Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2018

X. Xu
Alibaba
S. Bryant
Huawei
A. Farrel
Juniper
A. Bashandy
Cisco
W. Henderickx
Nokia
Z. Li
Huawei

March 1, 2018

SR-MPLS over IP
draft-xu-mpls-sr-over-ip-00

Abstract

MPLS Segment Routing (SR-MPLS in short) is an MPLS data plane-based source routing paradigm in which the sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels on the packet. SR-MPLS could be leveraged to realize a source routing mechanism across MPLS, IPv4, and IPv6 data planes by using an MPLS label stack as a source routing instruction set while preserving backward compatibility with SR-MPLS.

This document describes how SR-MPLS capable routers and IP-only routers can seamlessly co-exist and interoperate through the use of SR-MPLS label stacks and IP encapsulation/tunnelling such as MPLS-in-UDP [RFC7510].

Requirements Language

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

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

MPLS Segment Routing (SR-MPLS in short) [I-D.ietf-spring-segment-routing-mpls] is an MPLS data plane-based source routing paradigm in which the sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels on the packet. SR-MPLS could be leveraged to realize a source routing mechanism across MPLS, IPv4, and IPv6 data planes by using an MPLS label stack as a source routing instruction set while preserving backward compatibility with SR-MPLS. More specifically, the source routing instruction set information contained in a source routed packet could be uniformly encoded as an MPLS label stack no matter whether the underlay is IPv4, IPv6, or MPLS.

This document describes how SR-MPLS capable routers and IP-only routers can seamlessly co-exist and interoperate through the use of SR-MPLS label stacks and IP encapsulation/tunnelling such as MPLS-in-UDP [RFC7510].

Although the source routing instructions are encoded as MPLS labels, this is a hardware convenience rather than an indication that the whole MPLS protocol stack needs to be deployed. In particular, the MPLS control protocols are not used in this or any other form of SR-MPLS.

Section 3 describes various use cases for the tunneling SR-MPLS over IP. Section 4 describes a typical application scenario and how the packet forwarding happens. Section 5 describes the forwarding procedures of different elements when UDP encapsulation is adopted for source routing.

2. Terminology

This memo makes use of the terms defined in [RFC3031] and [I-D.ietf-spring-segment-routing-mpls].

3. Use Cases

Tunnelling SR-MPLS using IPv4 and/or IPv6 tunnels is useful at least in the following use cases:

- Incremental deployment of the SR-MPLS technology may be facilitated by tunnelling SR-MPLS packets across parts of a network that are not SR-MPLS enabled using an IP tunneling mechanism such as MPLS-in-UDP [RFC7510]. The tunnel destination address is the address of the next SR-MPLS-capable node along the path (i.e., the egress of the active node segment). This is shown in Figure 1.

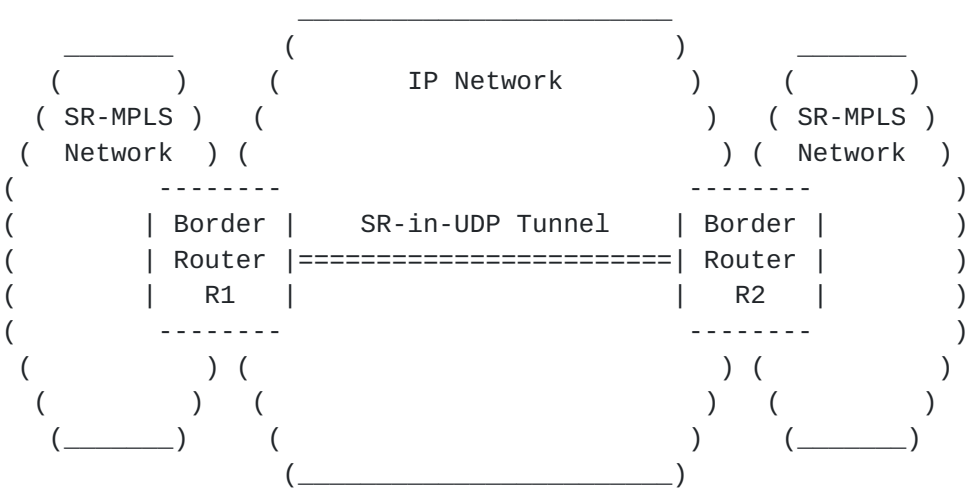
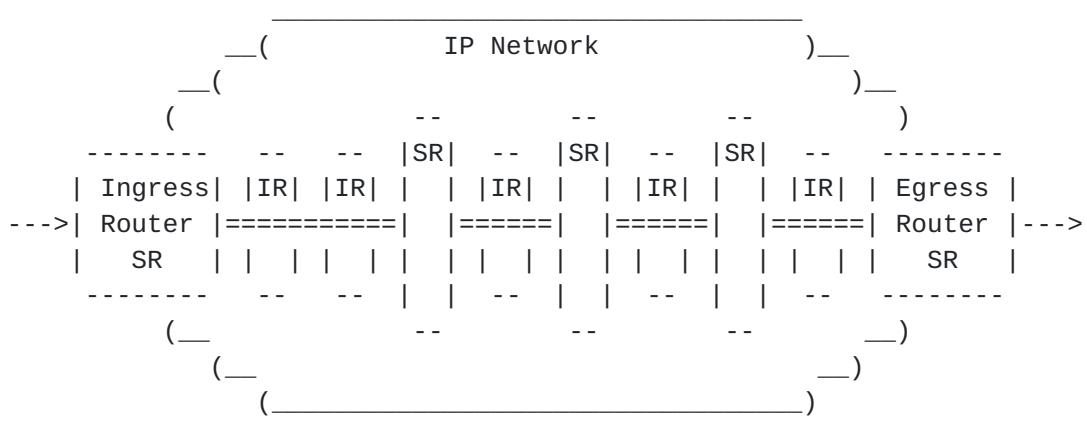


Figure 1: SR-MPLS in UDP to Tunnel Between SR-MPLS Sites

- If encoding of entropy is desired, IP tunneling mechanims that allow encoding of entrpopy, such as MPLS-in-UDP encapsulation [RFC7510] where the source port of the UDP header is used as an entropy field, may be used to maximize the untilization of ECMP and/or UCMP, specially when it is difficult to make use of entropy label mechanism. Refer to [I-D.ietf-mpls-spring-entropy-label]) for more discussion about using entropy label in SR-MPLS.

- Tunneling MPLS into IP provides a transition technology that enables SR in an IPv4 and/or IPv6 network where many routers have not yet been upgraded to have SRv6 capabilities [I-D.ietf-6man-segment-routing-header]. It could be deployed as an interim until full featured SRv6 is available on more platforms. This is shown in Figure 2.



Key:
IR : IP-only Router
SR : SR-MPLS-capable Router
== : SR-MPLS in UDP Tunnel

Figure 2: SR-MPLS Enabled Within an IP Network

4. Procedures of SR-MPLS over IP

This section describes the construction of forwarding information base (FIB) entries and the forwarding behavior that allow the deployment of SR-MPLS when some routers in the network are IP only (i.e., do not support SR-MPLS). Note that the examples described in Section 4.1 and Section 4.2 assume that OSPF or ISIS is enabled: in fact, other mechanisms of discovery and advertisement could be used including other routing protocols (such as BGP) or a central controller.

4.1. Forwarding Entry Construction

This sub-section describes the how to construct the forwarding information base (FIB) entry on an SR-MPLS-capable router when some or all of the next-hops along the shortest path towards a prefix-SID are IP-only routers.

Consider router A that receives a labeled packet with top label L(E) that corresponds to the prefix-SID SID(E) of prefix P(E) advertised by router E. Suppose the ith next-hop router (termed NHi) along the shortest path from router A toward SID(E) is not SR-MPLS capable. That is both routers A and E are SR-MPLS capable, but some router NHi along the shortest path from A to E is not SR-MPLS capable. The following processing steps apply:

- Router E is SR-MPLS capable so it advertises the SR-Capabilities sub-TLV including the SRGB as described in [I-D.ietf-ospf-segment-routing-extensions] and [I-D.ietf-isis-segment-routing-extensions].

- Router E advertises the prefix-SID SID(E) of prefix P(E) so MUST also advertise the encapsulation endpoint and the tunnel type of any tunnel used to reach E. It does this using the mechanisms described in [I-D.ietf-isis-encapsulation-cap] or [I-D.ietf-ospf-encapsulation-cap].

- If A and E are in different IGP areas/levels, then:

  * The OSPF Tunnel Encapsulation TLV [I-D.ietf-ospf-encapsulation-cap] or the ISIS Tunnel Encapsulation sub-TLV [I-D.ietf-isis-encapsulation-cap] is flooded domain-wide.

  * The OSPF SID/label range TLV [I-D.ietf-ospf-segment-routing-extensions] or the ISIS SR-Capabilities Sub-TLV [I-D.ietf-isis-segment-routing-extensions] is advertised domain-wide. This way router A knows the characteristics of the router that originated the advertisement of SID(E) (i.e., router E).

  * When router E advertises the prefix P(E):

    + If router E is running ISIS it uses the extended reachability TLV (TLVs 135, 235, 236, 237) and associates the IPv4/IPv6 or IPv4/IPv6 source router ID sub-TLV(s) [RFC7794].

    + If router E is running OSPF it uses the OSPFv2 Extended Prefix Opaque LSA [RFC7684] and sets the flooding scope to AS-wide.

  * If router E is running ISIS and advertises the ISIS capabilities TLV (TLV 242) [RFC7981], it MUST set the "router-ID" field to a valid value or include an IPV6 TE router-ID sub-TLV (TLV 12), or do both. The "S" bit (flooding scope) of the ISIS capabilities TLV (TLV 242) MUST be set to "1" .

- Router A programs the FIB entry for prefix P(E) corresponding to the SID(E) as follows:

  * If the NP flag in OSPF or the P flag in ISIS is clear:

    pop the top label

* If the NP flag in OSPF or the P flag in ISIS is set:

  swap the top label to a value equal to SID(E) plus the lower bound of the SRGB of E

* Encapsulate the packet according to the encapsulation advertised in [I-D.ietf-isis-encapsulation-cap] or [I-D.ietf-ospf-encapsulation-cap]

* Send the packet towards the next hop NHi.

4.2. Packet Forwarding Procedures

4.2.1. Packet Forwarding with Penultimate Hop Popping

The description in this section assumes that the label associated with each prefix-SID is advertised by the owner of the prefix-SID is a Penultimate Hop Popping (PHP) label. That is, the NP flag in OSPF or the P flag in ISIS associated with the prefix SID is not set.

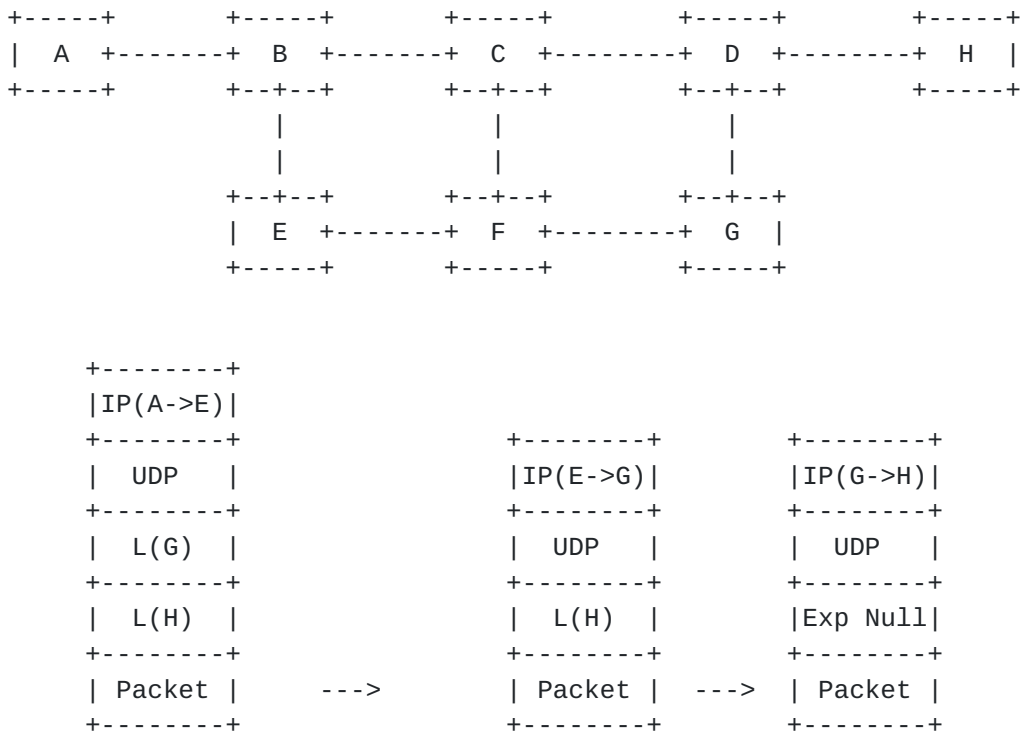


Figure 3: Packet Forwarding Example with PHP

In the example shown in Figure 3, assume that routers A, E, G, and H are SR-MPLS-capable while the remaining routers (B, C, D, and F) are

only capable of forwarding IP packets. Routers A, E, G, and H advertise their Segment Routing related information via IS-IS or OSPF.

Now assume that router A wants to send a packet via the explicit path {E->G->H}. Router A will impose an MPLS label stack corresponding to that explicit path on the packet. Since the next hop toward router E is only IP-capable, router A replaces the top label (that indicated router E) with a UDP-based tunnel for MPLS (i.e., MPLS-over-UDP [RFC7510]) to router E and then sends the packet. In other words, router A pops the top label and then encapsulates the MPLS packet in a UDP tunnel to router E.

When the IP-encapsulated MPLS packet arrives at router E, router E strips the IP-based tunnel header and then process the decapsulated MPLS packet. The top label indicates that the packet must be forwarded toward router G. Since the next hop toward router G is only IP-capable, router E replaces the current top label with an MPLS-over-UDP tunnel toward router G and sends it out. That is, router E pops the top label and then encapsulates the MPLS packet in a UDP tunnel to router G.

When the packet arrives at router G, router G will strip the IP-based tunnel header and then process the decapsulated MPLS packet. The top label indicates that the packet must be forwarded toward router H. Since the next hop toward router H is only IP-capable, router G would replace the current top label with an MPLS-over-UDP tunnel toward router H and send it out. However, this would leave the original packet that router A wanted to send to router H encapsulated in UDP as if it was MPLS even though the original packet could have been any protocol. That is, the final SR-MPLS has been popped exposing the payload packet.

To handle this, when a router (here it is router G) pops the final SR-MPLS label, it inserts an explicit null label [RFC3032] before encapsulating the packet with an MPLS-over-UDP tunnel toward router H and sending it out. That is, router G pops the top label, discovers it has reached the bottom of stack, pushes an explicit null label, and then encapsulates the MPLS packet in a UDP tunnel to router H.

4.2.2. Packet Forwarding without Penultimate Hop Popping

Figure 4 demonstrates the packet walk in the case where the label associated with each prefix-SID advertised by the owner of the prefix-SID is not a Penultimate Hop Popping (PHP) label (i.e., the the NP flag in OSPF or the P flag in ISIS associated with the prefix SID is set).

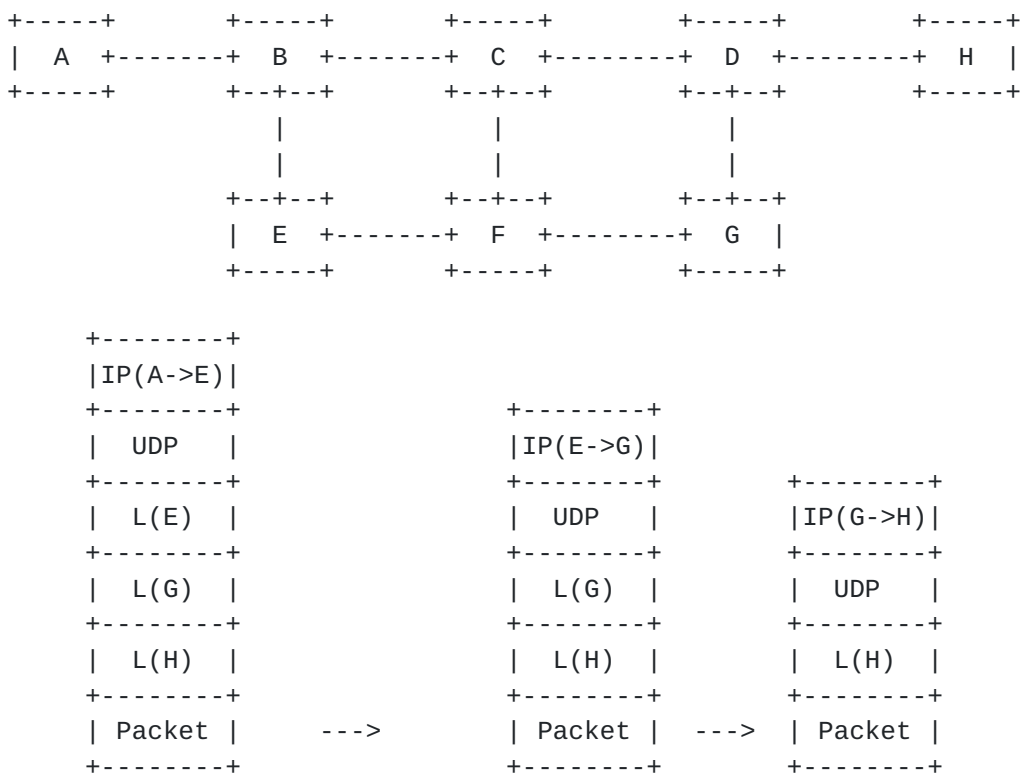


Figure 4: Packet Forwarding Example without PHP

As can be seen from the figure, the SR-MPLS label for each segment is left in place until the end of the segment where it is popped and the next instruction is processed. Further description can be found in Section 5.

4.2.3. Additional Forwarding Procedures

Although the description in the previous two sections is based on the use of prefix-SIDs, tunneling SR-MPLS packets are useful when the top label of a received SR-MPLS packet indicates an adjacncy-SID and the corresponding adjacent node to that adjacency-SID is not capable of MPLS forwarding but can still process SR-MPLS packets. In this scenario the top label would be replaced by an IP tunnel toward that adjacent node and then forwarded over the corresponding link indicated by the adjacency-SID.

When encapsulating an MPLS packet with an IP tunnel header that is capable of encoding entropy (such as [RFC7510]), the corresponding entropy field (the source port in case UDP tunnel) MAY be filled with an entropy value that is generated by the encapsulator to uniquely identify a flow. However, what constitutes a flow is locally determined by the encapsulator. For instance, if the MPLS label

stack contains at least one entropy label and the encapsulator is capable of reading that entropy label, the entropy label value could be directly copied to the source port of the UDP header. Otherwise, the encapsulator may have to perform a hash on the whole label stack or the five-tuple of the SR-MPLS payload if the payload is determined as an IP packet. To avoid re-performing the hash or hunting for the entropy label each time the packet is encapsulated in a UDP tunnel it MAY be desireable that the entropy value contained in the incoming packet (i.e., the UDP source port value) is retained when stripping the UDP header and is re-used as the entropy value of the outgoing packet.

5. Forwarding Details of SR-MPLS over UDP

This section provides supplementary details to the description found in Section 4.

[RFC7510] specifies an IP-based encapsulation for MPLS, i.e., MPLS-in-UDP, which is applicable in some circumstances where IP-based encapsulation for MPLS is required and further fine-grained load balancing of MPLS packets over IP networks over Equal-Cost Multipath (ECMP) and/or Link Aggregation Groups (LAGs) is required as well. This section provides details about the forwarding procedure when when UDP encapsulation is adopted for SR-MPLS over IP.

Nodes that are SR capable can process SR-MPLS packets. Not all of the nodes in an SR domain are SR capable. Some nodes may be "legacy routers" that cannot handle SR packets but can forward IP packets. An SR capable node may advertise its capabilities using the IGP as described in Section 4. There are six types of node in an SR domain:

- Domain ingress nodes that receive packets and encapsulate them for transmission across the domain. Those packets may be any payload protocol including native IP packets or packets that are already MPLS encapsulated.

- Legacy transit nodes that are IP routers but that are not SR capable (i.e., are not able to perform segment routing).

- Transit nodes that are SR capable but that are not identified by a SID in the SID stack.

- Transit nodes that are SR capable and need to perform SR routing because they are identified by a SID in the SID stack.

- The penultimate SR capable node on the path that processes the last SID on the stack on behalf of the domain egress node.

- The domain egress node that forwards the payload packet for ultimate delivery.

The following sub-sections describe the processing behavior in each case.

5.1. Domain Ingress Nodes

Domain ingress nodes receive packets from outside the domain and encapsulate them to be forwarded across the domain. Received packets may already be SR-MPLS packets (in the case of connecting two SR-MPLS networks across a native IP network), or may be native IP or MPLS packets.

In the latter case, the packet is classified by the domain ingress node and an SR-MPLS stack is imposed. In the former case the SR-MPLS stack is already in the packet. The top entry in the stack is popped from the stack and retained for use below.

The packet is then encapsulated in UDP with the destination port set to 6635 to indicate "MPLS-UDP" or to 6636 to indicate "MPLS-UDP-DTLS" as described in [RFC7510]. The source UDP port is set randomly or to provide entropy as described in [RFC7510] and Section 4.2.3, above.

The packet is then encapsulated in IP for transmission across the network. The IP source address is set to the domain ingress node, and the destination address is set to the address corresponding to the label that was previously popped from the stack.

This processing is equivalent to sending the packet out of a virtual interface that corresponds to a virtual link between the ingress node and the next hop SR node realized by a UDP tunnel. The packet is then sent into the IP network and is routed according to the local FIB and applying hashing to resolve any ECMP choices.

5.2. Legacy Transit Nodes

A legacy transit node is an IP router that has no SR capabilities. When such a router receives an SR-MPLS-in-UDP packet it will carry out normal TTL processing and if the packet is still live it will forward it as it would any other UDP-in-IP packet. The packet will be routed toward the destination indicated in the packet header using the local FIB and applying hashing to resolve any ECMP choices.

If the packet is mistakenly addressed to the legacy router, the UDP tunnel will be terminated and the packet will be discarded either because the MPLS-in-UDP port is not supported or because the

uncovered top label has not been allocated. This is, however, a misconnection and should not occur unless there is a routing error.

5.3. On-Path Pass-Through SR Nodes

Just because a node is SR capable and receives an SR-MPLS-in-UDP packet does not mean that it performs SR processing on the packet. Only routers identified by SIDs in the SR stack need to do such processing.

Routers that are not addressed by the destination address in the IP header simply treat the packet as a normal UDP-in-IP packet carrying out normal TTL processing and if the packet is still live routing the packet according to the local FIB and applying hashing to resolve any ECMP choices.

This is important because it means that the SR stack can be kept relatively small and the packet can be steered through the network using shortest path first routing between selected SR nodes.

5.4. SR Transit Nodes

An SR capable node that is addressed by the top most SID in the stack when that is not the last SID in the stack (i.e., the S bit is not set) is an SR transit node. When an SR transit node receives an SR-MPLS-in-UDP packet that is addressed to it, it acts as follows.

- Perform TTL processing as normal for an IP packet.
- Determine that the packet is addressed to the local node.
- Find that the payload is UDP and that the destination port indicates MPLS-in-UDP.
- Strip the IP and UDP headers.
- Examine the label at the top of the stack and process according to the FIB entry (see Section 4.1.
  - If the top label identifies this node then no PHP was used on the incoming segment and the label is popped. Continue the processing with the new top label.
  - Retain the value of the top label.
  - If the top label was advertised requesting PHP, pop the label. (Note that the case where this is the last label in the stack is covered in Section 5.5.)

- Encapsulate the packet in UDP with the destination port set to 6635 (or 6636 for DTLS) and the source port set for entropy. The entropy value SHOULD be retained from the received UDP header or MAY be freshly generated since this is a new UDP tunnel (see Section 4.2.3).

- Encapsulate the packet in IP with the IP source address set to this transit router, and the destination address set to the address corresponding to the SID for the label value retained earlier.

- Send the packet into the IP network routing the packet according to the local FIB and applying hashing to resolve any ECMP choices.

5.5. Penultimate SR Transit Nodes

The penultimate SR transit node is an SR transit node as described in Section 5.4 where the top label is the last label on the stack. When a penultimate SR transit node receives an SR-MPLS-in-UDP packet that is addressed to it, it processes as for any other transit node (see Section 5.4) except for a special case if PHP is supported for the final SID.

If PHP is allowed for the final SID the penultimate SR transit node acts as follows:

- Perform TTL processing as normal for an IP packet.

- Determine that the packet is addressed to the local node.

- Find that the payload is UDP and that the destination port indicates MPLS-in-UDP.

- Strip the IP and UDP headers.

- Examine the label at the top of the stack and process according to the FIB entry (see Section 4.1.

  * If the top label identifies this node then no PHP was used on the incoming segment and the label is popped. Continue the processing with the new top label.

  * Retain the value of the top label.

  * If the top label was advertised requesting PHP, pop the label. This will have been the last label in the stack. Push an explicit null label [RFC3032] (0 for IPv4 and 2 for IPv6) with bottom of stack (S bit) set.

- Encapsulate the packet in UDP with the destination port set to 6635 (or 6636 for DTLS) and the source port set for entropy. The entropy value SHOULD be retained from the received UDP header or MAY be freshly generated since this is a new UDP tunnel.

- Encapsulate the packet in IP with the IP source address set to this transit router, and the destination address set to the domain egress node IP address corresponding to the SID for the label value retained earlier.

- Send the packet into the IP network routing the packet according to the local FIB and applying hashing to resolve any ECMP choices.

5.5.1. A Note on Segment Routing Paths and Penultimate Hop Popping

End-to-end SR paths are comprised of multiple segments. The end point of each segment is identified by a SID in the SID stack. In normal SR processing a penultimate hop is the router that performs SR routing immediately prior to the end-of-segment router. PHP applies at the penultimate router in a segment.

With SR-MPLS-in-UDP encapsulation, each SR segment is achieved using an MPLS-in-UDP tunnel that runs the full length of the segment. The SR SID stack on a packet is only examined at the head and tail ends of this segment. Thus, each segment is effectively one hop long in the SR overlay network and if there is any PHP processing it takes place at the head-end of the segment.

5.6. Domain Egress Nodes

The domain egress acts as follows:

- Perform TTL processing as normal for an IP packet.

- Determine that the packet is addressed to the local node.

- Find that the payload is UDP and that the destination port indicates MPLS-in-UDP.

- Strip the IP and UDP headers.

- Examine the label at the top of the stack and process according to the FIB entry (see Section 4.1.

  * If the top label identifies this node then no PHP was used on the incoming segment and the label is popped. Continue the processing with the new top label.

    * If there is another label it should be the explicit null. Pop it but retain its value.

- o Forward the payload packet according to its type (as potentially indicated by the value of the popped explicit null label) and the local routing/forwarding mechanisms.

6. Contributors

Clarence Filsfils
Cisco
Email: cfilsfil@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

Shaowen Ma
Juniper
Email: mashao@juniper.net

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

Hamid Assarpour
Broadcom
Email:hamid.assarpour@broadcom.com

Robert Raszuk
Bloomberg LP
Email: robert@raszuk.net

Uma Chunduri
Huawei
Email: uma.chunduri@gmail.com

Luis M. Contreras
Telefonica I+D
Email: luismiguel.contrerasmurillo@telefonica.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

Gunter Van De Velde
Nokia
Email: gunter.van_de_velde@nokia.com

Tal Mizrahi
Marvell
Email: talmi@marvell.com

Jeff Tantsura
Individual
Email: jefftant@gmail.com

7. Acknowledgements

Thanks to Joel Halpern, Bruno Decraene, Loa Andersson, Ron Bonica, Eric Rosen, Jim Guichard, and Gunter Van De Velde for their insightful comments on this draft.

8. IANA Considerations

No IANA action is required.

9. Security Considerations

TBD.

10. References

10.1. Normative References

[I-D.ietf-isis-encapsulation-cap]
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in IS-IS", draft-ietf-isis-encapsulation-cap-01 (work in progress), April 2017.

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

[I-D.ietf-ospf-encapsulation-cap]
Xu, X., Decraene, B., Raszuk, R., Contreras, L., and L. Jalil, "The Tunnel Encapsulations OSPF Router Information", draft-ietf-ospf-encapsulation-cap-09 (work in progress), October 2017.

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

[I-D.ietf-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.

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

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

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

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

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

[RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794, March 2016, <https://www.rfc-editor.org/info/rfc7794>.

[RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <https://www.rfc-editor.org/info/rfc7981>.

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

10.2. Informative References

[I-D.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-08 (work in progress), January 2018.

[I-D.ietf-mpls-spring-entropy-label]
Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and J. Tantsura, "Entropy label for SPRING tunnels", draft-ietf-mpls-spring-entropy-label-08 (work in progress), January 2018.

Authors' Addresses

Xiaohu Xu
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Adrian Farrel
Juniper

Email: afarrel@juniper.net

Ahmed Bashandy
Cisco

Email: bashandy@cisco.com

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

Zhenbin Li
Huawei

Email: lizhenbin@huawei.com