MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

S. Bryant, Ed.
Huawei
A. Farrel, Ed.
J. Drake
Juniper Networks
J. Tantsura
Individual

October 30, 2017

MPLS Segment Routing in IP Networks
draft-bryant-mpls-unified-ip-sr-03

Abstract

Segment routing is a source routed forwarding method that allows packets to be steered through a network on paths other than the shortest path derived from the routing protocol. The approach uses information encoded in the packet header to partially or completely specify the route the packet takes through the network, and does not make use of a signaling protocol to pre-install paths in the network.

Two different encapsulations have been defined to enable segment routing in an MPLS network or in an IPv6 network. While acknowledging that there is a strong need to support segment routing in both environments, this document defines a mechanism to carry MPLS segment routing packets encapsulated in UDP. The resulting approach is applicable to both IPv4 and IPv6 networks without the need for any changes to the IP or segment routing specifications.

This document makes no changes to the segment routing architecture and builds on existing protocol mechanisms such as the encapsulation of MPLS within UDP defined in RFC 7510.

No new procedures are introduced, but existing mechanisms are combined to achieve the desired result.

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.

This Internet-Draft will expire on May 3, 2018.

Table of Contents

1. Introduction

Segment routing (SR) [I-D.ietf-spring-segment-routing] is a source routed forwarding method that allows packets to be steered through a network on paths other than the shortest path derived from the routing protocol. SR also allows the packets to be steered through a set of packet processing functions along that path. SR uses information encoded in the packet header to partially or completely specify the route the packet takes through the network and does not make use of a signaling protocol to pre-install paths in the network.

The approach to segment routing in IPv6 networks is known as SRv6 and is described in [I-D.ietf-6man-segment-routing-header]. The mechanism described encodes the segment routing instruction list as an ordered list of 128-bit IPv6 addresses that is carried in a new IPv6 extension header: the Source Routing Header (SRH).

MPLS Segment Routing (MPLS-SR) [I-D.ietf-spring-segment-routing-mpls] encodes the route the packet takes through the network and the instructions to be applied to the packet as it transits the network by imposing a stack of MPLS label stack entries on the packet.

This document describes a method for running SR in IPv4 or IPv6 networks by using an MPLS-SR label stack carried in UDP. No change is made to the MPLS-SR encoding mechanism as described in [I-D.ietf-spring-segment-routing-mpls] where a sequence of 32 bit units, one for each instruction, called the Segment Routing Instruction Stack (SRIS) is used. Each basic unit is encoded as an MPLS label stack entry and the segment routing instructions (i.e., the Segment Identifiers, SIDs) are encoded in the 20 bit MPLS Label fields.

In summary, the processing described in this document is a combination of normal MPLS-over-UDP behavior as described in [RFC7510], MPLS-SR lookup and label-pop behavior as described in [I-D.ietf-spring-segment-routing-mpls], and normal IP forwarding. No new procedures are introduced, but existing mechanisms are combined to achieve the desired result.

The method defined is a complementary way of running SR in an IP network that can be used alongside or interchangeably with that defined in [I-D.ietf-6man-segment-routing-header]. Implementers and deployers should consider the benefits and drawbacks of each method and select the approach most suited to their needs.

2. The MPLS-SR-over-UDP Encoding Stack

The MPLS-SR-over-UDP encoding stack is shown in Figure 1.

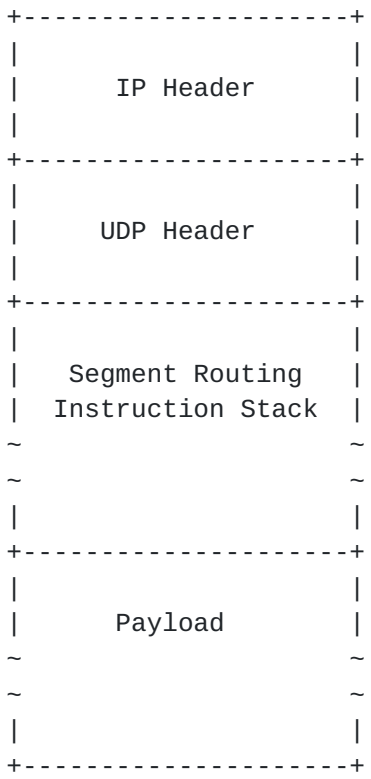


Figure 1: Packet Encapsulation

The payload may be of any type that, with an appropriate convergence layer, can be carried over a packet network. It is anticipated that the most common packet types will be IPv4, IPv6, native MPLS, and pseudowires [RFC3985].

Preceding the Payload is the Segment Routing Instruction Stack (SRIS) that carries the sequence of instructions to be executed on the packet as it traverses the network. This is the Segment Identifier (SID) stack that is the ordered list of segments described in [I-D.ietf-spring-segment-routing].

Preceding the SRIS is a UDP header. The UDP header is included to:

- Introduce entropy to allow equal-cost multi-path load balancing (ECMP) [RFC2992] in the IP layer [RFC7510].
- Provide a protocol multiplexing layer as an alternative to using a new IP type/next header.
- Allow transit through firewalls and other middleboxes.
- Provide disaggregation.

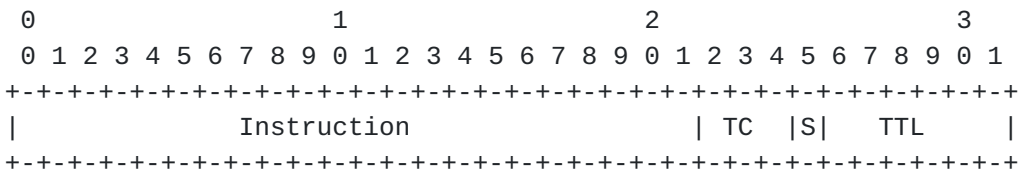
Preceding the UDP header is the IP header which may be IPv4 or IPv6.

3. The Segment Routing Instruction Stack

The Segment Routing Instruction Stack (SRIS) consists of a sequence of Segment Identifiers (SIDs) as described in [I-D.ietf-spring-segment-routing] encoded as an MPLS label stack as described in [I-D.ietf-spring-segment-routing-mpls].

The top SRIS entry is the next instruction to be executed. When the node to which this instruction is directed has processed the instruction it is removed (popped) from the SRIS, and the next instruction is processed.

Each instruction is encoded in a single Label Stack Entry (LSE) as shown in Figure 2. The structure of the LSE is unchanged from [RFC3032].



Instruction: Label Value, 20 bits
TC: Traffic Class, 3 bits
S: Bottom of Stack, 1 bit
TTL: Time to Live, 8 bits

Figure 2: SRIS Label Stack Entry

As with [I-D.ietf-spring-segment-routing-mpls] a 32 bit LSE is used to carry each SR instruction. The instruction itself is carried in the 20 bit Label Value field. The TC field has the normal meaning as

defined in [RFC3032] and modified in [RFC5462]. The S bit has bottom of stack semantics defined in [RFC3032]. TTL is discussed in Section 3.1.

3.1. TTL

The setting of the TTL is application specific, but the following operational consideration should be born in mind. In SR the size of the label stack may be increased within a single routing domain by various operations such as the pushing of a Binding SID. Furthermore, in SR packets are not necessarily constrained to travel on the shortest path within a routing domain. Therefore, considration has to be given to the possibility that there may be a forwarding loop. To mitigate against this it is RECOMMENDED that the TTL is decremented at each hop as the packet passes through the SR network regardless of any other changes to the network layer encapsulation.

Further discussion of the use of TTL during tunnelling can be found in [RFC4023].

4. UDP/IP Encapsulation

[RFC7510] specifies the values to be used in the UDP Source Port, Destination Port, and Checksum fields.

An administrative domain, or set of administrative domains that are sufficiently well managed and monitored to be able to safely use IP segment routing is likely to comply with the requirements called out in [RFC7510] to permit operation with a zero UDP checksum over IP. However each operator needs to validate the decision on whether or not to use a UDP checksum for themselves.

The [RFC7510] UDP header may be carried over IPv4 or over IPv6.

The IP source address is the address of the encapsulating device. The IP destination address is implied by the instruction at the top of the instruction stack.

If IPv4 is in use, fragmentation is not permitted.

5. Elements of Procedure

Nodes that are SR capable can process MPLS-SR 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 8. 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.

In summary, the processing is a combination of normal MPLS-over-UDP behavior as described in [RFC7510], MPLS-SR lookup and label-pop behavior as described in [I-D.ietf-spring-segment-routing-mpls], and normal IP forwarding. No new procedures are introduced, but existing mechanisms ae combined to achieve the desired result.

The descriptions in the following sections represent the functional behavior. Optimizations on this behavior may be possible in implementations.

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 MPLS-SR packets (in the case of connecting two MPLS-SR networks across a native IP network), or may be nativee IP or MPLS packets.

In the latter case, the packet is classified by the domain ingress node and an MPLS-SR stack is imposed. In the former case the MPLS-SR 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].

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 MPLS-SR-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 MPLS-SR-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 MPLS-SR-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.
- Pop the top label from the SID stack and retain it for use below.
- 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 address corresponding to the next SID in the stack.
- 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 SID for the node is directly followed by the final SID (i.e., that of domain egress node). When a penultimate SR transit node receives an MPLS-SR-in-UDP packet that is addressed to it, it acts according to whether penultimate hop popping (PHP) is supported for the final SID. That information could be indicated using the control plane as described in Section 8. It is worth making some additional observations about PHP in SR: these are collected in Section 6.

If PHP is allowed 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.
- Pop the top label from the SID stack and retain it for use below.
- Pop the next label from the SID stack.
- 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 label that was previously popped from the stack.
- Send the packet into the IP network routing the packet according to the local FIB and applying hashing to resolve any ECMP choices.

If PHP is not supported, the penultimate SR transit node just acts as a normal SR transit node just as described in Section 5.4. However, the penultimate SR transit node may be required to replace the final SID with an MPLS-SR label stack entry carrying an explicit null label value (0 for IPv4 and 2 for IPv6) before forwarding the packet. This requirement may also be indicated by the control plane as described in Section 8.

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.
- Pop the outermost SID if present (i.e., if PHP was not performed as described in Section 5.5.

- Pop the explicit null label if it is present in the label stack as requested by the domain egress and communicated in the control plane as described in Section 8.

- Forward the payload packet according to its type and the local routing/forwarding mechanisms.

6. 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. Penultimate hop popping (PHP) is processing that applies at the penultimate router in a segment.

With MPLS-SR-in-UDP encapsulation, each SR segment is achieved using 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 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.

However, in order to simplify processing at each MPLS-SR-in-UDP end point, it is RECOMMENDED that PHP processing is only used for the final segment in an SR path as described in Section 5.5.

7. Modes of Deployment

As previously noted, the procedures described in this document may be used to connect islands of SR functionality across an IP backbone, or can provide SR function within a native IP network. This section briefly expounds upon those two deployment modes.

7.1. Interconnection of SR Domains

Figure 3 shows two SR domains interconnected by an IP network. The procedures described in this document are deployed at border routers R1 and R2 and packets are carried across the backbone network in a UDP tunnel.

R1 acts as the domain ingress as described in Section 5.1. It takes the MPLS-SR packet from the SR domain, pops the top label and uses it to identify its peer border router R2. R1 then encapsulates the packet in UDP in IP and sends it toward R2.

Routers within the IP network simply forward the packet using normal IP routing.

R2 acts as a domain egress router as described in Section 5.6. It receives a packet that is addressed to it, strips the IP and UDP headers, and acts on the payload SR label stack to continue to route the packet.

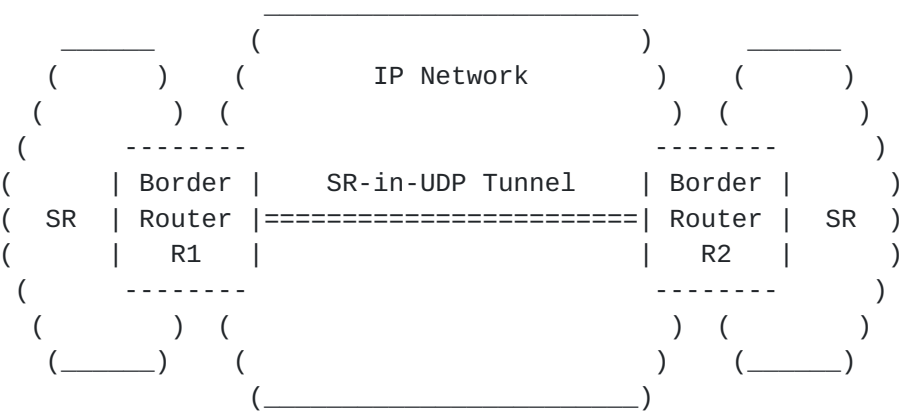


Figure 3: SR in UDP to Tunnel Between SR Sites

7.2. SR Within an IP Network

Figure 4 shows the procedures defined in this document to provide SR function across an IP network.

R1 receives a native packet and classifies it, determining that it should be sent on the SR path R2-R3-R4-R5. It imposes a label stack accordingly and then acts as a domain ingress as described in Section 5.1. It pops the label for R2, and encapsulates the packet in UDP in IP, sets the IP source to R1 and the IP destination to R2, and sends the packet into the IP network.

Routers Ra and Rb are transit routers that simply forward the packets using normal IP forwarding. They may be legacy transit routers (see Section 5.2) or on-path pass-through SR nodes (see Section 5.3).

R2 is an SR transit nodes as described in Section 5.4. It receives a packet addressed to it, strips the IP and UDP headers, and processes the SR label stack. It pops the top label and uses it to identify the next SR hop which is R3. R2 then encapsulates the packet in UDP in IP setting the IP source to R2 and the IP destination to R3.

Rc, Rd, and Re are transit routers and perform as Ra and Rb.

R3 is an SR transit node and performs as R2.

R4 is a penultimate SR transit node as described in Section 5.5. It receives a packet addressed to it, strips the IP and UDP headers, and processes the SR label stack. It pops the top label and uses it to identify the next SR hop which is R5.

R5 is the domain egress as described in Section 5.6. It receives a packet addressed to it, strips the IP and UDP headers.

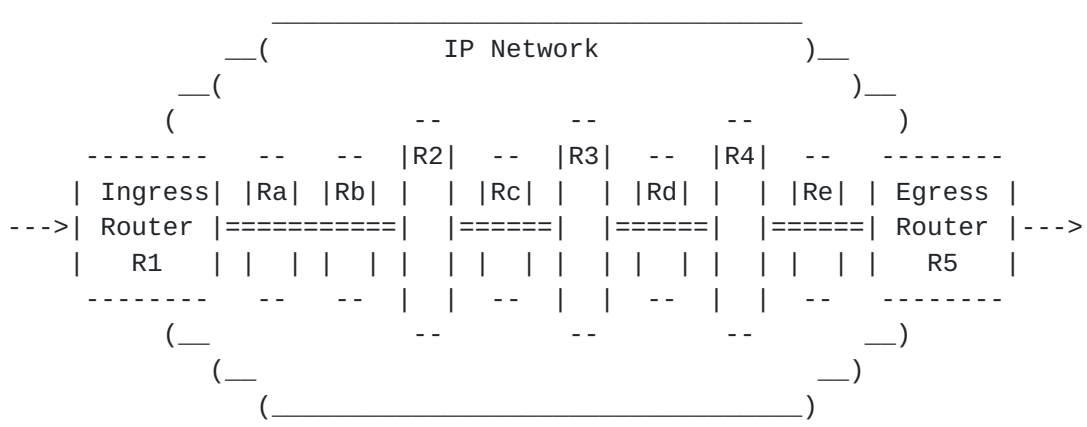


Figure 4: SR Within an IP Network

8. Control Plane

This document is concerned with forwarding plane issues, and a description of applicable control plane mechanisms is out of scope. This section is provided only as a collection of references. No changes to the control plane mechanisms for MPLS-SR are needed or proposed.

A routers that is able to support SR can advertise the fact in the IGP as follows:

- In IS-IS, by using the SR-Capabilities TLV as defined in [I-D.ietf-isis-segment-routing-extensions]

- In OSPF/OSPFv3 by using the Router Information LSA as defined in [I-D.ietf-ospf-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

Nodes can advertise SIDs using the mechanisms defined in [I-D.ietf-isis-segment-routing-extensions], [I-D.ietf-ospf-segment-routing-extensions], or [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

Support for PHP can be indicated in a SID advertisement using flags in the advertisements as follows:

- For IS-IS, the N (no-PHP) flag in the Prefix-SID sub-TLV indicates whether PHP is not to be used.

- For OSPF/OSPFv3, the NP (no-PHP) flag in the Prefix SID Sub-TLV indicates whether PHP is not to be used.

The requirement to use an explicit null SID if PHP is not in use can be indicated in SID advertisement using the Explicit-Null Flag (E-Flag). If set, the penultimate SR transit node replaces the final SID with a SID containing an Explicit-NULL value (0 for IPv4 and 2 for IPv6) before forwarding the packet.

The method of advertising the tunnel encapsulation capability of a router using IS-IS or OSPF are specified in [I-D.ietf-isis-encapsulation-cap] and [I-D.ietf-ospf-encapsulation-cap] respectively. No changes to those procedures are needed in support of this work.

9. OAM

OAM at the payload layer follows the normal OAM procedures for the payload. To the payload the whole SR network looks like a tunnel.

OAM in the IP domain follows the normal IP procedures. This can only be carried out between on the IP hops between pairs of SR nodes.

OAM between instruction processing entities i.e., at the SR layer uses the procedures documented for MPLS.

10. Security Considerations

The security consideration of [I-D.ietf-spring-ipv6-use-cases] and [RFC7510] apply. DTLS [RFC6347] SHOULD be used where security is needed on an MPLS-SR-over-UDP segment.

It is difficult for an attacker to pass a raw MPLS encoded packet into a network and operators have considerable experience at excluding such packets at the network boundaries.

It is easy for an ingress node to detect any attempt to smuggle IP packet into the network since it would see that the UDP destination port was set to MPLS. SR packets not having a destination address terminating in the network would be transparently carried and would pose no security risk to the network under consideration.

11. IANA Considerations

This document makes no IANA requests.

12. Acknowledgements

This draft was partly inspired by [I-D.xu-mpls-unified-source-routing-instruction], and we acknowledge the following authors of version -02 of that draft: Robert Raszuk, Uma Chunduri, Luis M. Contreras, Luay Jalil, Hamid Assarpour, Gunter Van De Velde, Jeff Tantsura, and Shaowen Ma.

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

13. Contributors

o Mach Chen, Huawei Technologies, mach.chen@huawei.com

14. References

14.1. 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-13 (work in progress), October 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[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>.

[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>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[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>.

14.2. Informative References

[I-D.ietf-6man-segment-routing-header] Previdi, S., Filsfils, C., Raza, K., 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-07 (work in progress), July 2017.

[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., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and j. jefftant@gmail.com, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-13 (work in progress), June 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-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-10 (work in progress), September 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-21 (work in progress), October 2017.

[I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Filsfils, C., Maglione, R., and M. Townsley, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-11 (work in progress), June 2017.

[I-D.xu-mpls-unified-source-routing-instruction]
Xu, X., Bashandy, A., Assarpour, H., Ma, S., Henderickx, W., and j. jefftant@gmail.com, "Unified Source Routing Instructions using MPLS Label Stack", draft-xu-mpls-unified-source-routing-instruction-04 (work in progress), September 2017.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <https://www.rfc-editor.org/info/rfc3985>.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <https://www.rfc-editor.org/info/rfc4023>.

Authors' Addresses

Stewart Bryant (editor)
Huawei

Email: stewart.bryant@gmail.com

Adrian Farrel (editor)
Juniper Networks

Email: afarrel@juniper.net

John Drake
Juniper Networks

Email: jdrake@juniper.net

Jeff Tantsura
Individual

Email: jefftant.ietf@gmail.com