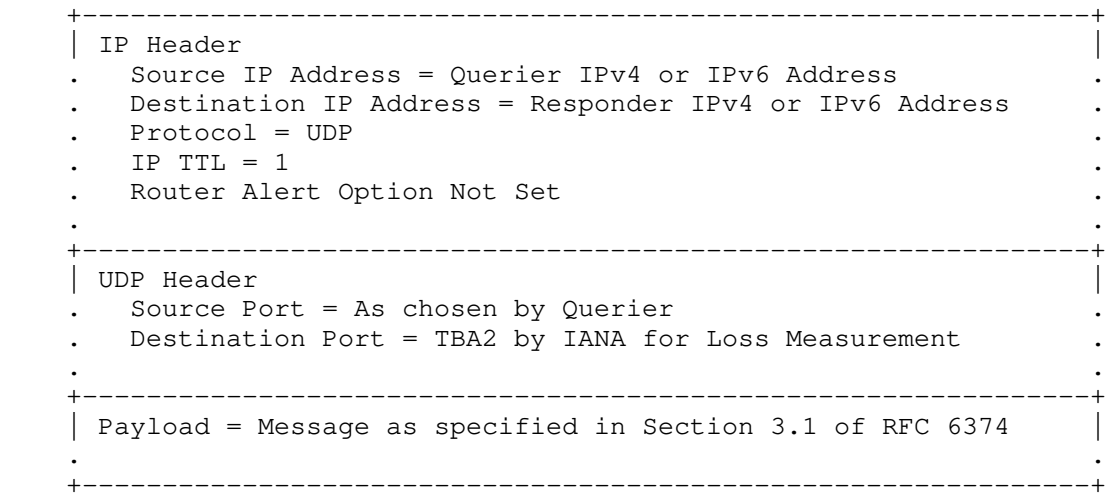


Figure 2: LM Probe Query Message

The path segment identifier [I-D.spring-mpls-path-segment] [I-D.pce-sr-path-segment] of the SR Policy is required for accounting received traffic on the egress node for loss measurement.

3.1.2.1. Block Number TLV

The Loss Measurement using Alternate-Marking method defined in [RFC8321] requires to identify the Block Number (color) of the traffic counters carried by the probe query and response messages. Probe query and response messages specified in [RFC6374] for Loss Measurement do not define any means to carry the Block Number.

[RFC6374] defines probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA8) is defined in this document to carry Block Number (32-bit) for the traffic counters in the probe query and response messages for loss measurement. The format of the Block Number TLV is shown in Figure 11:

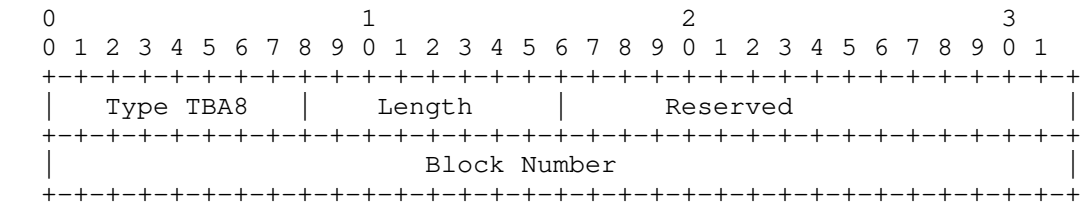


Figure 11: Block Number TLV

The Block Number TLV is optional. The PM querier node SHOULD only insert one Block Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Block Number TLV from the probe query messages and ignore other Block Number TLVs if present. In both probe query and response messages, the counters MUST belong to the same Block Number.

3.1.3. In-band Probe Query for SR Links

The probe query message as defined in Figure 1 is sent in-band for Delay measurement. The probe query message as defined in Figure 2 is sent in-band for Loss measurement.

3.1.4. In-band Probe Query for End-to-end Measurement of SR Policy

3.1.4.1. In-band Probe Query Message for SR-MPLS Policy

The message content for in-band probe query message using UDP header

for end-to-end performance measurement of SR-MPLS Policy is shown in Figure 3.

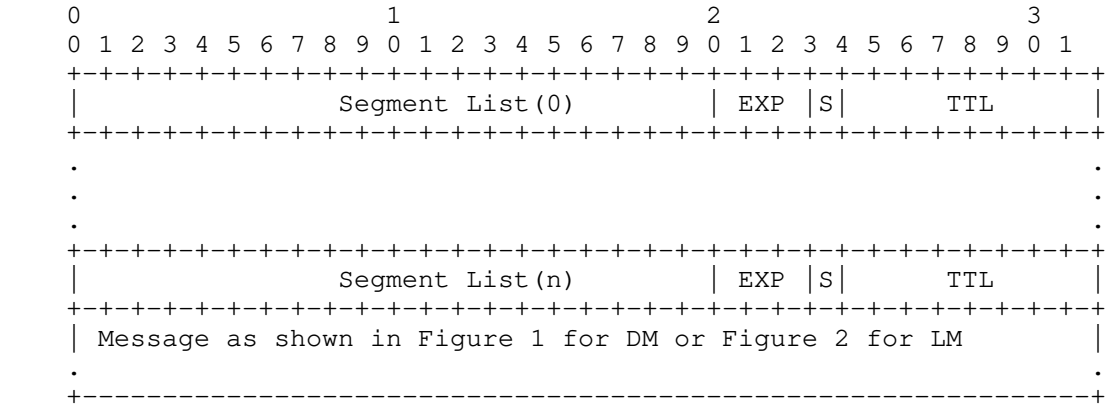


Figure 3: In-band Probe Query Message for SR-MPLS Policy

The Segment List (SL) can be empty to indicate Implicit NULL label case.

3.1.4.2. In-band Probe Query Message for SRv6 Policy

The in-band probe query messages using UDP header for end-to-end performance measurement of an SRv6 Policy is sent using SRv6 Segment Routing Header (SRH) and Segment List of the SRv6 Policy as defined in [I-D.6man-segment-routing-header] and is shown in Figure 4.

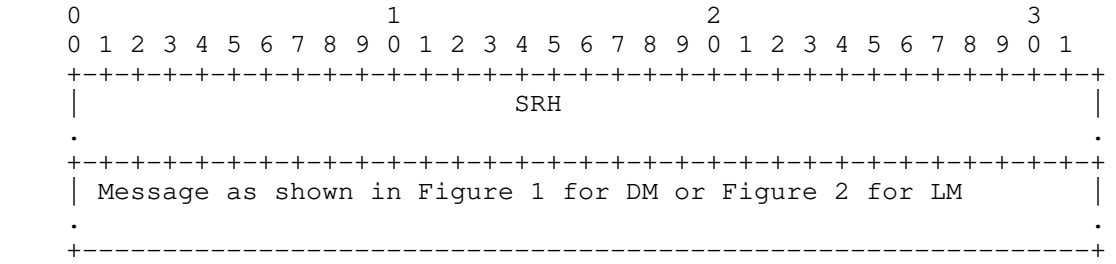


Figure 4: In-band Probe Query Message for SRv6 Policy

3.2. Probe Response Message

When the received probe query message does not contain any UDP Return Object (URO) TLV [RFC7876], the probe response message is sent using the IP/UDP information from the probe query message. The content of

the probe response message is shown in Figure 5.

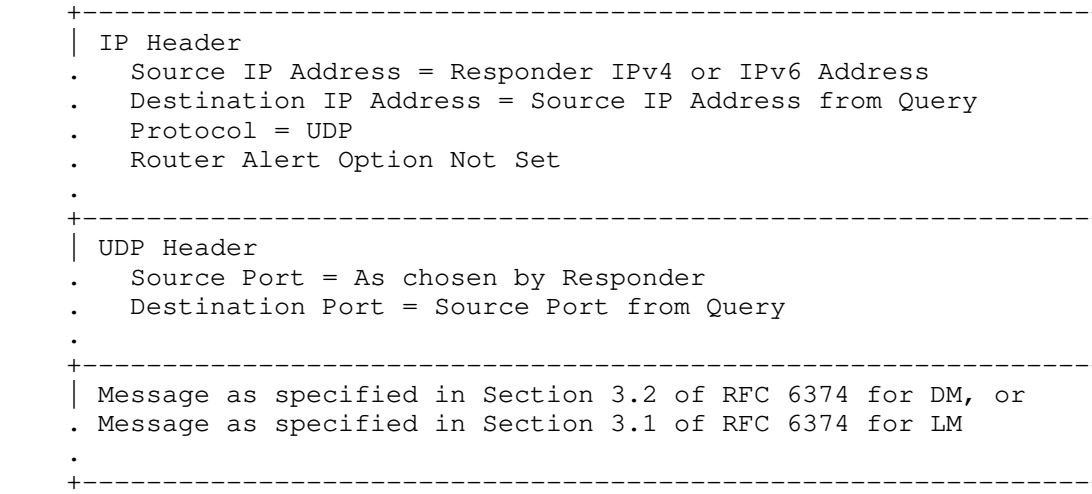


Figure 5: Probe Response Message

When the received probe query message contains UDP Return Object (URO) TLV [RFC7876], the probe response message the message uses the IP/UDP information from the URO in the probe query message. The content of the probe response message is shown in Figure 6.

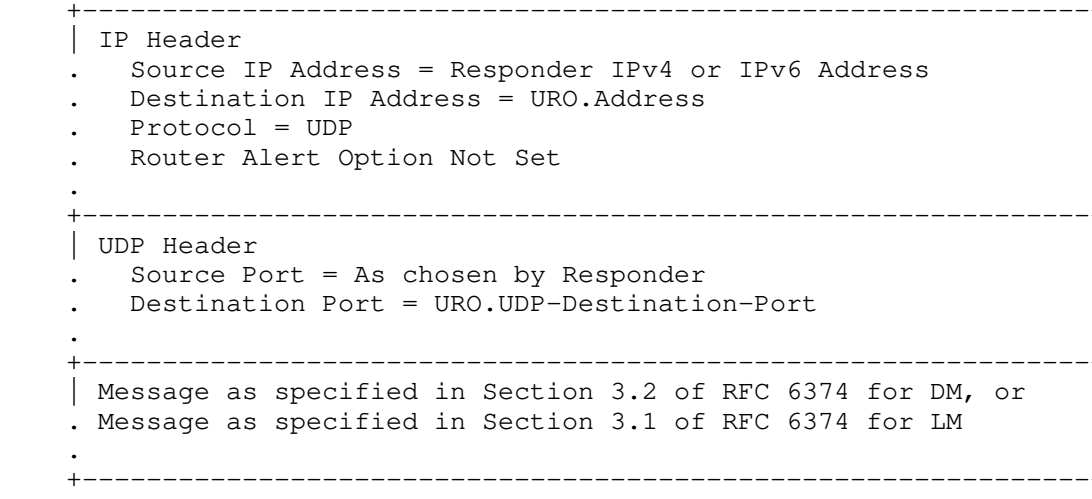


Figure 6: Probe Response Message Using URO from Probe Query Message

3.2.1. One-way Measurement for SR Link and end-to-end SR Policy

For one-way performance measurement, the probe response message as defined in Figure 5 or Figure 6 is sent out-of-band for both SR links and SR Policies.

The PM querier node can receive probe response message back by properly setting its own IP address as Source Address of the header or by adding URO TLV in the probe query message and setting its own IP address in the IP Address in the URO TLV (Type=131) [RFC7876]. In addition, the "control code" in the probe query message is set to "out-of-band response requested". The "Source Address" TLV (Type 130), and "Return Address" TLV (Type 1), if present in the probe query message, are not used to send probe response message.

3.2.1.1. Probe Response Message to Controller

As shown in the Reference Topology, if the querier node requires the probe response message to be sent to the controller R100, it adds URO TLV in the probe query message and sets the IP address of R100 in the IP Address field and UDP port TBA1 for DM and TBA2 for LM in the UDP-Destination-Port field of the URO TLV (Type=131) [RFC7876].

3.2.2. Two-way Measurement for SR Links

For two-way performance measurement, when using a bidirectional channel, the probe response message as defined in Figure 5 or Figure 6 is sent back in-band to the querier node for SR links. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374].

3.2.3. Two-way End-to-end Measurement of SR Policy

For two-way performance measurement, when using a bidirectional channel, the probe response message is sent back in-band to the querier node for end-to-end measurement of SR Policies. In this case, the "control code" in the probe query message is set to "in-band response requested" [RFC6374].

The path segment identifier [I-D.spring-mpls-path-segment] [I-D.pce-sr-path-segment] of the forward SR Policy can be used to find the reverse SR Policy to send the probe response message in the absence of RPSL TLV defined in the following Section.

3.2.3.1. Return Path Segment List TLV

For two-way performance measurement, the responder node needs to send the probe response message in-band on a specific reverse SR path. This way the destination node does not require any additional SR Policy state. The querier node can request in the probe query

message to the responder node to send a response back on a given reverse path (typically co-routed path for two-way measurement).

[RFC6374] defines DM and LM probe query messages that can include one or more optional TLVs. New TLV Types are defined in this document for Return Path Segment List (RPSL) to carry reverse SR path for probe response messages. The format of the RPSL TLV is shown in Figure 7:

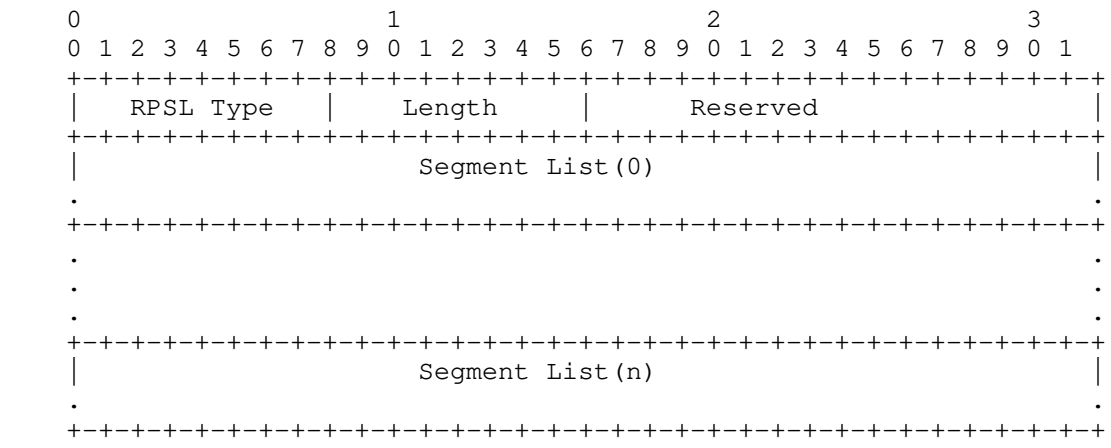


Figure 7: Return Path Segment List TLV

The RPSL can be one of following Types:

- RPSL Type (value TBA3): SR-MPLS Label Stack of the Reverse SR Policy
- RPSL Type (value TBA4): SRv6 Segment List of the Reverse SR Policy
- RPSL Type (value TBA5): SR-MPLS Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy
- RPSL Type (value TBA6): SRv6 Binding SID [I-D.pce-binding-label-sid] of the Reverse SR Policy

The Segment List(0) can be used by the responder node to compute the next-hop IP address and outgoing interface to send the probe response messages.

The RPSL TLV is optional. The PM querier node MUST only insert one RPSL TLV in the probe query message and the responder node MUST only process the first RPSL TLV in the probe query message and ignore

other RPSL TLVs if present. The responder node MUST send probe response message back on the reverse path specified in the RPSL TLV and MUST NOT add RPSL TLV in the probe response message.

3.2.3.2. In-band Probe Response Message for SR-MPLS Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SR-MPLS Policy is shown in Figure 8. The SR-MPLS label stack in the packet header is built using the Segment List received in the RPSL TLV in the probe query message.

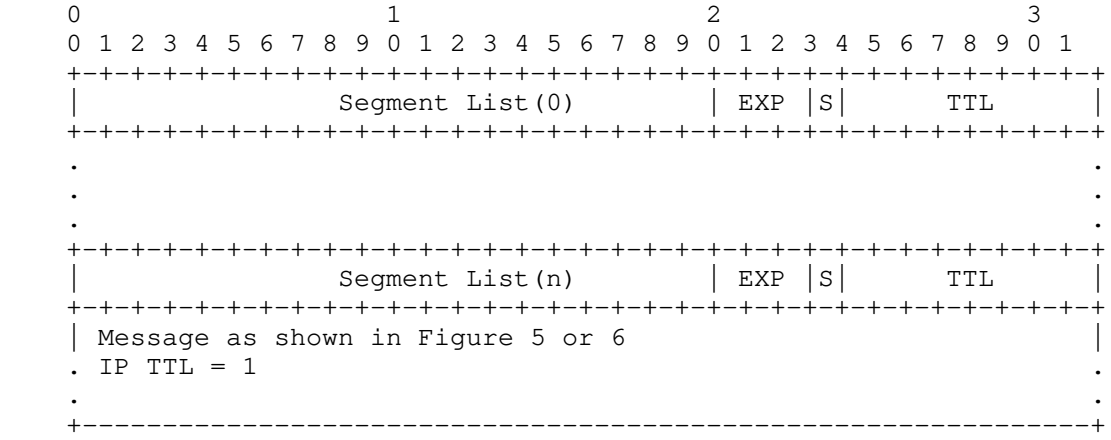


Figure 8: In-band Probe Response Message for SR-MPLS Policy

3.2.3.3. In-band Probe Response Message for SRv6 Policy

The message content for sending probe response message in-band using UDP header for two-way end-to-end performance measurement of an SRv6 Policy is shown in Figure 9. For SRv6 Policy, the SRv6 SID list in the SRH of the probe response message is built using the SRv6 Segment List received in the RPSL TLV in the probe query message.

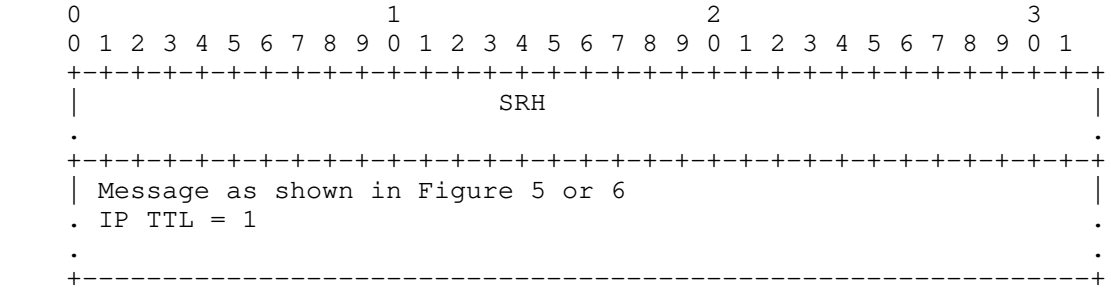


Figure 9: In-band Probe Response Message for SRv6 Policy

4. Performance Measurement for P2MP SR Policies

The procedures for delay and loss measurement described in this document for Point-to-Point (P2P) SR-MPLS Policies are also equally applicable to the Point-to-Multipoint (P2MP) SR Policies.

5. ECMP Support

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 can be sent to traverse different ECMP paths to measure performance of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. Following mechanisms can be used in PM probe messages to take advantage of the hashing function in forwarding plane to influence the path taken by them.

- The mechanisms described in [RFC8029] [RFC5884] for handling ECMPs are also applicable to the performance measurement. In the IP/UDP header of the PM probe messages, Destination Addresses in 127/8 range for IPv4 or 0:0:0:0:0:FFFF:7F00/104 range for IPv6 can be used to exercise a particular ECMP path. In addition, different Source Addresses or different Source UDP ports can be used for this purpose. As specified in [RFC6437], 3-tuple of Flow Label, Source Address and Destination Address fields in the IPv6 header can also be used.

- For SR-MPLS, entropy label [RFC6790] in the PM probe messages can be used.

- For SRv6, Flow Label in SRH [I-D.6man-segment-routing-header] of the PM probe messages can be used.

6. Sequence Number TLV

The message formats for DM and LM [RFC6374] do not contain sequence number for probe query packets. Sequence numbers can be useful when some probe query messages are lost or they arrive out of order.

[RFC6374] defines DM and LM probe query and response messages that can include one or more optional TLVs. New TLV Type (value TBA7) is defined in this document to carry sequence number for probe query and response messages for delay and loss measurement. The format of the

Sequence Number TLV is shown in Figure 10:

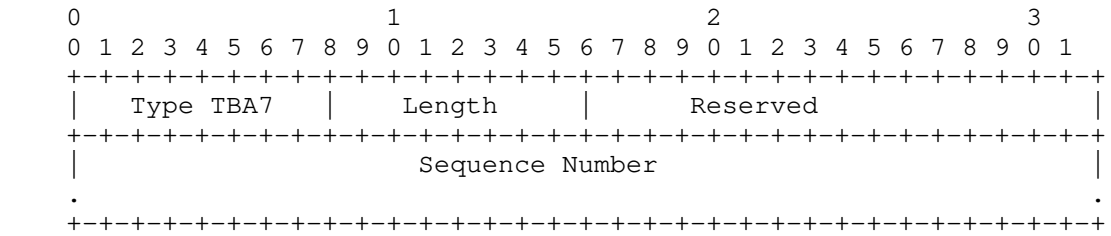


Figure 10: Sequence Number TLV

The sequence numbers start with 0 and are incremented by one for each subsequent probe query packet. The sequence number can be of any length determined by the querier node. The Sequence Number TLV is optional. The PM querier node SHOULD only insert one Sequence Number TLV in the probe query message and the responder node in the probe response message SHOULD return the first Sequence Number TLV from the probe query message and ignore other Sequence Number TLVs if present.

7. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. The security considerations described in Section 8 of [RFC6374] are applicable to this specification, and particular attention should be paid to the last two paragraphs. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

8. IANA Considerations

IANA is requested to allocate following UDP ports for performance measurements:

- UDP Port TBA1: Delay Performance Measurement
- UDP Port TBA2: Loss Performance Measurement

IANA is also requested to allocate values for the following Return Path Segment List TLV Types for RFC 6374 to be carried in PM probe query messages:

- Type TBA3: SR-MPLS Label Stack of the Reverse SR Policy

- Type TBA4: SRv6 Segment List of the Reverse SR Policy
- Type TBA5: SR-MPLS Binding SID of the Reverse SR Policy
- Type TBA6: SRv6 Binding SID of the Reverse SR Policy

IANA is also requested to allocate a value for the following Sequence Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for delay and loss measurement:

- Type TBA7: Sequence Number TLV

IANA is also requested to allocate a value for the following Block Number TLV Type for RFC 6374 to be carried in the PM probe query and response messages for loss measurement:

- Type TBA8: Block Number TLV

9. References

9.1. Normative References

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

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

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS networks', RFC 6374, September 2011.

[RFC7876] Bryant, S., Sivabalan, S., and Soni, S., "UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks", RFC 7876, July 2016.

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

9.2. Informative References

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

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, September 2006.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, October 2008.

[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.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Kumar, N., Aldrin, S. and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, March 2017.

[RFC8321] Fioccola, G. Ed., "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, January 2018.

[I-D.spring-segment-routing-policy] Filsfils, C., et al., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy, work in progress.

[I-D.spring-sr-p2mp-policy] Voyer, D. Ed., et al., "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy, work in progress.

[I-D.6man-segment-routing-header] Filsfils, C., et al., "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header, work in progress.

[I-D.spring-sr-mpls-pm] Filsfils, C., Gandhi, R. Ed., et al. "Performance Measurement in Segment Routing Networks with MPLS Data Plane", draft-gandhi-spring-sr-mpls-pm, work in progress.

[I-D.pce-binding-label-sid] Filsfils, C., et al., "Carrying Binding

Label Segment-ID in PCE-based Networks",
draft-sivabalan-pce-binding-label-sid, work in progress.

[I-D.spring-mpls-path-segment] Cheng, W., et al., "Path Segment in MPLS Based Segment Routing Network", draft-cheng-spring-mpls-path-segment, work in progress.

[I-D.pce-sr-path-segment] Li, C., et al., "Path Computation Element Communication Protocol (PCEP) Extension for Path Identification in Segment Routing (SR)", draft-li-pce-sr-path-segment, work in progress.

[I-D.ippm-stamp] Mirsky, G. et al. "Simple Two-way Active Measurement Protocol", draft-ietf-ippm-stamp, work in progress.

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

Acknowledgments

The authors would like to thank Nagendra Kumar and Carlos Pignataro for the discussion on SRv6 Performance Measurement.

Contributors

Sagar Soni
Cisco Systems, Inc.
Email: sagsoni@cisco.com

Patrick Khordoc
Cisco Systems, Inc.
Email: pkhordoc@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Daniel Bernier
Bell Canada
Email: daniel.bernier@bell.ca

Dirk Steinberg
Steinberg Consulting
Germany
Email: dws@dirksteinberg.de

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

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy
Email: stefano.salsano@uniroma2.it

Pier Luigi Ventre
CNIT
Italy
Email: pierluigi.ventre@cnit.it

Mach(Guoyi) Chen
Huawei
Email: mach.chen@huawei.com

SPRING
Internet-Draft
Intended status: Standards Track
Expires: March 31, 2019

J. Guichard, Ed.
H. Song
Huawei
J. Tantsura
Nuage Networks
J. Halpern
Ericsson
W. Henderickx
Nokia
M. Boucadair
Orange

September 27, 2018

NSH and Segment Routing Integration for Service Function Chaining (SFC) draft-guichard-spring-nsh-sr-00

Abstract

This document describes two application scenarios where Network Service Header (NSH) and Segment Routing (SR) techniques can be deployed together to support Service Function Chaining (SFC) in an efficient manner while maintaining separation of the service and transport planes as originally intended by the SFC architecture.

In the first scenario, an NSH-based SFC is created using SR as the transport between SFFs. SR in this case is just one of many encapsulations that could be used to maintain the transport-independent nature of NSH-based service chains.

In the second scenario, SR is used to represent each service hop of the NSH-based SFC as a segment within the segment-list. SR and NSH in this case are integrated.

In both scenarios SR is responsible for steering packets between SFFs along a given SFP while NSH is responsible for maintaining the integrity of the service plane, the SFC instance context, and any associated metadata.

These application scenarios demonstrate that NSH and SR can work jointly and complement each other leaving the network operator with the flexibility to use whichever transport technology makes sense in specific areas of their network infrastructure, and still maintain an end-to-end service plane using NSH.

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 31, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

1.1. SFC Overview and Rationale

The dynamic enforcement of a service-derived, adequate forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators and service providers. Particularly, cascading SFs, for example at the Gi interface in the context of mobile network infrastructure, have shown their limits, such as the same redundant classification features must be supported by many SFs in order to execute their function, some SFs are receiving traffic that they are not supposed to process (e.g., TCP proxies receiving UDP traffic), which inevitably affects their dimensioning and performance, an increased design complexity related to the properly ordered invocation of several SFs, etc.

In order to solve those problems and to avoid the adherence with the underlying physical network topology while allowing for simplified service delivery, Service Function Chaining (SFC) techniques have been introduced.

SFC techniques are meant to rationalize the service delivery logic and master the companion complexity while optimizing service activation time cycles for operators that need more agile service delivery procedures to better accommodate ever-demanding customer requirements. Indeed, SFC allows to dynamically create service planes that can be used by specific traffic flows. Each service plane is realized by invoking and chaining the relevant service functions in the right sequence. [RFC7498] provides an overview of the SFC problem space and [RFC7665] specifies an SFC architecture. The SFC architecture has the merit to not make assumptions on how advanced features (e.g., load-balancing, loose or strict service paths) have to be enabled with a domain. Various deployment options are made available to operators with the SFC architecture and this approach is fundamental to accommodate various and heterogeneous deployment contexts.

Many approaches can be considered for encoding the information required for SFC purposes (e.g., communicate a service chain pointer, encode a list of loose/explicit paths, disseminate a service chain identifier together with a set of context information, etc.). Likewise, many approaches can also be considered for the channel to be used to carry SFC-specific information (e.g., define a new header, re-use existing fields, define an IPv6 extension header, etc.).

Among all these approaches, the IETF endorsed a transport-independent SFC encapsulation scheme: NSH [RFC8300]; which is the most mature SFC encapsulation solution. This design is pragmatic as it does not require replicating the same specification effort as a function of underlying transport encapsulation. Moreover, this design approach encourages consistent SFC-based service delivery in networks enabling distinct transport protocols in various segments of the network or even between SFFs vs SF-SFF hops.

1.2. SFC within SR Networks

As described in [I-D.ietf-spring-segment-routing], Segment Routing (SR) leverages the source routing technique. Concretely, a node steers a packet through an SR policy instantiated as an ordered list of instructions called segments. While initially designed for policy-based source routing, SR also finds its application in supporting SFC [I-D.xu-clad-spring-sr-service-chaining]. The two SR flavors, namely MPLS-SR [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.ietf-6man-segment-routing-header], can both encode a Service Function (SF) as a segment so that an SFC can be specified as a segment list. Nevertheless, and as discussed in [RFC7498], traffic steering is only a subset of the issues that motivated the design of the SFC architecture. Further considerations such as simplifying classification at intermediate SFs and allowing for coordinated behaviors among SFs by means of supplying context information should be taken into account when designing an SFC data plane solution.

While each scheme (i.e., NSH-based SFC and SR-based SFC) can work independently, this document describes how the two can be used together in concert and complement each other through two representative application scenarios. Both application scenarios may be supported using either MPLS-SR or SRv6:

- NSH-based SFC with SR-based transport plane: in this scenario segment routing provides the transport encapsulation between SFFs while NSH is used to convey and trigger SFC polices.

- SR-based SFC with integrated NSH service plane: in this scenario each service hop of the SFC is represented as a segment of the SR segment-list. SR is responsible for steering traffic through the necessary SFFs as part of the segment routing path and NSH is responsible for maintaining the service plane, and holding the SFC instance context and associated metadata.

It is of course possible to combine both of these two scenarios so as to support specific deployment requirements and use cases.

2. NSH-based SFC with SR-based transport tunnel

Because of the transport-independent nature of NSH-based service chains, it is expected that the NSH has broad applicability across different domains of a network. By way of illustration the various SFs involved in a service chain are available in a single data center, or spread throughout multiple locations (e.g., data centers, different POPs), depending upon the operator preference and/or availability of service resources. Regardless of where the service resources are deployed it is necessary to provide traffic steering through a set of SFFs and NSH-based service chains provide the flexibility for the network operator to choose which particular transport encapsulation to use between SFFs, which may be different depending upon which area of the network the SFFs/SFs are currently deployed. Therefore from an SFC architecture perspective, segment routing is simply one of multiple available transport encapsulations that can be used for traffic steering between SFFs. Concretely, NSH does not require to use a unique transport encapsulation when traversing a service chain. NSH-based service forwarding relies upon underlying service node capabilities.

The following three figures provide an example of an SFC established for flow F that has SF instances located in different data centers, DC1 and DC2. For the purpose of illustration, let the SFC's Service Path Identifier (SPI) be 100 and the initial Service Index (SI) be 255.

Referring to Figure 1, packets of flow F in DC1 are classified into an NSH-based SFC and encapsulated after classification as <Inner Pkt><NSH: SPI 100, SI 255><Outer-transport> and forwarded to SFF1 (which is the first SFF hop for this service chain).

After removing the outer transport encapsulation, that may or may not be MPLS-SR or SRv6, SFF1 uses the SPI and SI carried within the NSH encapsulation to determine that it should forward the packet to SF1. SF1 applies its service, decrements the SI by 1, and returns the packet to SFF1. SFF1 therefore has <SPI 100, SI 254> when the packet comes back from SF1. SFF1 does a lookup on <SPI 100, SI 254> which results in <next-hop: DC1-GW1> and forwards the packet to DC1-GW1.

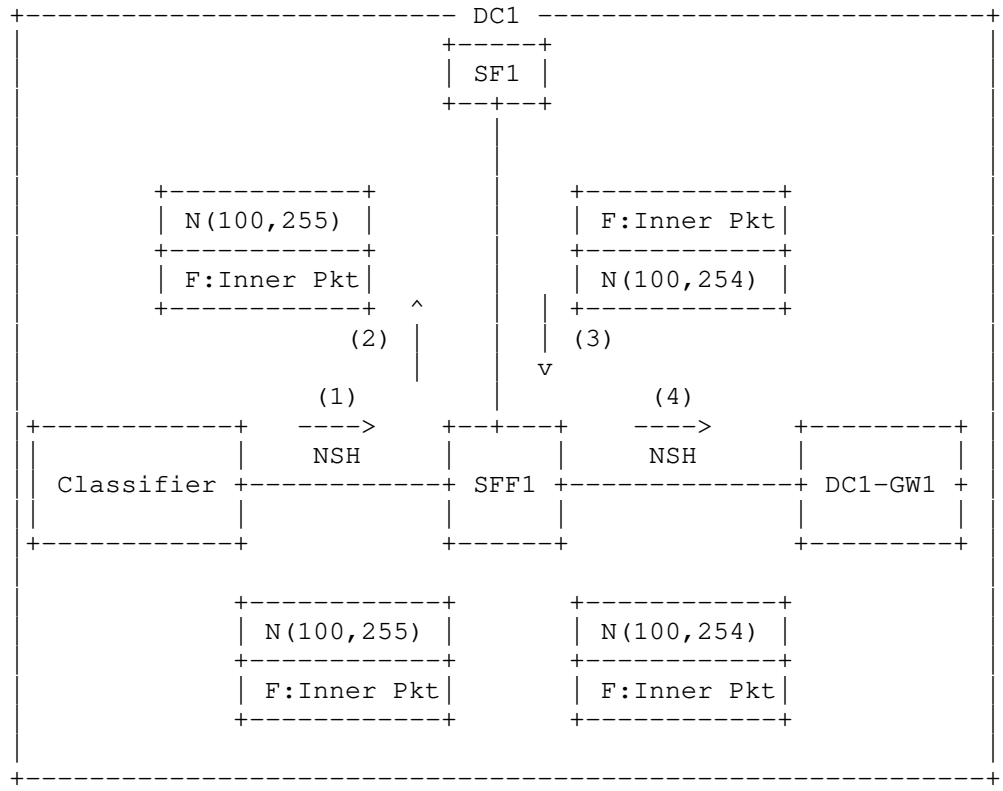


Figure 1: SR for inter-DC SFC - Part 1

Referring now to Figure 2, DC1-GW1 performs a lookup on the information conveyed in the NSH which results in <next-hop: DC2-GW1, encapsulation: SR>. The SR encapsulation has the SR segment-list to forward the packet across the inter-DC network to DC2.

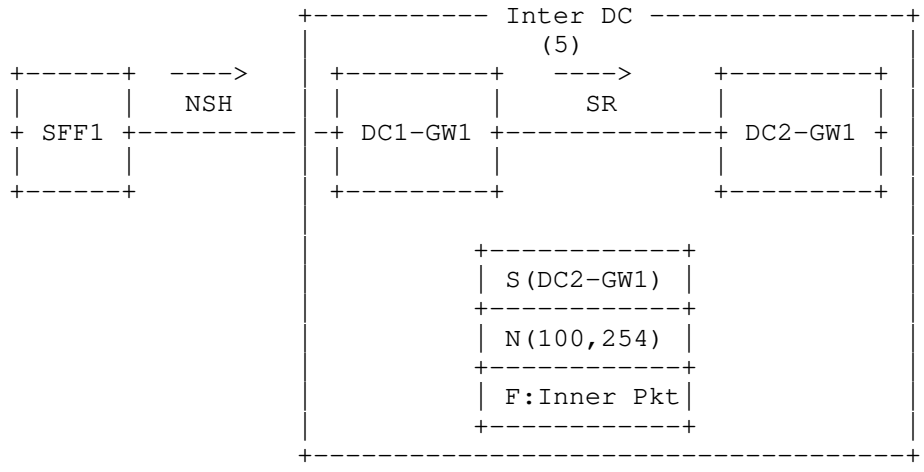


Figure 2: SR for inter-DC SFC - Part 2

When the packet arrives at DC2, as shown in Figure 3, the SR encapsulation is removed and DC2-GW1 performs a lookup on the NSH which results in next-hop: SFF2. The outer transport encapsulation may be any transport that is able to identify NSH as the next protocol.

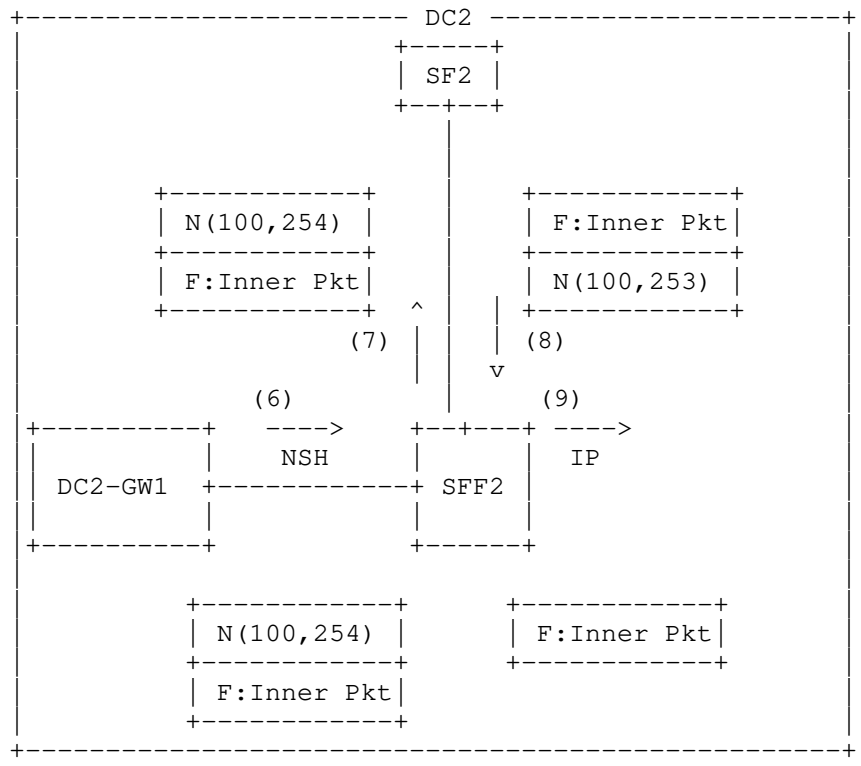


Figure 3: SR for inter-DC SFC - Part 3

The benefits of this scheme are listed hereafter:

- The network operator is able to take advantage of the transport-independent nature of the NSH encapsulation.
- The network operator is able to take advantage of the traffic steering capability of SR where appropriate.
- Light-weight NSH is used in the data center for SFC and avoids more complex hierarchical SFC schemes between data centers.
- Clear responsibility division and scope between NSH and SR.

Note that this scenario is applicable to any case where multiple segments of a service chain are distributed into multiple domains or where traffic-engineered paths are necessary between SFFs (strict forwarding paths for example). Further note that the above example can also be implemented using end to end segment routing between SFF1

and SFF2. (As such DC-GW1 and DC-GW2 are forwarding the packets based on segment routing instructions and are not looking at the NSH header for forwarding).

3. SR-based SFC with Integrated NSH Service Plane

In this scenario we assume that the SFs are NSH-aware and therefore it should not be necessary to implement an SFC proxy to achieve Service Function Chaining. The operation relies upon SR to perform SFF-SFF transport and NSH to provide the service plane between SFs thereby maintaining SFC context and metadata.

When a service chain is established, a packet associated with that chain will first encapsulate an NSH that will be used to maintain the end-to-end service plane through use of the SFC context. The SFC context (e.g., the service plane path referenced by the SPI) is used by an SFF to determine the SR segment list for forwarding the packet to the next-hop SFFs. The packet is then encapsulated using the (transport-specific) SR header and forwarded in the SR domain following normal SR operation.

When a packet has to be forwarded to an SF attached to an SFF, the SFF performs a lookup on the prefix SID associated with the SF to retrieve the next-hop context between the SFF and SF. E.g. to retrieve the destination MAC address in case native ethernet encapsulation is used between SFF and SF. How the next-hop context is populated is out of the scope of this document. The SFF strips the SR information of the packet, updates the SR information, and saves it to a cache indexed by the NSH SPI. This saved SR information is used to encapsulate and forward the packet(s) coming back from the SF.

When the SF receives the packet, it processes it as usual and sends it back to the SFF. Once the SFF receives this packet, it extracts the SR information using the NSH SPI as the index into the cache. The SFF then pushes the SR header on top of the NSH header, and forwards the packet to the next segment in the segment list.

Figure 4 illustrates an example of this scenario.

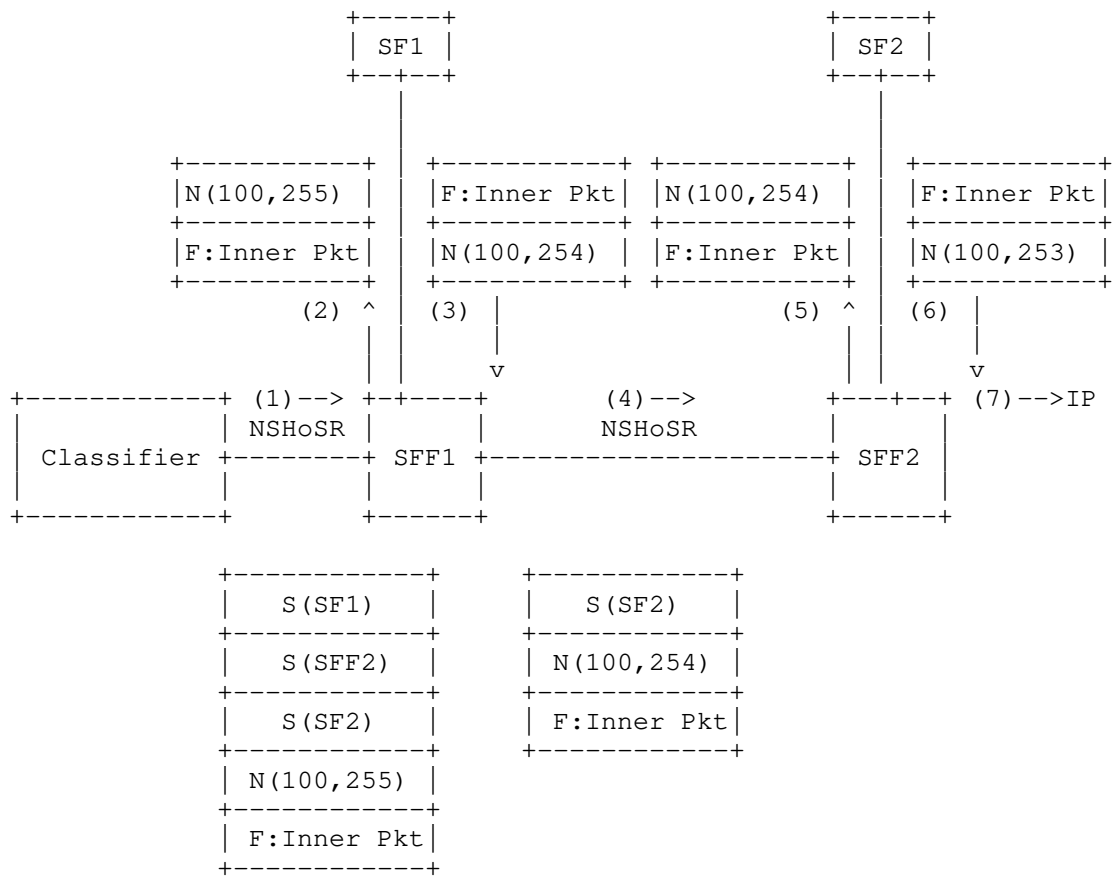


Figure 4: NSH over SR for SFC

The benefits of this scheme include:

- It is economically sound for SF vendors to only support one unified SFC solution. The SF is unaware of the SR.
- It simplifies the SFF (i.e., the SR router) by nullifying the needs for re-classification and SR proxy.
- It provides a unique and standard way to pass metadata to SFs. Note that currently there is no solution for MPLS-SR to carry metadata and there is no solution to pass metadata to SR-unaware SFs.
- SR is also used for forwarding purposes including between SFFs.

- o It takes advantage of SR to eliminate the NSH forwarding state in SFFs. This applies each time strict or loose SFPs are in use.

- o It requires no interworking as would be the case if MPLS-SR based SFC and NSH-based SFC were deployed as independent mechanisms in different parts of the network.

4. Encapsulation Details

4.1. NSH using MPLS-SR Transport

MPLS-SR instantiates Segment IDs (SIDs) as MPLS labels and therefore the segment routing header is a stack of MPLS labels.

When carrying NSH within an MPLS-SR transport, the full encapsulation headers are as illustrated in Figure 5.

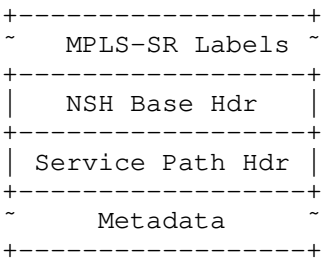


Figure 5: NSH using MPLS-SR Transport

As described in [I-D.ietf-spring-segment-routing] the IGP signaling extension for IGP-Prefix segment includes a flag to indicate whether directly connected neighbors of the node on which the prefix is attached should perform the NEXT operation or the CONTINUE operation when processing the SID. When NSH is carried beneath MPLS-SR it is necessary to terminate the NSH-based SFC at the tail-end node of the MPLS-SR label stack. This is the equivalent of MPLS Ultimate Hop Popping (UHP) and therefore the prefix-SID associated with the tail-end of the SFC MUST be advertised with the CONTINUE operation so that the penultimate hop node does not pop the top label of the MPLS-SR label stack and thereby expose NSH to the wrong SFF. It is RECOMMENDED that a specific prefix-SID be allocated at each node for use by the SFC application for this purpose.

At the end of the MPLS-SR path it is necessary to provide an indication to the tail-end that NSH follows the MPLS-SR label stack.

There are several ways to achieve this but its specification is outside the scope of this document.

4.2. NSH using SRv6 Transport

When carrying NSH within an SRv6 transport the full encapsulation is as illustrated in Figure 6.

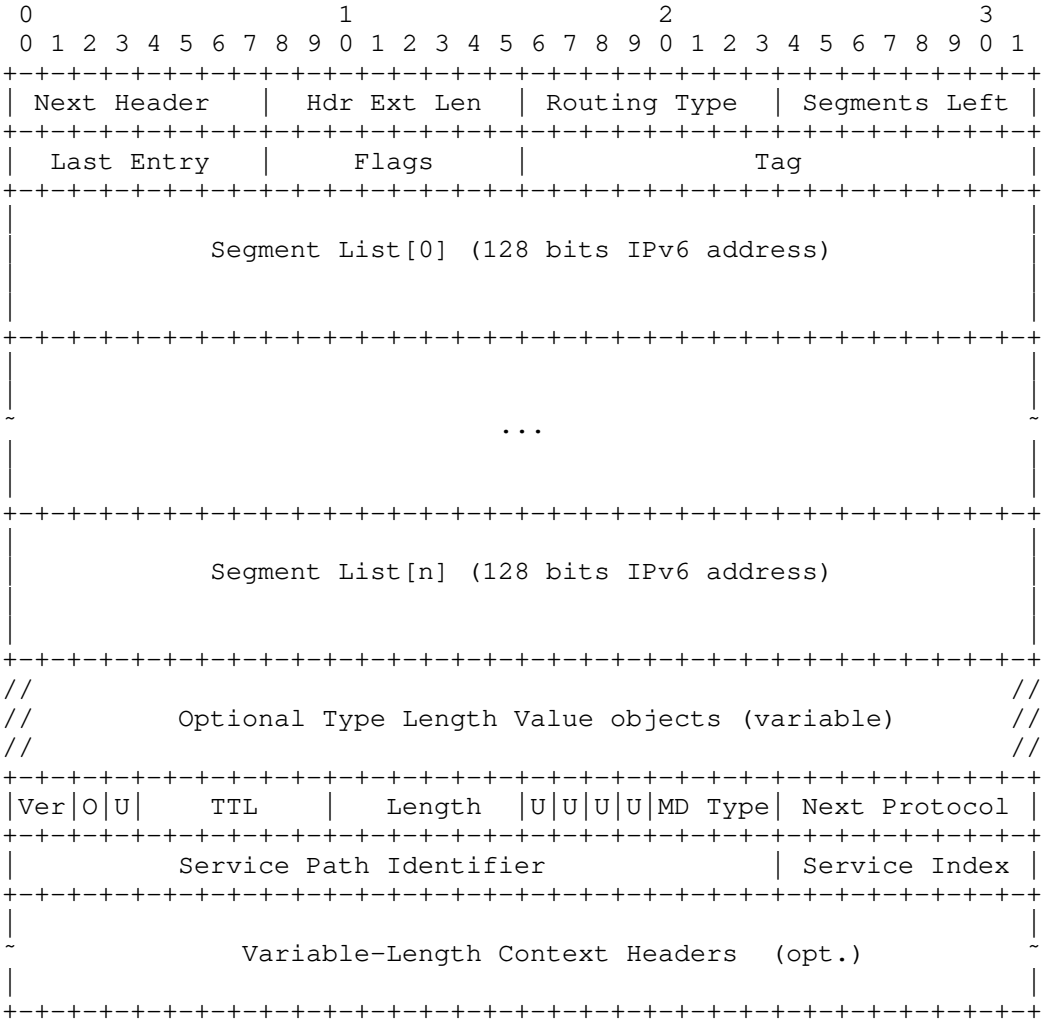


Figure 6: NSH using SRv6 Transport

Encapsulation of NSH following SRv6 may be indicated either by encapsulating NSH in UDP (UDP port TBA1) and indicating UDP in the Next Header field of the SRH, or by indicating an IP protocol number for NSH in the Next Header of the SRH. The behavior for encapsulating NSH over UDP, including the selection of the source port number in particular, adheres to similar considerations as those discussed in [RFC8086].

5. Security Considerations

Generic SFC-related security considerations are discussed in [RFC7665]. NSH-specific security considerations are discussed in [RFC8300]. NSH-in-UDP with DTLS [RFC6347] should follow the considerations discussed in Section 5 of [RFC8086], with a destination port number set to TBA2

6. IANA Considerations

6.1. UDP Port Number for NSH

IANA is requested to assign the UDP port numbers TBA1 and TBA2 to the NSH from t
he "Service Name and Transport Protocol Port Number Registry" available at
https://www.iana.org/assignments/service-names-port-numbers/service-names-port-n
umbers.xhtml:

Service Name: NSH-in-UDP
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP Encapsulation
Reference: [ThisDocument]
Port Number: TBA1
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

Service Name: NSH-UDP-DTLS
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP with DTLS Encapsulation
Reference: [ThisDocument]
Port Number: TBA2
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

6.2. Protocol Number for NSH

IANA is requested to assign a protocol number TBA3 for the NSH from the "Assigned Internet Protocol Numbers" registry available at https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

Decimal	Keyword	Protocol	IPv6 Extension Header	Reference
TBA3	NSH	Network Service Header	N	[ThisDocument]

7. Acknowledgments

TBD.

8. References

8.1. Normative References

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

[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-12 (work in progress), February 2018.

[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>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <https://www.rfc-editor.org/info/rfc8086>.

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

8.2. Informative References

[I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-09 (work in progress), March 2018.

[I-D.xu-clad-spring-sr-service-chaining]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Decraene, B., Yadlapalli, C., Henderickx, W., Salsano, S., and S. Ma, "Segment Routing for Service Chaining", draft-xu-clad-spring-sr-service-chaining-00 (work in progress), December 2017.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

Authors' Addresses

James N Guichard (editor)
Huawei
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@huawei.com

Haoyu Song
Huawei
2330 Central Express Way
Santa Clara
USA

Email: haoyu.song@huawei.com

Jeff Tantsura
Nuage Networks
USA

Email: jefftant.ietf@gmail.com

Joel Halpern
Ericsson
USA

Email: joel.halpern@ericsson.com

Wim Henderickx
Nokia
USA

Email: wim.henderickx@nokia.com

Mohamed Boucadair
Orange
USA

Email: mohamed.boucadair@orange.com

SPRING
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

J. Guichard, Ed.
H. Song
Huawei
J. Tantsura
Nuage Networks
J. Halpern
Ericsson
W. Henderickx
Nokia
M. Boucadair
Orange
S. Hassan
Cisco Systems
March 11, 2019

NSH and Segment Routing Integration for Service Function Chaining (SFC) draft-guichard-spring-nsh-sr-01

Abstract

This document describes two application scenarios where Network Service Header (NSH) and Segment Routing (SR) techniques can be deployed together to support Service Function Chaining (SFC) in an efficient manner while maintaining separation of the service and transport planes as originally intended by the SFC architecture.

In the first scenario, an NSH-based SFC is created using SR as the transport between SFFs. SR in this case is just one of many encapsulations that could be used to maintain the transport-independent nature of NSH-based service chains.

In the second scenario, SR is used to represent each service hop of the NSH-based SFC as a segment within the segment-list. SR and NSH in this case are integrated.

In both scenarios SR is responsible for steering packets between SFFs along a given SFP while NSH is responsible for maintaining the integrity of the service plane, the SFC instance context, and any associated metadata.

These application scenarios demonstrate that NSH and SR can work jointly and complement each other leaving the network operator with the flexibility to use whichever transport technology makes sense in specific areas of their network infrastructure, and still maintain an end-to-end service plane using NSH.

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 12, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

1.1. SFC Overview and Rationale

The dynamic enforcement of a service-derived, adequate forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators and service providers. Particularly, cascading SFs, for example at the Gi interface in the context of mobile network infrastructure, have shown their limits, such as the same redundant classification features must be supported by many SFs in order to execute their function, some SFs are receiving traffic that they are not supposed to process (e.g., TCP proxies receiving UDP traffic), which inevitably affects their dimensioning and performance, an increased design complexity related to the properly ordered invocation of several SFs, etc.

In order to solve those problems and to avoid the adherence with the underlying physical network topology while allowing for simplified service delivery, Service Function Chaining (SFC) techniques have been introduced.

SFC techniques are meant to rationalize the service delivery logic and master the companion complexity while optimizing service activation time cycles for operators that need more agile service delivery procedures to better accommodate ever-demanding customer requirements. Indeed, SFC allows to dynamically create service planes that can be used by specific traffic flows. Each service plane is realized by invoking and chaining the relevant service functions in the right sequence. [RFC7498] provides an overview of the SFC problem space and [RFC7665] specifies an SFC architecture. The SFC architecture has the merit to not make assumptions on how advanced features (e.g., load-balancing, loose or strict service paths) have to be enabled with a domain. Various deployment options are made available to operators with the SFC architecture and this approach is fundamental to accommodate various and heterogeneous deployment contexts.

Many approaches can be considered for encoding the information required for SFC purposes (e.g., communicate a service chain pointer, encode a list of loose/explicit paths, disseminate a service chain identifier together with a set of context information, etc.). Likewise, many approaches can also be considered for the channel to be used to carry SFC-specific information (e.g., define a new header, re-use existing fields, define an IPv6 extension header, etc.).

Among all these approaches, the IETF endorsed a transport-independent SFC encapsulation scheme: NSH [RFC8300]; which is the most mature SFC encapsulation solution. This design is pragmatic as it does not require replicating the same specification effort as a function of underlying transport encapsulation. Moreover, this design approach encourages consistent SFC-based service delivery in networks enabling distinct transport protocols in various segments of the network or even between SFFs vs SF-SFF hops.

1.2. SFC within SR Networks

As described in [I-D.ietf-spring-segment-routing], Segment Routing (SR) leverages the source routing technique. Concretely, a node steers a packet through an SR policy instantiated as an ordered list of instructions called segments. While initially designed for policy-based source routing, SR also finds its application in supporting SFC [I-D.xu-clad-spring-sr-service-chaining]. The two SR flavors, namely MPLS-SR [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.ietf-6man-segment-routing-header], can both encode a Service Function (SF) as a segment so that an SFC can be specified as a segment list. Nevertheless, and as discussed in [RFC7498], traffic steering is only a subset of the issues that motivated the design of the SFC architecture. Further considerations such as simplifying classification at intermediate SFs and allowing for coordinated behaviors among SFs by means of supplying context information should be taken into account when designing an SFC data plane solution.

While each scheme (i.e., NSH-based SFC and SR-based SFC) can work independently, this document describes how the two can be used together in concert and complement each other through two representative application scenarios. Both application scenarios may be supported using either MPLS-SR or SRv6:

- NSH-based SFC with SR-based transport plane: in this scenario segment routing provides the transport encapsulation between SFFs while NSH is used to convey and trigger SFC polices.

- SR-based SFC with integrated NSH service plane: in this scenario each service hop of the SFC is represented as a segment of the SR segment-list. SR is responsible for steering traffic through the necessary SFFs as part of the segment routing path and NSH is responsible for maintaining the service plane, and holding the SFC instance context and associated metadata.

It is of course possible to combine both of these two scenarios so as to support specific deployment requirements and use cases.

2. NSH-based SFC with SR-based transport tunnel

Because of the transport-independent nature of NSH-based service chains, it is expected that the NSH has broad applicability across different domains of a network. By way of illustration the various SFs involved in a service chain are available in a single data center, or spread throughout multiple locations (e.g., data centers, different POPs), depending upon the operator preference and/or availability of service resources. Regardless of where the service resources are deployed it is necessary to provide traffic steering through a set of SFFs and NSH-based service chains provide the flexibility for the network operator to choose which particular transport encapsulation to use between SFFs, which may be different depending upon which area of the network the SFFs/SFs are currently deployed. Therefore from an SFC architecture perspective, segment routing is simply one of multiple available transport encapsulations that can be used for traffic steering between SFFs. Concretely, NSH does not require to use a unique transport encapsulation when traversing a service chain. NSH-based service forwarding relies upon underlying service node capabilities.

The following three figures provide an example of an SFC established for flow F that has SF instances located in different data centers, DC1 and DC2. For the purpose of illustration, let the SFC's Service Path Identifier (SPI) be 100 and the initial Service Index (SI) be 255.

Referring to Figure 1, packets of flow F in DC1 are classified into an NSH-based SFC and encapsulated after classification as <Inner Pkt><NSH: SPI 100, SI 255><Outer-transport> and forwarded to SFF1 (which is the first SFF hop for this service chain).

After removing the outer transport encapsulation, that may or may not be MPLS-SR or SRv6, SFF1 uses the SPI and SI carried within the NSH encapsulation to determine that it should forward the packet to SF1. SF1 applies its service, decrements the SI by 1, and returns the packet to SFF1. SFF1 therefore has <SPI 100, SI 254> when the packet comes back from SF1. SFF1 does a lookup on <SPI 100, SI 254> which results in <next-hop: DC1-GW1> and forwards the packet to DC1-GW1.

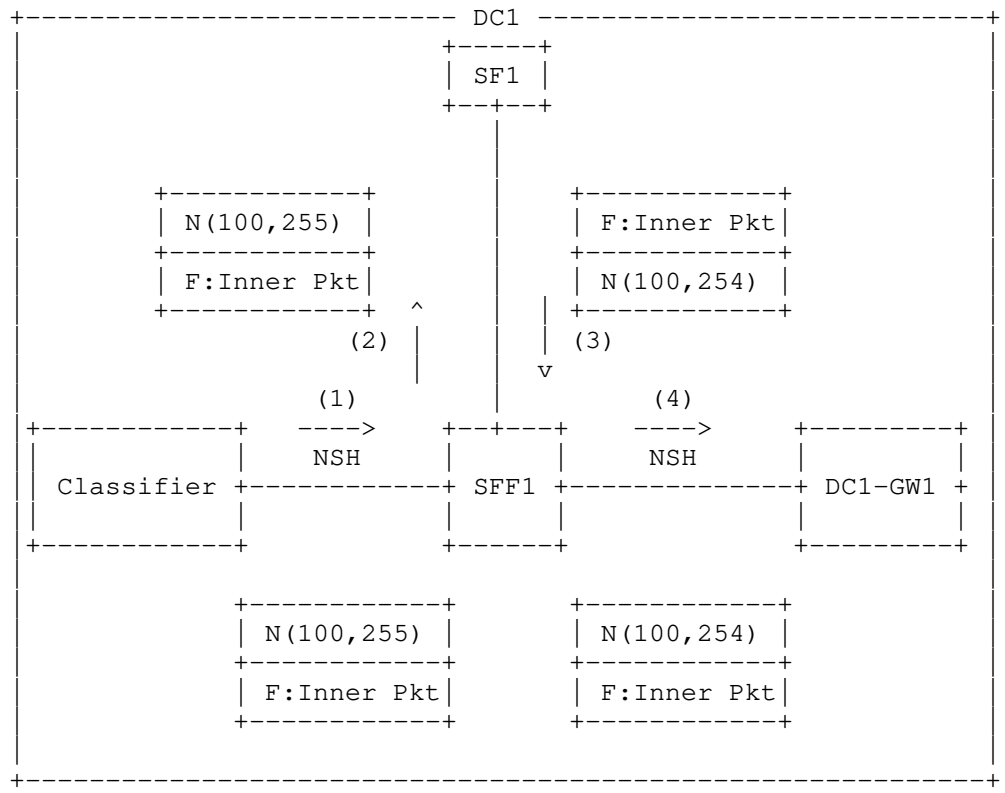


Figure 1: SR for inter-DC SFC - Part 1

Referring now to Figure 2, DC1-GW1 performs a lookup on the information conveyed in the NSH which results in <next-hop: DC2-GW1, encapsulation: SR>. The SR encapsulation has the SR segment-list to forward the packet across the inter-DC network to DC2.

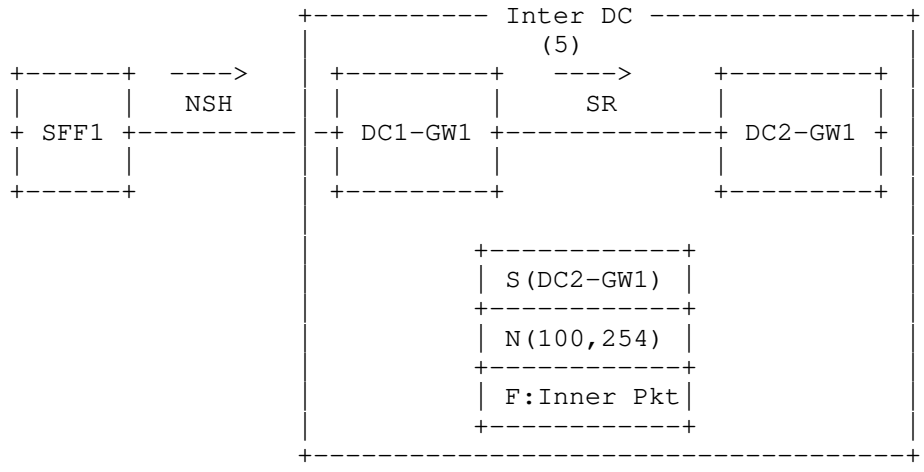


Figure 2: SR for inter-DC SFC - Part 2

When the packet arrives at DC2, as shown in Figure 3, the SR encapsulation is removed and DC2-GW1 performs a lookup on the NSH which results in next-hop: SFF2. The outer transport encapsulation may be any transport that is able to identify NSH as the next protocol.

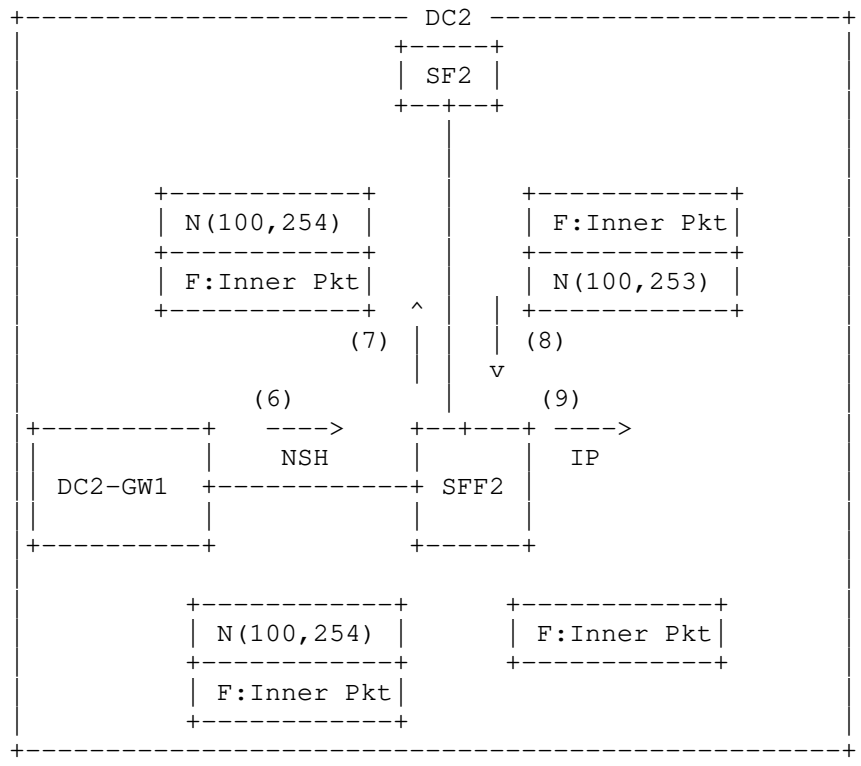


Figure 3: SR for inter-DC SFC - Part 3

The benefits of this scheme are listed hereafter:

- The network operator is able to take advantage of the transport-independent nature of the NSH encapsulation.
- The network operator is able to take advantage of the traffic steering capability of SR where appropriate.
- Light-weight NSH is used in the data center for SFC and avoids more complex hierarchical SFC schemes between data centers.
- Clear responsibility division and scope between NSH and SR.

Note that this scenario is applicable to any case where multiple segments of a service chain are distributed into multiple domains or where traffic-engineered paths are necessary between SFFs (strict forwarding paths for example). Further note that the above example can also be implemented using end to end segment routing between SFF1

and SFF2. (As such DC-GW1 and DC-GW2 are forwarding the packets based on segment routing instructions and are not looking at the NSH header for forwarding).

3. SR-based SFC with Integrated NSH Service Plane

In this scenario we assume that the SFs are NSH-aware and therefore it should not be necessary to implement an SFC proxy to achieve Service Function Chaining. The operation relies upon SR to perform SFF-SFF transport and NSH to provide the service plane between SFs thereby maintaining SFC context and metadata.

When a service chain is established, a packet associated with that chain will first encapsulate an NSH that will be used to maintain the end-to-end service plane through use of the SFC context. The SFC context (e.g., the service plane path referenced by the SPI) is used by an SFF to determine the SR segment list for forwarding the packet to the next-hop SFFs. The packet is then encapsulated using the (transport-specific) SR header and forwarded in the SR domain following normal SR operation.

When a packet has to be forwarded to an SF attached to an SFF, the SFF performs a lookup on the prefix SID associated with the SF to retrieve the next-hop context between the SFF and SF. E.g. to retrieve the destination MAC address in case native ethernet encapsulation is used between SFF and SF. How the next-hop context is populated is out of the scope of this document. The SFF strips the SR information of the packet, updates the SR information, and saves it to a cache indexed by the NSH SPI. This saved SR information is used to encapsulate and forward the packet(s) coming back from the SF.

When the SF receives the packet, it processes it as usual and sends it back to the SFF. Once the SFF receives this packet, it extracts the SR information using the NSH SPI as the index into the cache. The SFF then pushes the SR header on top of the NSH header, and forwards the packet to the next segment in the segment list.

Figure 4 illustrates an example of this scenario.

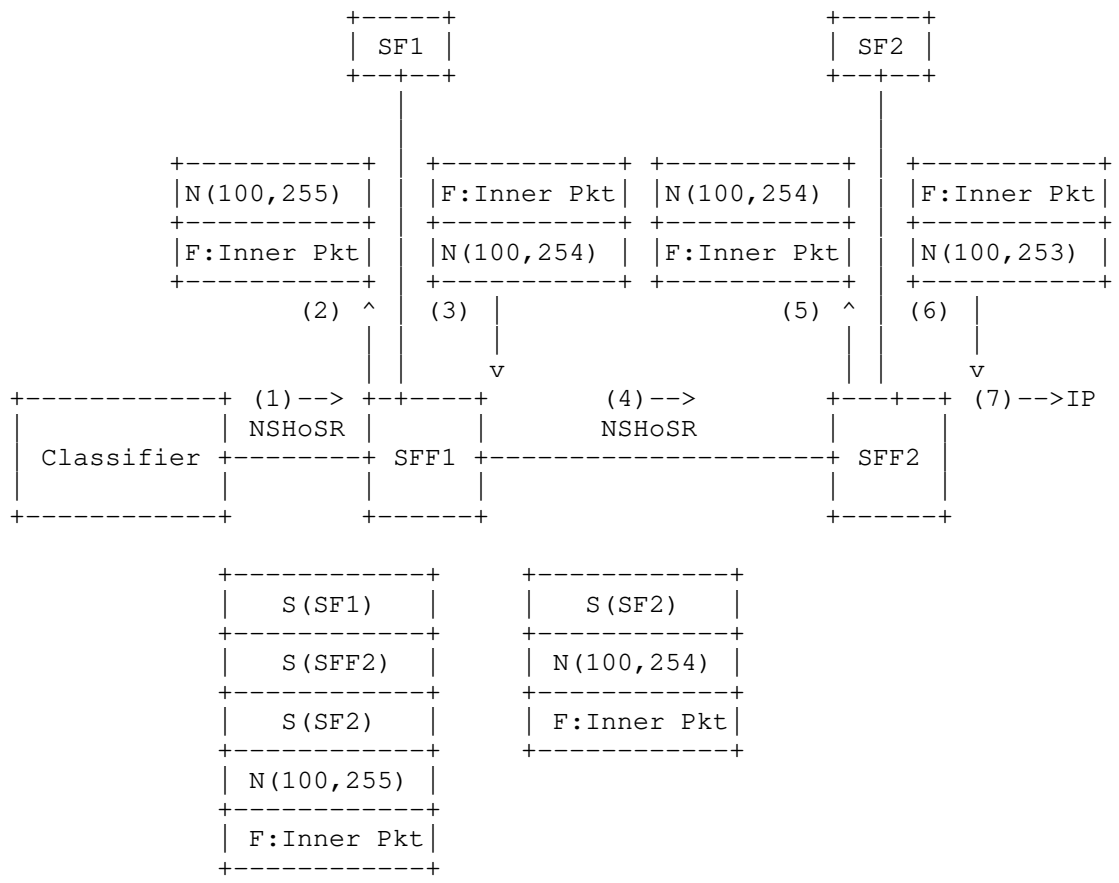


Figure 4: NSH over SR for SFC

The benefits of this scheme include:

- It is economically sound for SF vendors to only support one unified SFC solution. The SF is unaware of the SR.
- It simplifies the SFF (i.e., the SR router) by nullifying the needs for re-classification and SR proxy.
- It provides a unique and standard way to pass metadata to SFs. Note that currently there is no solution for MPLS-SR to carry metadata and there is no solution to pass metadata to SR-unaware SFs.
- SR is also used for forwarding purposes including between SFFs.

- o It takes advantage of SR to eliminate the NSH forwarding state in SFFs. This applies each time strict or loose SFPs are in use.

- o It requires no interworking as would be the case if MPLS-SR based SFC and NSH-based SFC were deployed as independent mechanisms in different parts of the network.

4. Encapsulation Details

4.1. NSH using MPLS-SR Transport

MPLS-SR instantiates Segment IDs (SIDs) as MPLS labels and therefore the segment routing header is a stack of MPLS labels.

When carrying NSH within an MPLS-SR transport, the full encapsulation headers are as illustrated in Figure 5.

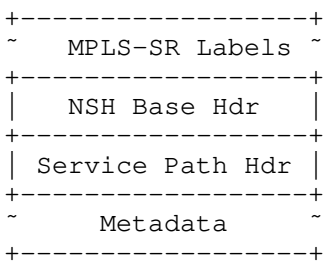


Figure 5: NSH using MPLS-SR Transport

As described in [I-D.ietf-spring-segment-routing] the IGP signaling extension for IGP-Prefix segment includes a flag to indicate whether directly connected neighbors of the node on which the prefix is attached should perform the NEXT operation or the CONTINUE operation when processing the SID. When NSH is carried beneath MPLS-SR it is necessary to terminate the NSH-based SFC at the tail-end node of the MPLS-SR label stack. This is the equivalent of MPLS Ultimate Hop Popping (UHP) and therefore the prefix-SID associated with the tail-end of the SFC MUST be advertised with the CONTINUE operation so that the penultimate hop node does not pop the top label of the MPLS-SR label stack and thereby expose NSH to the wrong SFF. It is RECOMMENDED that a specific prefix-SID be allocated at each node for use by the SFC application for this purpose.

At the end of the MPLS-SR path it is necessary to provide an indication to the tail-end that NSH follows the MPLS-SR label stack.

There are several ways to achieve this but its specification is outside the scope of this document.

4.2. NSH using SRv6 Transport

When carrying NSH within an SRv6 transport the full encapsulation is as illustrated in Figure 6.

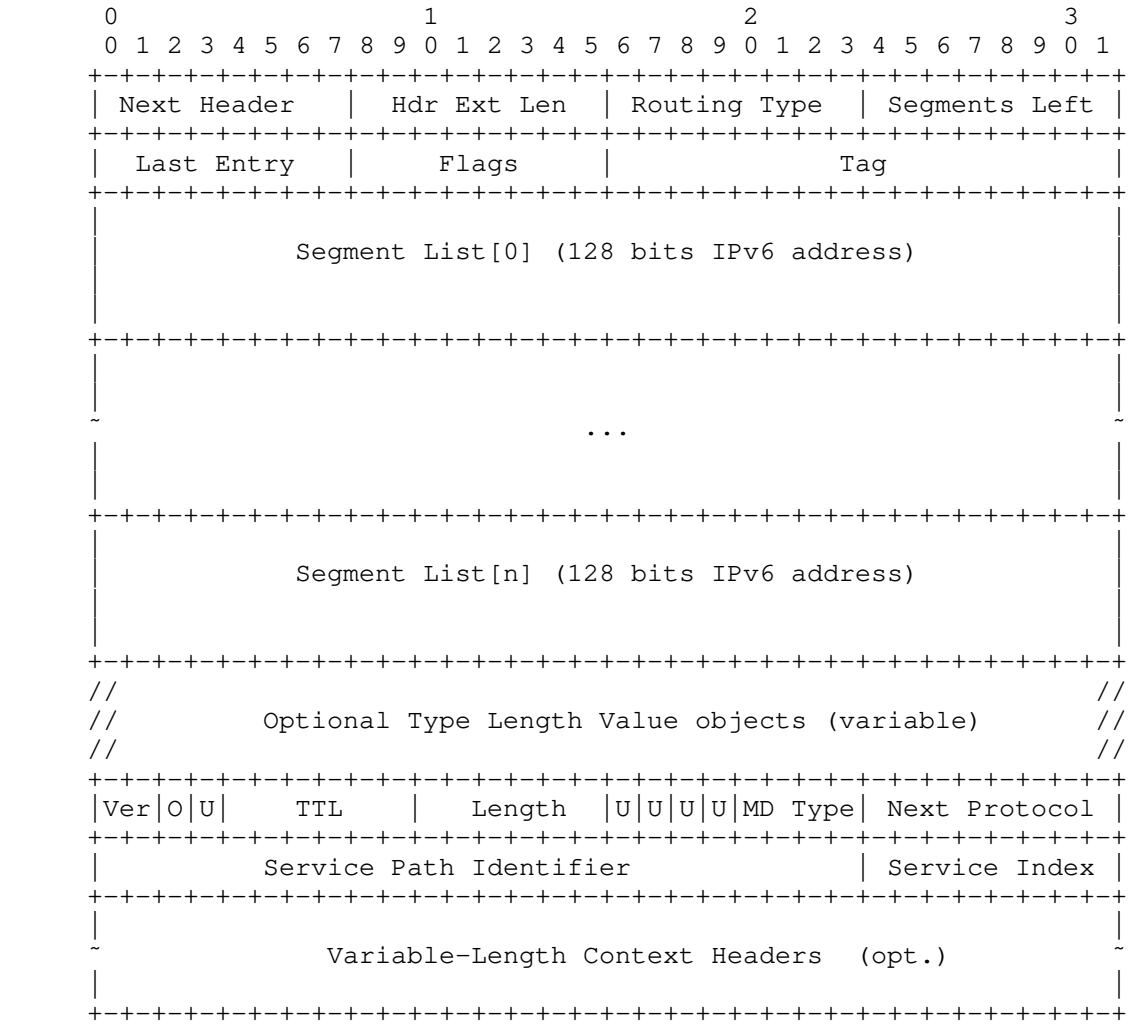


Figure 6: NSH using SRv6 Transport

Encapsulation of NSH following SRv6 may be indicated either by encapsulating NSH in UDP (UDP port TBA1) and indicating UDP in the Next Header field of the SRH, or by indicating an IP protocol number for NSH in the Next Header of the SRH. The behavior for encapsulating NSH over UDP, including the selection of the source port number in particular, adheres to similar considerations as those discussed in [RFC8086].

5. Security Considerations

Generic SFC-related security considerations are discussed in [RFC7665]. NSH-specific security considerations are discussed in [RFC8300]. NSH-in-UDP with DTLS [RFC6347] should follow the considerations discussed in Section 5 of [RFC8086], with a destination port number set to TBA2

6. IANA Considerations

6.1. UDP Port Number for NSH

IANA is requested to assign the UDP port numbers TBA1 and TBA2 to the NSH from t
he "Service Name and Transport Protocol Port Number Registry" available at
https://www.iana.org/assignments/service-names-port-numbers/service-names-port-n
umbers.xhtml:

Service Name: NSH-in-UDP
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP Encapsulation
Reference: [ThisDocument]
Port Number: TBA1
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

Service Name: NSH-UDP-DTLS
Transport Protocol(s): UDP
Assignee: IESG iesg@ietf.org
Contact: IETF Chair chair@ietf.org
Description: NSH-in-UDP with DTLS Encapsulation
Reference: [ThisDocument]
Port Number: TBA2
Service Code: N/A
Known Unauthorized Uses: N/A
Assignment Notes: N/A

6.2. Protocol Number for NSH

IANA is requested to assign a protocol number TBA3 for the NSH from the "Assigned Internet Protocol Numbers" registry available at https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

Decimal	Keyword	Protocol	IPv6 Extension Header	Reference
TBA3	NSH	Network Service Header	N	[ThisDocument]

7. Acknowledgments

TBD.

8. References

8.1. Normative References

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

[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-12 (work in progress), February 2018.

[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>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <https://www.rfc-editor.org/info/rfc8086>.

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

8.2. Informative References

[I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-09 (work in progress), March 2018.

[I-D.xu-clad-spring-sr-service-chaining]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Decraene, B., Yadlapalli, C., Henderickx, W., Salsano, S., and S. Ma, "Segment Routing for Service Chaining", draft-xu-clad-spring-sr-service-chaining-00 (work in progress), December 2017.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

Authors' Addresses

James N Guichard (editor)
Huawei
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@huawei.com

Haoyu Song
Huawei
2330 Central Express Way
Santa Clara
USA

Email: haoyu.song@huawei.com

Jeff Tantsura
Nuage Networks
USA

Email: jefftant.ietf@gmail.com

Joel Halpern
Ericsson
USA

Email: joel.halpern@ericsson.com

Wim Henderickx
Nokia
USA

Email: wim.henderickx@nokia.com

Mohamed Boucadair
Orange
USA

Email: mohamed.boucadair@orange.com

Syed Hassan
Cisco Systems
USA

Email: shassan@cisco.com

Routing area
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

S. Hegde
K. Arora
Juniper Networks Inc.

October 16, 2018

Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) Egress Peer engineering Segment Identifiers (SIDs) with MPLS Data Planes

draft-hegde-mpls-spring-epe-oam-00

Abstract

Egress Peer Engineering is an application of Segment Routing to solve the problem of egress peer selection. The SR-based BGP-EPE solution allows a centralized (Software Defined Network, SDN)controller to program any egress peer. The EPE solution requires a node to program PeerNodeSID, PeerAdjSID, PeerSetSID as described in [I-D.ietf-spring-segment-routing-central-epe].This document provides Target FEC stack TLV definitions as defined in [RFC8029] for the EPE SIDs.

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 April 19, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

Egress Peer Engineering (EPE) as defined in [I-D.ietf-spring-segment-routing-central-epe] is an effective mechanism to select the egress peer link based on different criteria. The EPE SIDs provide means to represent egress peer links. Many network deployments have built their networks consisting of multiple Autonomous Systems either for ease of operations or as a result of network mergers and acquisitons. The egress links connecting the two Autonomous Systems could be managed using EPE-SIDs in this case as well. It is important to be able to validate the control plane to forwarding plane synchronization for these SIDs so that any anomaly can be detected easily by the operator.

This document provides Target FEC stack TLV definitions for EPE SIDs. Other procedures for mpls ping and traceroute as defined in [RFC8287] are applicable for EPE-SIDs as well.

2. FEC Definitions

As described in [RFC8287] sec 5, 3 new type of segment IDs are defined for the Target FEC stack TLV corresponding to each label in the label stack

2.1. PeerNodeSID/PeerAdjSID

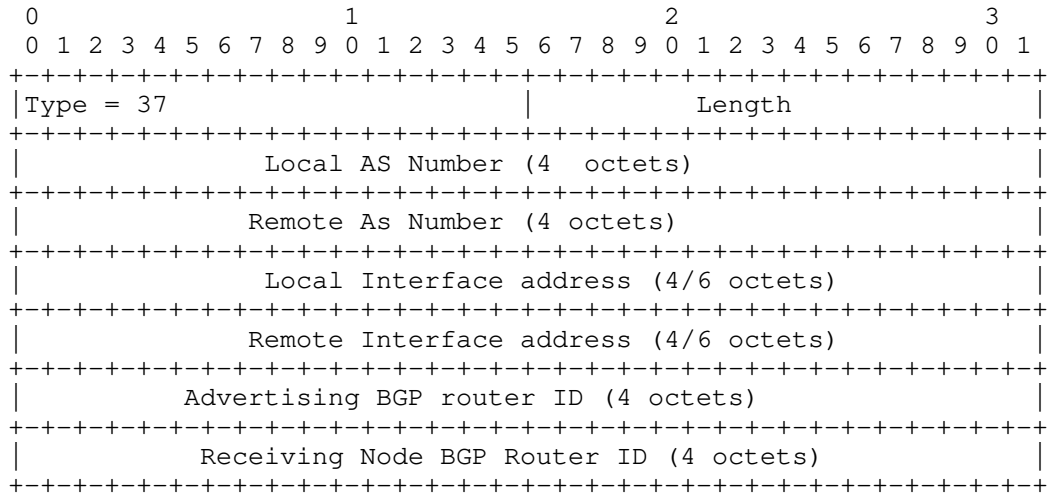


Figure 1: Peer Node/Adj Segment ID Sub TLV

Type: 37 (TBD)

Length: variable based on ipv4/ipv6 interface address

AS Number: 4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]

Interface Address: BGP session IPv4/IPv6 local/remote address.

BGP Router ID: 4 octet unsigned integer representing the BGP Identifier as defined in [RFC4271] and [RFC6286].

2.2. PeerSetSID

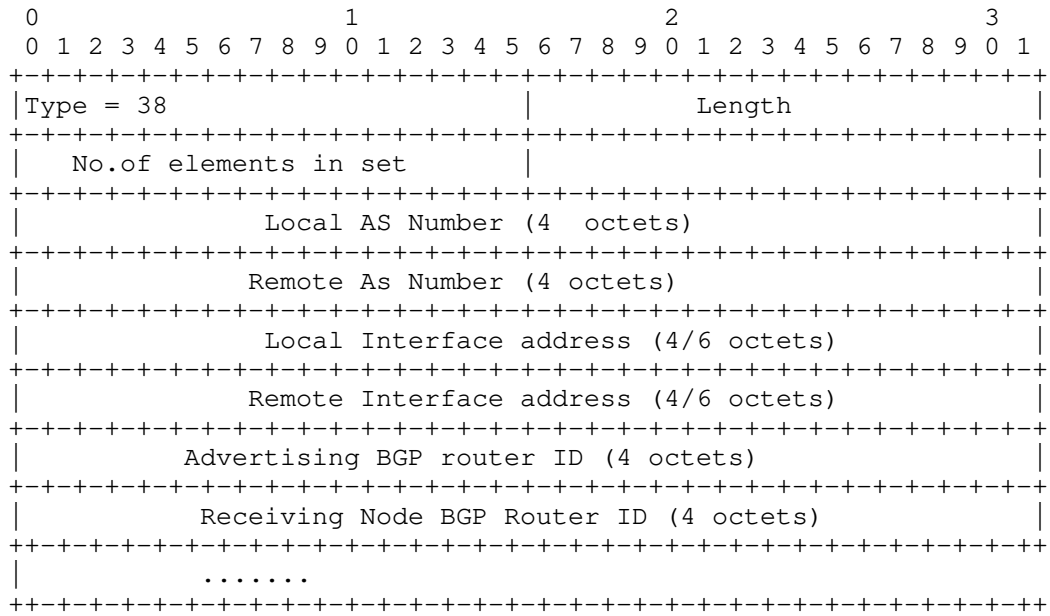


Figure 2: Peer set SID Segment ID Sub TLV

Type : 38 (TBD)

Length : variable based on ipv4/ipv6 interface address

No.of elements in set : Number of links in the set.

AS Number : 4 octet unsigned integer representing the Member ASN inside the Confederation.[RFC5065]

Interface Address : BGP session IPv4/IPv6 local/remote address.

BGP Router ID : 4 octet unsigned integer representing the BGP Identifier as defined in [RFC4271] and [RFC6286]

3. Security Considerations

TBD

4. IANA Considerations

New Target FEC stack sub-TLV from the "sub-TLVs for TLV types 1,16 and 21" subregistry of the "Multi-Protocol Label switching (MPLs) Label Switched Paths 9LSPs) Ping parameters" registry

PeerNode/PeerAdjSID segment ID Sub-TLV : 37 (suggested)

PeerSetSID segment ID Sub-TLV : 38 (suggested)

5. Acknowledgments

6. References

6.1. Normative References

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

[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>.

6.2. 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>.

[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>.

Authors' Addresses

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

Email: shraddha@juniper.net

Kapil Arora
Juniper Networks Inc.

Email: kapilaro@juniper.net

SPRING
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

K. Kompella
A. Deshmukh
R. Torvi
Juniper Networks, Inc.

October 22, 2018

Resilient MPLS Rings
draft-kompella-spring-rmr-00

Abstract

This document describes the use of the SPRING MPLS data plane for Resilient MPLS Rings. It describes how to create the bidirectional ring LSPs with SPRING, and how protection works.

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 April 25, 2019.

Copyright Notice

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

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

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

Table of Contents

1. Introduction

Rings are a very common topology in transport networks. A ring is the simplest topology offering link and node resilience. Rings are nearly ubiquitous in access and aggregation networks. As MPLS increases its presence in such networks, and takes on a greater role in transport, it is imperative that MPLS handles rings well; [I-D.ietf-mpls-rmr] shows how this can be done. [I-D.ietf-teas-rsvp-rmr-extension] shows how RSVP-TE [RFC3209] can be used to signal RMR ring LSPs. [I-D.ietf-mpls-ldp-rmr-extensions] shows how LDP [RFC5036] can be used to signal RMR LSPs. This document shows how SPRING SID bindings can be used to create RMR LSPs, how the basic bidirectional LSPs are set up, and how protection works.

While RMR looks at rings potentially with "express links", this document focuses on simple rings. These are most common in access networks. Future revisions will look at more general rings.

1.1. Definitions

A (directed) graph G = (V, E) consists of a set of vertices (or nodes) V and a set of edges (or links) E. An edge is an ordered pair of nodes (a, b), where a and b are in V. (In this document, the terms node and link will be used instead of vertex and edge.)

A ring is a subgraph of G. A ring consists of a subset of n nodes {R_i, 0 <= i < n} of V. The directed edges {(R_i, R_i+1) and (R_i+1, R_i), 0 <= i < n-1} must be a subset of E (note that index arithmetic is done modulo n). We define the direction from node R_i to R_i+1 as "clockwise" (CW) and the reverse direction as "anticlockwise" (AC). As there may be several rings in a graph, we number each ring with a distinct ring ID RID.

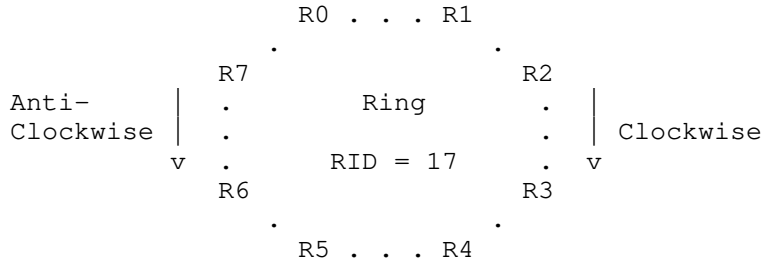


Figure 1: Ring with 8 nodes

The following terminology is used for ring LSPs:

Ring ID (RID): A non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network. A node may belong to multiple rings.

Ring node: A member of a ring. Note that a device may belong to several rings.

Node index: A logical numbering of nodes in a ring, from zero upto one less than the ring size. Used purely for exposition in this document.

Ring master: The ring master initiates the ring identification process. Mastership is indicated in the IGP by a two-bit field.

Ring neighbors: Nodes whose indices differ by one (modulo ring size).

Ring size: The ring size for a given instantiation is N. This can change as nodes are added or removed, or go up or down.

Ring links: Links that connnect ring neighbors.

Express links: Links that connnect non-neighboring ring nodes.

Ring direction: A two-bit field in the IGP indicating the direction of a link. The choices are:

UN: 00 undefined link

CW: 01 clockwise ring link

AC: 10 anticlockwise ring link

EX: 11 express link

Ring Identification: The process of discovering ring nodes, ring links, link directions, and express links.

The following notation is used for ring LSPs:

R_k: A ring node with index k. R_k has AC neighbor R_(k-1) and CW neighbor R_(k+1).

NS_k: Node SID for node R_k. Note that index arithmetic is done modulo the ring size N.

CAS_k, AAS_k: Clockwise adjacency SID at R_k, i.e., link R_k, R_k+1 and anticlockwise adjacency SID R_k, R_k-1 respectively. Note that index arithmetic is done modulo the ring size N.

CSS_jk: A clockwise node SID stack, typically with one or two SIDs, to be pushed by R_j to reach R_k in a clockwise direction.

ASS_jk: An anticlockwise node SID stack, typically with one or two SIDs, to be pushed by R_j to reach R_k in an anticlockwise direction.

2. Motivation

A ring is the simplest topology that offers resilience. This is perhaps the main reason to lay out fiber in a ring. Thus, effective mechanisms for fast failover on rings are needed. Furthermore, there are large numbers of rings. Thus, configuration of rings needs to be as simple as possible.

The goals of this document are to present mechanisms for improved resilience in ring networks (using ideas that are reminiscent of Bidirectional Line Switched Rings), for automatic bring-up of LSPs, better bandwidth management and for auto-hierarchy. These goals are achieved using extensions to existing IGP. This document shows how to do this using SPRING techniques, in particular, node SIDs. Note

that in a simple ring topology, there is no need for complex algorithms to find loop-free protection paths.

3. Theory of Operation

We assume that a ring R has been configured, IGP advertisements have been made, and ring discovery is complete ([I-D.ietf-mpls-rmr]). We also assume that node and adjacency SIDs have been distributed.

3.1. Installing Primary LFIB Entries

Ring LSPs are not provisioned. Once a ring node R_i knows its RID, its ring links and directions, it kicks off ring LSP computation automatically. In particular, R_j computes clockwise and anticlockwise SID stacks CSS_jk and ASS_jk to node R_k. R_j then installs two FIB entries for R_k, CSS_jk and ASS_jk. It is up to an application to choose whether to go clockwise or anticlockwise from R_j to R_k.

R_j also computes CSS_jj and ASS_jj. Clearly, R_j does not act as ingress for its own LSPs. However, R_j can send OAM messages, for example, an MPLS ping or traceroute ([I-D.ietf-mpls-rfc4379bis]), using CSS_jj or ASS_jj, to test the entire ring LSP anchored at R_j in both directions.

3.2. Installing Protection LFIB Entries

At the same time that R_j sets up its primary clockwise and anticlockwise SID stacks, it sets up protection for each other node R_k. R_j does this by installing a protection SID stack for the node SID to R_k, NS_k. If the shortest path to R_k is clockwise, then the protection SID stack for NS_k is ASS_jk. Otherwise, it is CSS_jk.

Similarly, the protection entry for an adjacency SID CAS_j is ASS_j,j+1 and for AAS_j is CSS_j,j-1.

3.3. Protection

If a node R_j detects a failure from R_j+1 -- either all links to R_j+1 fail, or R_j+1 itself fails, R_j switches traffic on all CW node and adjacency SIDs to their protection LFIB entries. This switchover can be very fast, as the protection LFIB entries can be preprogrammed. Fast detection and fast switchover lead to minimal traffic loss.

R_j then sends an indication to R_j-1 that the CW direction is not working, so that R_j-1 can similarly switch traffic to the AC direction. This can be by an IGP update; other, potentially quicker,

mechanisms would be preferable. These indications propagate AC until each traffic source on the ring AC of the failure is aware of the failure. Thus, within a short period, traffic will be flowing on the reverse path than that which was chosen, since there is a failure on the ring.

4. Security Considerations

It is not anticipated that either the notion of MPLS rings or the extensions to various protocols to support them will cause new security loopholes. As this document is updated, this section will also be updated.

5. IANA Considerations

There are no requests as yet to IANA for this document.

6. References

6.1. Normative References

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

6.2. Informative References

[I-D.ietf-mpls-ldp-rmr-extensions]
Esale, S. and K. Kompella, "LDP Extensions for RMR", 2018.

[I-D.ietf-mpls-rfc4379bis]
Kompella, K., Swallow, G., Pignataro, C., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", draft-ietf-mpls-rfc4379bis-09 (work in progress), October 2016.

[I-D.ietf-mpls-rmr]
Kompella, K. and L. Contreras, "Resilient MPLS Rings", 2018.

[I-D.ietf-teas-rsvp-rmr-extension]
Deshmukh, A. and K. Kompella, "RSVP Extensions for RMR", 2018.

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

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

Authors' Addresses

Kireeti Kompella
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: kireeti.kompella@gmail.com

Abhishek Deshmukh
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: adeshmukh@juniper.net

Ravi Torvi
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: rtorvi@juniper.net

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2020

K. Kompella
T. Saad
A. Deshmukh
Juniper Networks
March 08, 2020

Resilient MPLS Rings using SPRING
draft-kompella-spring-rmr-02

Abstract

This document describes the use of SPRING to setup LSP(s) for resilient MPLS ring networks. It specifies how clockwise and anti-clockwise ring SIDs are allocated and signaled using IGP protocol extensions, and how such ring SIDs achieve ring protection.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 9, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction

Ring topologies are very common in transport networks, and are ubiquitous in access and aggregation networks.

This draft introduces extensions to IGP protocols to establish SPRING Label Switched Paths (LSPs) for Resilient MPLS Rings (RMR). An RMR LSP is a multipoint to point (MP2P) LSP with simple protection.

SPRING [RFC8402] defines the notion of node SIDs which guide packets along the IGP shortest path to the advertising node. Node SIDs are typically advertised by an IGP. This draft uses a similar notion to guide packets along a "clockwise" (CW) or "anti-clockwise" (AC) path to the advertising node. The concept of CW and AC are well-defined in RMR rings.

Rings are auto-discovered using the mechanisms described in the [I-D.ietf-mpls-rmr]. Signaling extensions for IS-IS and OSPF are introduced in [I-D.kompella-isis-ospf-rmr] to enable the auto-discovery of ring topologies. After the ring topology is discovered, each node in the ring determines its CW and AC ring neighbors and associated ring links.

[I-D.ietf-teas-rsvp-rmr-extension] describes RSVP-TE [RFC3209] extensions to set up Resilient MPLS Ring (RMR) LSPs. [I-D.ietf-mpls-ldp-rmr-extensions] describes LDP [RFC5036] extensions to set up LDP RMR LSPs. This document does the same for SPRING RMR LSPs and the associated protection LSPs.

The first revisions of this document will focus on the simple most common ring topologies in access networks that do not include any "express links". Future revisions of this document will expand on express link(s) for the more general rings.

2. Terminology

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

2.1. Ring Terminology

A ring consists of a subset of n nodes {R_i, 0 <= i < n}. The direction from node R_i to R_i+1 is defined as as "clockwise" (CW) and the reverse direction is defined as "anti-clockwise" (AC). As there may be several rings in a graph, each ring is numbered with a distinct Ring ID (RID).

The following terminology is used for rings:

Ring ID (RID): A non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network. A node may belong to multiple rings, each identified by its unique RID

Ring Node: A member of a ring. Note that a node may belong to several rings.

Node Index: A logical numbering of nodes in a ring, from zero up to one less than the ring size. Used purely for exposition in this document.

Ring Master: The ring master initiates the ring identification process. Mastership is indicated in the IGP by a two-bit field.

Ring Neighbors: Nodes whose indices differ by one (modulo ring size).

Ring Size: The ring size for a given instantiation is N. This can change as nodes are added or removed, or go up or down.

Ring Links: Links that connect ring neighbors.

Express Links: Links that connect non-neighboring ring nodes.

Ring LSP: Each LSP in the ring is a multipoint to point LSP such that LSP can have multiple ingress nodes and one egress node.

Ring Identification: The process of discovering ring nodes, ring links, link directions, and express links.

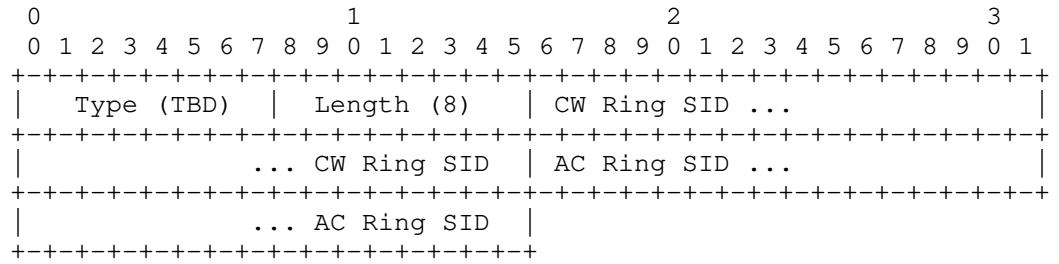
Ring SID: Each ring node W advertises a pair of unique Ring Segment Identifiers (Ring SIDs): a CW ring SID CW-W, to send traffic clockwise to W, and an AC ring SID AC-W to send traffic anti-clockwise to W.

3. Protocol extensions

A node participating in an RMR ring that is capable of SPRING can choose to advertise this by setting the SPRING bit in the "Supported Signaling Protocols" (SS), as specified in [I-D.ietf-mpls-rmr]. If so, the node advertises a Ring SID sub-TLV as a sub-TLV of the RMR Node TLV, as follows:

[RMR-SID Type] [Length = 8] [CW Ring SID Index] [AC Ring SID Index]

This sub-TLV has the following format in IS-IS:



where:

Type: is the type of the RMR SID sub-TLV (TBD)

Length: 8

Value: The CW SID followed by the AC SID.

In OSPF, the sub-TLV has the same format, except that the Type is two octets and the Length is two octets.

4. Ring SPRING LSPs

The semantics of Ring SIDs is very straightforward: a CW Ring SID advertised by node W (CW-W) simply says: to get to W, take the CW path. Similarly, the AC Ring SID AC-W to W says, take the AC path to W.

Protection involves switching directions and labels.

4.1. Ring SID Assignment

CW and AC Ring SIDs MUST be unique for each (RID, node) combination. This uniqueness can be guaranteed by configuration, or by the use of DHCP, as described next.

4.1.1. Ring SID Assignment Using DHCP

The Dynamic Host Configuration Protocol is uniquely well-suited for handling node and ring SID assignments. When ring directions have been established for all links in the ring, each node can request, as a DHCP client, a pair of ring SIDs. The DHCP server responds with two unique values from the SID block(s) for Ring SIDs with which it has been configured. The DHCP server SHOULD be configured with very long leases for such assignments, as well as "sticky" assignments; that is, should a lease expire, the pair of values assigned should not be offered to another client unless the server has run out of ring SID values; also, should the same client re-request ring SIDs, the server SHOULD return the same SIDs if at all possible.

Further details are provided in [I-D.kompella-spring-dhcp].

4.2. Ring SPRING LSP Set Up

When ring identification of ring R is complete, each node W that is SPRING capable advertises a pair of CW and AC Ring SIDs, CW-W and AC-W. Each node Y that is a member of ring R then installs a FIB entry as follows:

1. Let X be Y's AC neighbor, and Z be Y's CW neighbor.

2. For each received CW Ring SID CW-W, Y installs an LFIB entry to forward the packet to Z.

3. For each received CW Ring SID AC-W, Y installs an LFIB entry to forward the packet to X.

Note that Y must deal with the SRGB mapping corresponding to X and Z.

4.3. Protection and Fastreroute

At the same time that Y installs LFIB entries for CW-W and AC-W, it also installs backup LFIB entries as follows:

1. Let X be Y's AC neighbor, and Z be Y's CW neighbor.

2. For each received CW Ring SID CW-W, Y installs a backup LFIB entry to swap CW-W with AC-W and forward the packet to X (swap label and ring direction). If all nodes AC of Y until W are NFFRR capable, the NFFRR SPL [I-D.kompella-mpls-nffrr] SHOULD be pushed below AC-W.

3. For each received AC Ring SID AC-W, Y installs a backup LFIB entry to swap AC-W with CW-W and forward the packet to Z (swap label and ring direction). If all nodes CW of Y until W are NFFRR capable, the NFFRR SPL SHOULD be pushed below CW-W.

Again, Y must deal with the corresponding SRGB mappings.

5. IANA Considerations

Need to allocate Ring SID sub-TLV Types for IS-IS and OSPF.

6. Security Considerations

It is not anticipated the extensions to IGP SR protocols described in this document may introduce additional security risk(s). Future revisions of this document will update this section with details about those risks.

7. Contributors

Raveendra Torvi
Juniper Networks

Email: rtorvi@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

8. References

8.1. Normative References

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

[I-D.kompella-isis-ospf-rmr]
Kompella, K., "IGP Extensions for Resilient MPLS Rings", draft-kompella-isis-ospf-rmr-00 (work in progress), October 2016.

[I-D.kompella-mpls-nffrr]
Kompella, K. and W. Lin, "No Further Fast Reroute", draft-kompella-mpls-nffrr-00 (work in progress), March 2020.

[I-D.kompella-spring-dhcp]
Kompella, K. and R. Bonica, "Using DHCP to Manage Node and Ring SID Assignment", draft-kompella-spring-dhcp-00 (work in progress), July 2019.

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

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

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

8.2. Informative References

[I-D.ietf-mpls-ldp-rmr-extensions]
Esale, S. and K. Kompella, "LDP Extensions for RMR", draft-ietf-mpls-ldp-rmr-extensions-02 (work in progress), June 2019.

[I-D.ietf-teas-rsvp-rmr-extension]
Deshmukh, A. and K. Kompella, "RSVP Extensions for RMR", draft-ietf-teas-rsvp-rmr-extension-02 (work in progress), July 2019.

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

[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>.

Authors' Addresses

Kireeti Kompella
Juniper Networks

Email: kireeti.kompella@gmail.com

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Abhishek Deshmukh
Juniper Networks

Email: adeshmukh@juniper.net

Network Work group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2019

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

October 23, 2018

Label Switched Path (LSP) Ping/Traceroute for Segment Routing SIDs with MPLS Data-plane
draft-nainar-mpls-spring-lsp-ping-sids-00

Abstract

RFC8402 introduces Segment Routing architecture that leverages source routing and tunneling paradigms and can be directly applied to the Multi Protocol Label Switching (MPLS) data plane. A node steers a packet through a controlled set of instructions called segments, by prepending the packet with Segment Routing header. SR architecture defines different types of segments with different forwarding semantics associated.

RFC8287 defines the extensions to MPLS LSP Ping and Traceroute for Segment Routing IGP-Prefix and IGP-Adjacency Segment Identifier (SIDs) with an MPLS data plane. RFC8287 defines the Target FEC Stack Sub-TLVs and the procedures to apply RFC8029 on SR architecture with MPLS data plane.

This document defines the Target FEC Stack Sub-TLVs and the extension required for other SR Segments.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 26, 2019.

Copyright Notice

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

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

Table of Contents

1. Introduction

[RFC8402] introduces and describes a Segment Routing architecture that leverages the source routing and tunneling paradigms. A node steers a packet through a controlled set of instructions called segments, by prepending the packet with Segment Routing header. A detailed definition of the Segment Routing architecture is available in [RFC8402]

As described in [RFC8402] and [I-D.ietf-spring-segment-routing-mpls], the Segment Routing architecture can be directly applied to an MPLS data plane, the Segment identifier (Segment ID) will be of 20-bits size and the Segment Routing header is the label stack.

[RFC8287] defines the mechanism to perform LSP Ping and Traceroute for Segment Routing with MPLS data plane. [RFC8287] defines the Target FEC Stack Sub-TLVs for IGP-Prefix Segment ID and IGP-Adjacency Segment ID.

There are various other Segment IDs proposed by different documents that are applicable for SR architecture.
[I-D.ietf-idr-bgp-prefix-sid] defines BGP Prefix Segment ID,
[I-D.ietf-idr-bgpls-segment-routing-epe] defines BGP Peering Segment ID such as Peer Node SID, Peer Adj SID and Peer Set SID.
[I-D.sivabalan-pce-binding-label-sid] defines Path Binding Segment ID.

As above Segment IDs get deployed in the field, operators require corresponding MPLS OAM procedures for the SIDs. This document describes the target FEC Stack Sub-TLVs and the procedure to use LSP Ping and Traceroute for the above defined Segment IDs to support path validation and fault isolation.

2. Requirements notation

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

3. Terminology

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

The term "BGP EPE node" is used to refer to node assigning and advertising BGP Peering Segment SIDs to steer traffic towards a BGP peer, as described in [I-D.ietf-spring-segment-routing-central-epe].

4. Segment ID sub-TLV

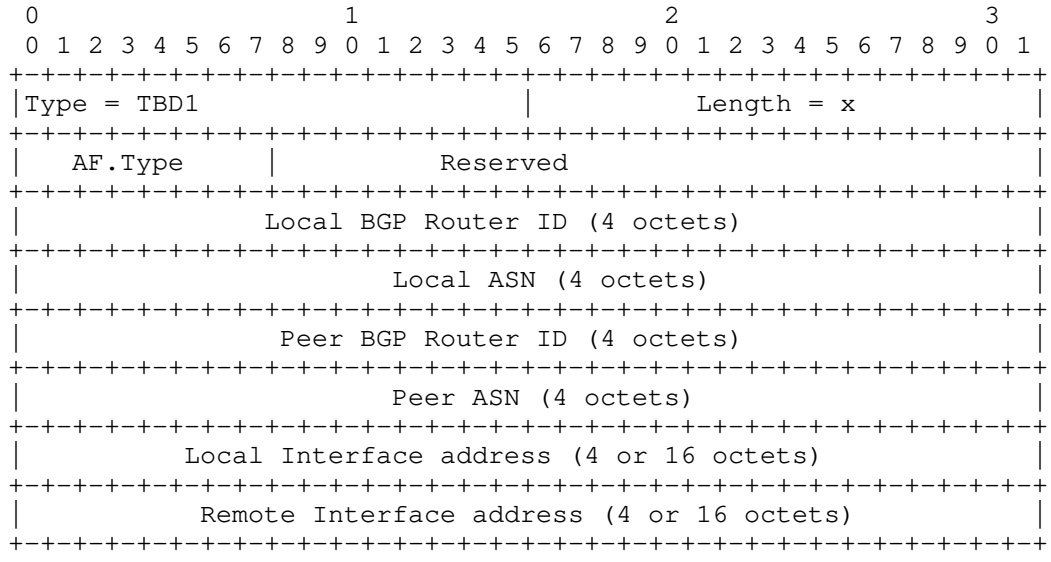
As defined in Section 5 of [RFC8287], the format of the following Segment ID sub-TLVs defined in this document follows the philosophy of Target FEC Stack TLV carrying FECs corresponding to each label in the label stack.

4.1. BGP Prefix Segment ID

Section 3.2.13 and 3.2.14 of [RFC8029] defines the Sub-TLV for BGP labeled IPv4 and IPv6 prefix respectively. This document proposes the use of the same Sub-TLV for IPv4 and IPv6 BGP Prefix SID without any change.

4.2. BGP Peering Segment - Peer-Node-SID

Peer-Node-SID identifies the peer node in the BGP Peering Segment. The sub-TLV format for Peer-Node-SID of BGP Peering Segment MUST be set as shown in the below TLV format:



AF.Type

Set to 4 if the address in Local/Remote Interface address field is IPv4 and set to 6 if the address in Local/Remote Interface address field is IPv6.

Reserved

MUST be set to 0 on send and MUST be ignored on receipt.

Local BGP Router ID

4-octet BGP Router ID of the node that assigns the Peer-Node-SID.

Local ASN

4-octet local ASN number of the node that assigns the Peer-Node-SID.

Peer BGP Router ID

4-octet BGP Router ID of the peer node.

Peer ASN

4-octet ASN number of the peer node.

Local Interface Address

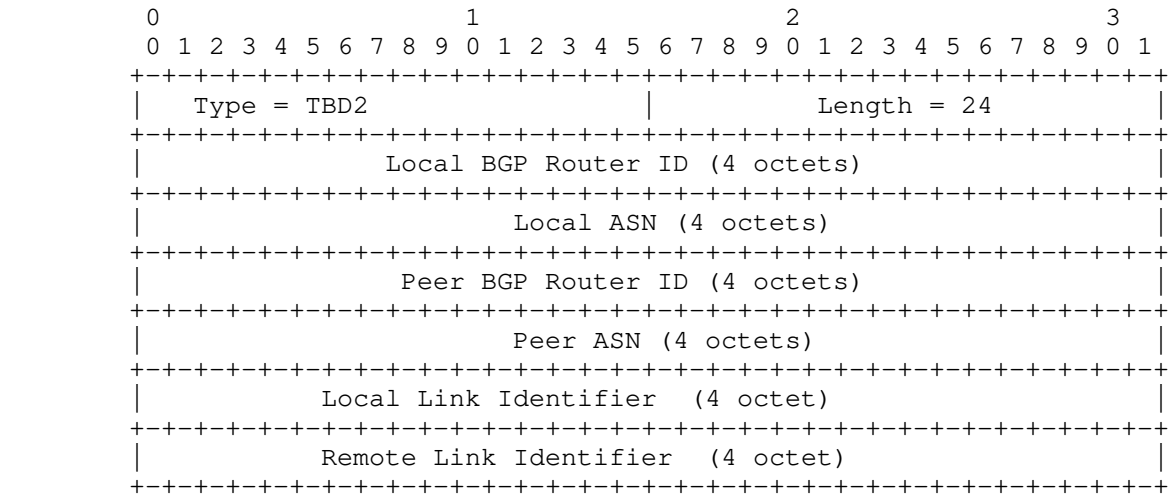
Set to the address used by the local node for BGP session peering. When AF.Type is set to 4, this address is 4-octet IPv4 address and when AF.Type is set to 6, this address is 16-octet IPv6 address.

Remote Interface Address

Set to the address used by the peer node for BGP session peering. When AF.Type is set to 4, this address is 4-octet IPv4 address and when AF.Type is set to 6, this address is 16-octet IPv6 address.

4.3. BGP Peering Segment - Peer-Adj-SID

Peer-Adj-SID identifies the underlying link to the BGP peer node. The sub-TLV format for Peer-Adj-SID of BGP Peering Segment MUST be set as shown in the below TLV format:



Local BGP Router ID

4-octet BGP Router ID of the node that assigns the Peer-Node-SID.

Local ASN

4-octet local ASN number of the node that assigns the Peer-Node-SID.

Peer BGP Router ID

4-octet BGP Router ID of the peer node.

Peer ASN

4-octet ASN number of the peer node.

Local Link Identifier

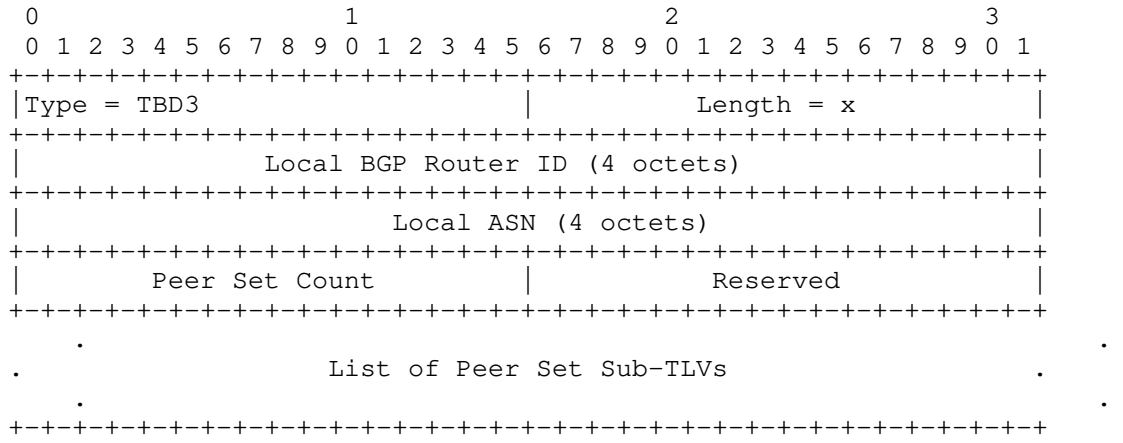
Set to 4-octet link identifier of the local interface to which Peer-Adj-SID is assigned to.

Remote Link Identifier

Set to 4-octet link identifier of the peer interface to which Peer-Adj-SID is assigned to. Set to all-zeros when this identifier is unknown.

4.4. BGP Peering Segment - Peer-Set-SID

The sub-TLV format for Peer-Node-SID of BGP Peering Segment MUST be set as shown in the below TLV format:



Local BGP Router ID

4-octet BGP Router ID of the node that assigns the Peer-Set-SID.

Local ASN

4-octet local ASN number of the node that assigns the Peer-Set-SID.

Peer Set Count

Set to the number of Peer Sub-TLVs included.

Sub-TLV Length

Total length in octets of the sub-TLVs associated with this TLV.

Peer Set Sub-TLV

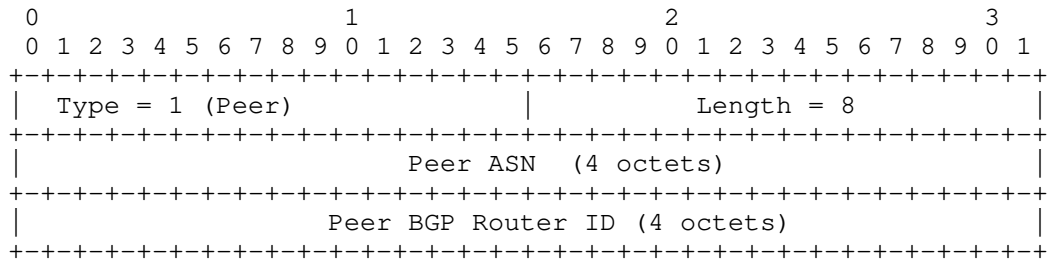
Carries the Sub-TLVs defined in section 4.4.1.

4.4.1. Peer Set Sub-TLV

As defined in section 5.3 of [I-D.ietf-idr-bgpls-segment-routing-epe], Peer-Set-SID can identify the set where the members can be Peer-Node or Peer-Adj from same or different ASN. The format of the Peer Set Sub-TLV will identify each such member.

4.4.1.1. Peer Node

The format for this sub-TLV MUST be set as below:



Peer ASN

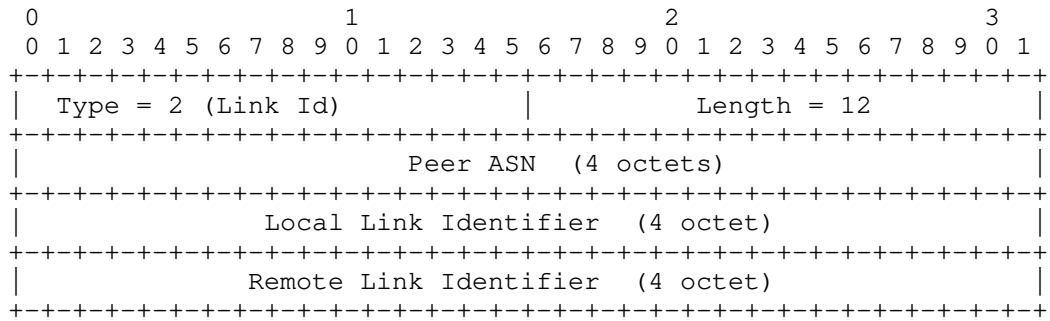
4-octet ASN number of the peer node.

Peer Router ID

4-octet BGP Router ID of the peer node.

4.4.1.2. Link Identifier

The format for this sub-TLV is as below:



Peer ASN

4-octet ASN number of the peer node.

Local Link Identifier

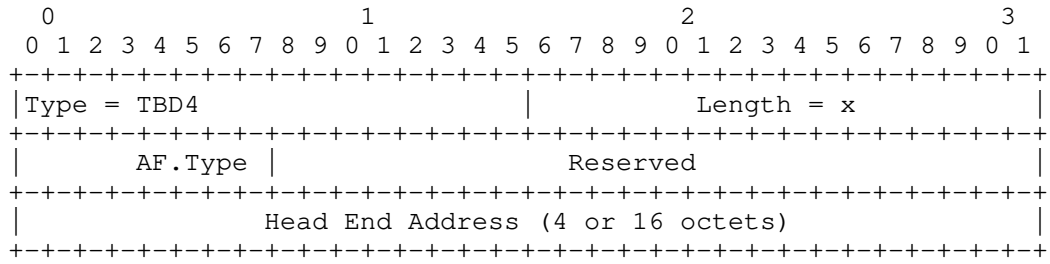
Set to 4-octet link identifier of the local interface to which Peer-Adj-SID is assigned to.

Remote Link Identifier

Set to 4-octet link identifier of the peer interface to which Peer-Adj-SID is assigned to. Set to all-zeros when this identifier is unknown.

4.5. Path Binding SID

Path Binding SID identifies the Binding Segment Identifier associated with an RSVP-TE or SR-TE path. The format for this sub-TLV is as below:



AF.Type

Set to 4 if the address in Head End Address field is IPv4 and set to 6 if the address in Head End address field is IPv6.

Reserved

MUST be set to 0 on send and MUST be ignored on receipt.

Head End Address

Set to the address of the head end node to which the policy is assigned. When AF.Type is 4, this address is IPv4 and when AF.Type is 6, it is IPv6.

4.6. Multicast Replication

[I-D.voyer-spring-sr-p2mp-policy] describes Segment Routing Multicast Replication Policy and introduces the notion of Tree SID to achieve this. A future version of this document will describe LSP Ping and Traceroute Target FEC Stack sub-TLV and procedures for Tree SID validation.

5. Procedures

This section describes the aspects of LSP Ping and Traceroute operations that require further considerations beyond [RFC8029] and [RFC8287].

5.1. BGP Prefix SID

The procedures described in [RFC8029] are sufficient for MPLS Ping and Traceroute operations for BGP Prefix SID using the FEC definitions from Section 3.2.13 and 3.2.14 of [RFC8029].

5.2. BGP Peering Segment Sub-TLVs

BGP Peering Segment sub-TLVs (BGP-Node-SID, BGP-Adj-SID, Peer-Set-SID) are assigned by BGP EPE node for a particular BGP neighbor, and advertised to the peer nodes. Any LSP Ping and Traceroute operation MUST be performed on the BGP EPE node, and not the remote neighbor node, as only the BGP EPE node can validate the contents of BGP Peering Segment sub-TLVs. Additionally, leaking the echo packet to the peer node may not be desirable for network operators.

5.2.1. Initiator Node Procedures

If the bottom-most label in the label stack is BGP Peer Segment label, the initiating node MUST set the TTL of the bottom-most label to 1 to ensure that MPLS TTL expires at the BGP EPE node, and the

echo packet does not leak to the BGP peer node. Echo packet MUST include one of BGP-Node-SID, BGP-Adj-SID, or Peer-Set-SID sub-TLV in the Target FEC Stack TLV corresponding to the BGP Peer Segment label. Operator MAY push one or more transport labels on top of the BGP Peer Segment label to forward the echo packet to the BGP EPE node.

5.2.2. Responder Node Procedures

In addition to procedures defined in [RFC8029], the responding node, upon TTL expiry of the echo packet, MUST process the incoming BGP Peer Segment sub-TLV of the Target FEC Stack. It MUST validate that contents of the sub-TLV and ensure the incoming label is advertised for the processed BGP Peer Segment sub-TLV.

5.3. Path Binding SID

5.3.1. Initiator Node Procedures

Similar to BGP Peering Segment sub-TLVs, Path Binding SID sub-TLV MUST be validated at the node assigning and advertising the Binding SID, instead of the endpoint of the path associated with the Binding SID. The initiating node MUST set the TTL of the Binding SID label to 1 and include the associated Path Binding SID TLV in the Target FEC Stack TLV of the echo request. Operator MAY push one or more transport labels on top of Binding SID label to forward echo packet from initiating node to the assigning node.

5.3.2. Responder Node Procedures

In addition to procedures defined in [RFC8029], the responding node, upon TTL expiry of the echo packet, MUST process the incoming Path Binding SID sub-TLV of the Target FEC Stack. The responding node MUST ensure that it is the advertising node specified in the Path Bindng SID sub-TLV, and the incoming Binding SID label matches the advertised label value.

6. IANA Considerations

To be Updated.

7. Security Considerations

To be Updated

8. Acknowledgement

TBD

9. Contributors

TBD

10. References

10.1. Normative References

[I-D.ietf-idr-bgp-prefix-sid]
Previdi, S., Filsfils, C., Lindem, A., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix SID extensions for BGP", draft-ietf-idr-bgp-prefix-sid-27 (work in progress), June 2018.

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., 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-15 (work in progress), March 2018.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Tantsura, J., Filsfils, C., Previdi, S., Hardwick, J., and D. Dhody, "Carrying Binding Label/ Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-04 (work in progress), March 2018.

[I-D.voyer-spring-sr-p2mp-policy]
daniel.voyer@bell.ca, d., Hassen, C., Gillis, K., Filsfils, C., Parekh, R., and H. Bidgoli, "SR Replication Policy for P2MP Service Delivery", draft-voyer-spring-sr-p2mp-policy-01 (work in progress), October 2018.

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

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <https://www.rfc-editor.org/info/rfc3443>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <https://www.rfc-editor.org/info/rfc4203>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <https://www.rfc-editor.org/info/rfc5307>.

[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>.

10.2. Informative References

[I-D.ietf-isis-segment-routing-extensions] Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-19 (work in progress), July 2018.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions] Psenak, P., Filsfils, C., Previdi, S., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-15 (work in progress), August 2018.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-25 (work in progress), April 2018.

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

[I-D.ietf-spring-segment-routing-ldp-interop]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing interworking with LDP", draft-ietf-spring-segment-routing-ldp-interop-15 (work in progress), September 2018.

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-14 (work in progress), June 2018.

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

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

Authors' Addresses

Nagendra Kumar (editor)
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: naikumar@cisco.com

Carlos Pignataro (editor)
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Faisal Iqbal
Cisco Systems, Inc.

Email: faiqbal@cisco.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com