SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 April 2026

X. Geng
M. Chen
Huawei Technologies
P. Camarillo, Ed.
Cisco Systems
G. Mishra
Verizon Inc.
B. Varga
F. Fejes
Ericsson

10 October 2025

# SRv6 for Redundancy Protection
## draft-ietf-spring-sr-redundancy-protection-05

## Abstract

Redundancy Protection is a generalized protection mechanism to achieve high reliability of service transmission in Segment Routing networks. The mechanism uses the "Live-Live" methodology. This document introduces one new SRv6 Segment Endpoint Behavior to provide replication and elimination functions on specific network nodes by leveraging SRv6 Network Programming capabilities.

## 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 13 April 2026.

## Copyright Notice

Table of Contents

## 1. Introduction

Redundancy Protection is a generalized protection mechanism to achieve the high reliability of service transmission in a Segment Routing (SR) network. Specifically, packets of flows are replicated at a replication network node into two or more copies, which are transported via different and disjoint paths in parallel. On the elimination network node, the multiple copies are received, redudant packets eliminated, and deliver only a single copy of the packet that is transmitted. This mechanism is commonly refered to as "Live-Live". One new SRv6 Segment Endpoint Behavior are introduced to provide the replication and elimination functions on specific network nodes by leveraging SRv6 Network Programming capabilities. As it is unnecessary to perform switchover of recieving packets between different paths, redundancy protection can facilitate to achieve zero packet loss target when failure on either path happens.

Redundancy protection provides ultra reliable protection to many services, for example Cloud VR/Game, IPTV service and other type of video services, high value private line service etc.

## 2. Terminology

### 2.1. Requirements Language

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

### 2.2. Terminology and Conventions

SR: Segment Routing

SRv6: Segment Routing over IPv6

SID: Segment Identifier

BSID: Binding SID

RSID: Redundancy SID

R-node: Redundancy node participating in the service protection.

Rep node: R-node doing replication. A network element that replicates incoming packets for parallel delivery.

Elm node: R-node doing elimination. A network element that reassembles and elimantes duplicates to forward a single copy.

RedInst: Redundancy instance, flow-specific redundancy function on the R-node.

FID: Flow Identification

SN: Sequence Number

## 3. Redundancy Protection in Segment Routing Scenario

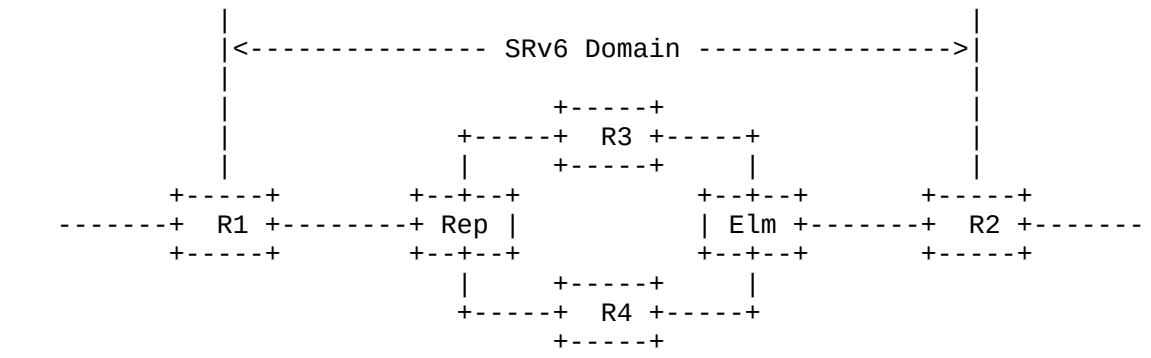


Figure 1: Example topology

Figure 1 shows an example of redundancy protection used in an SRv6 domain. R1, R2, R3, R4, Rep and Elm are SR-capable nodes. Rep and Elm are redundancy nodes. When a flow is sent into the SRv6 domain, the process is:

1) R1 receives the traffic flow and encapsulates packets with a list of segments destined to R2, which is instantiated as an ordered list of SRv6 SIDs.

2) When the packet flow arrives at Rep node (a redundancy node configured for replication), each packet is replicated into two or more copies. Each copy of the packet is encapsulated with a new segment list, which represents different disjoint forwarding paths towards the next R-node. The disjoint path is provisioned by a controller.

3) Meta data information including flow identification (FID) and sequence number (SN) is used to facilitate elimination of duplicate packets on Elimination node (Elm). Flow identification identifies the specific flow, and sequence number distinguishes the packet sequence within a flow. This packet meta data is included on each of the replicas at the redudancy node.

4) The multiple replicas go through different paths until they reach the next redundancy node i.e., the Elm node. The first received copy of each flow packet is transmitted from Elm node to R2, and the redundant packets are eliminated.

5) When there is any failure or packet loss in one path, the service transmission continues through the other path non-disruptively.

6) Sometimes, out-of-order packets may occur since service packets are recovered from different forwarding paths. In this case, the redundancy node or other network nodes behind the redundancy node MAY include a reordering function, which is implementation specific and out of the scope of this document, to guarantee in-order delivery of packets.

To minimize the jitter caused by random packet loss, the disjoint paths are RECOMMENDED to have similar path forwarding delay.

## 4. SRv6 Segment Behavior to Support Redundancy Protection

To achieve the packet replication and elimination functions, the following packet processing rules are defined as a new set of SRv6 SID behaviors regarding the Redundancy SID: (i) End.R, (ii) H.Encaps.R, (iii) H.Encaps.R.Red, (iv) H.Encaps.R.L2, and (v) H.Encaps.R.L2.Red.

Note, that the algorithm used by the Redundancy Functionality is not within the scope of this document.

### 4.1. Redundancy Segment Endpoint Behavior

This section describes the Redundancy specific behaviors that can be associated with a SID.

| | |
|---|---|
| End.R | Endpoint with decapsulation and Redundancy Processing |

Figure 1: Redundancy Endpoint Behavior

Redundancy Segment is the identifier of packets on which service protection need to be executed on the redundancy node. It has associated Redundancy Policy(s), instantiation of which provides service protection action(s). This is similar to the relationship between Binding SID and SR Policy [I-D.ietf-spring-segment-routing-policy], the use of Redundancy Segment triggers the Redundancy Policy instantiation on the redundancy node.

Redundancy Segment is associated with service instructions, indicating the following operations:

* Steers the packet into the corresponding redundancy instance.

* Encapsulates flow identification and sequence number in packets if the two information is not carried in packets.

* Packet replication/elimination and segment encapsulation/ decapsulation based on the information of redundancy policy, e.g., the number of replication copies, an ordered list of segments with a topological instruction.

In this document, a new behavior End.R for Redundancy Segment is defined. An instance of a redundancy SID is associated with a redundancy policy B and a source address A. In the following description, End.R behavior is specified with the encapsulation mode.

For service protection processing, two arguments are needed:

1. Flow-ID (FID): defines which flow the packet belongs to (what is used to determine which Redundancy instance has to be used on a node). (Note: for example DetNet uses 20 bits for FID [RFC8964].

2. Sequence Number (SN): defines the sequencing information, it is created at the first Redundancy node and used by replication and elimination functionalities. (Note: for example, DetNet uses the following SN sizes: 0/16/28 bits [RFC8964].

In order to eliminate the redundant packet of a flow, elimination node utilizes sequence number to evaluate the redundant status of a packet. Note that implementation specific mechanism could be applied to control the amount of state monitored on sequence number, so that system memory usage can be limited at a reasonable level.

As elimination node needs to maintain the state of flows, a centralized controller should have a knowledge of elimination nodes capability, and never provision the redundancy policy to redundancy node when the computation result goes beyond the flow recovery

capability of elimination node. The capability advertisement of elimination node will be specified separately elsewhere, which is not within the scope of this document.

The Redundancy SID (RSID) MUST be the last segment in an SR Policy and it is associated with the Redundancy functionality!

When an SRv6-capable node (N) receives an IPv6 packet whose destination address matches a local IPv6 address instantiated as an SRv6 SID (S), and S is a Redundancy SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.    Send an ICMP Parameter Problem message to the Source Address
             with Code 0 (Erroneous header field encountered),
             and Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet
S04.   }
S05.   Extract the ARG part of the SID
S06.   Remove the outer IPv6 header with all its extension headers
S07.   Forward the exposed payload and the ARG part to the Redundancy
       functionality
S08. }
```

### 4.2. SR Policy Headend Behaviors

This section describes a set of SRv6 Redundancy Policy Headend [RFC8402] behaviors.

| | |
|---|---|
| H.Encaps.R | SR Headend with Redundancy Encapsulation |
| H.Encaps.R.Red | H.Encaps with Reduced Redundancy Encapsulation |
| H.Encaps.R.L2 | H.Encaps.R Applied to Received L2 Frames |
| H.Encaps.R.L2.Red | H.Encaps.R.Red Applied to Received L2 Frames |

Figure 2: Redundancy specific SR Policy Headend Behaviors

#### 4.2.1. H.Encaps.R: SR Headend with Redundancy

When a node "N" receives a packet P=(A, B) identified as a Flow for redundancy. B is neither a local address nor SID of "N". It executes the Flow related Redundancy function(s), resulting in one or more member flow (P1=(A, B), P2=(A, B), ...) with related parameters ([Flow-ID1, SeqNum], [Flow-ID2, SeqNum], ...).

Node "N" is configured with an IPv6 address "T" (e.g., assigned to its loopback). "N" steers the egress packet P1 into an SRv6 Policy with a Source Address T and a segment list SP1=<S11, S12, S13>, where S13 is a Redundancy SID (LOC+FUNCT) with 0 as ARG.

The H.Encaps.R encapsulation behavior is defined as follows (SA: source address, DA: destination address):

S01. Push an IPv6 header with its own SRH
Set the ARG part of the LAST SID in the segment list
S02. Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segment list
S03. Set outer Payload Length, Traffic Class, Hop Limit, and Flow Label fields
S04. Set the outer Next Header value
S05. Decrement inner IPv6 Hop Limit or IPv4 TTL
S06. Submit the packet to the IPv6 module for transmission to S11

After the H.Encaps.R behavior, P1, and P2 respectively look like:

* (T, S11) (S13, S12, S11; SL=2) (A, B), note: S13.ARG=Flow-ID1, SeqNum

* (T, S21) (S23, S22, S21; SL=2) (A, B), note: S23.ARG=Flow-ID2, SeqNum

The member flow packet is encapsulated unmodified (with the exception of the IPv4 TTL or IPv6 Hop Limit that is decremented).

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV. In such cases the outer destination address is the Redundancy SID.

#### 4.2.2. H.Encaps.R.Red: H.Encaps.R with Reduced Encapsulation

The H.Encaps.R.Red behavior is an optimization of the H.Encaps.R behavior.

H.Encaps.R.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only placed in the Destination Address field of the pushed IPv6 header.

After the H.Encaps.R.Red behavior, P1, and P2 respectively look like:

* (T, S11) (S13, S12; SL=2) (A, B), note: S13.ARG=Flow-ID1, SeqNum

* (T, S21) (S23, S22; SL=2) (A, B), note: S23.ARG=Flow-ID2, SeqNum

#### 4.2.3. H.Encaps.R.L2: H.Encaps.R Applied to Received L2 Frames

The H.Encaps.R.L2 behavior encapsulates a received Ethernet frame and its attached VLAN header, if present, in an IPv6 packet with an SRH. The Ethernet frame becomes the payload of the new IPv6 packet.

The H.Encaps.R.L2 encapsulation behavior is similar to H.Encaps.R but sets an Ethernet specific outer Next Header and lacks the TTL/Hop Limit related action. H.Encaps.R.L2 is defined as follows:

```
S01. Push an IPv6 header with its own SRH
     Set the ARG part of the LAST SID in the segment list
S02. Set outer IPv6 SA = T and outer IPv6 DA to the first SID
     in the segment list
S03. Set outer Payload Length, Traffic Class, Hop Limit, and
     Flow Label fields
S04. Set the outer Next Header value
S05. <N/A>
S06. Submit the packet to the IPv6 module for transmission to S11
```

The Next Header field of the SRH MUST be set to 143.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV.

The encapsulating node MUST remove the preamble (if any) and frame check sequence (FCS) from the Ethernet frame upon encapsulation, and the decapsulating node MUST regenerate, as required, the preamble and FCS before forwarding the Ethernet frame.

#### 4.2.4. H.Encaps.R.L2.Red: H.Encaps.R.L2 with Reduced Encapsulation

The H.Encaps.R.L2.Red behavior is an optimization of the H.Encaps.R.L2 behavior.

H.Encaps.R.L2.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only placed in the Destination Address field of the pushed IPv6 header.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag, or TLV.

## 5. Meta Data to Support Redundancy Protection

To support the redundancy protection function, flow identification and sequence number are added in the packet and further used at redundancy node when the elimination function is executed. Flow identification identifies one specific flow of redundancy protection, and is usually allocated from centralized controller to SR ingress node or redundancy node in SR network. Note that flow identification can also be allocated and advertised by redundancy node. BGP, PCEP or Netconf protocols can facilitate the advertisement and distribution of flow identification among controller and redundancy nodes. Sequence number distinguishes the packets within a flow by specifying the order of packets. Not like the uniqueness of flow identification to one specific flow, sequence number keeps changing to each packet within a flow. It is RECOMMENDED to add the sequence number in forwarding plane as performance and scalability is required.

The explicit format of Redundancy SID (RSID) is network addressing design specific. Redundancy specific parameters are encoded as follows:

* LOC: specifies the redundancy node (same allocation rule applies as for any SRv6-enabled node).

* FUNCT: a single value represents the redundancy function of a redundancy node.

* ARG: Contains the Flow-ID and the Sequence Number parameters.

Note: if Function=RSID, Arg=0 is also a meaningful value and does not refer to the lack of arguments.

Note2: Encoding the FlowID and SeqNum as Arguments of the SID implies that when the RSID is in the IPv6 DA, the DA changes on a per packet basis for the redundancy protected flow, and it may alter the ECMP hashing. This can be avoided for example by using additional node specific SIDs before the RSID (e.g., End) or by excluding those bits from ECMP hashing.

## 6. Segment Routing Policy to Support Redundancy Protection

Redundancy Policy is a variation of SR Policy to conduct the replicas to multiple disjoint paths for redundancy protection. It extends SR policy [I-D.ietf-spring-segment-routing-policy] to include more than one active and parallel ordered lists of segments between redundancy node and merging node, and all the ordered lists of segments are used at the same time to steer each copy of flow into different disjoint

paths.

## 7. IANA Considerations

This document requires registration of End.R behavior in "SRv6 Endpoint Behaviors" sub-registry of "Segment Routing Parameters" registry.

IANA maintains The "SRv6 Endpoint Behaviors" sub-registry of the "Segment Routing Parameters" registry. IANA is reuqested to make one new assignments from the First Come First Served portion of the registry as follows:

| Value | Hex | Endpoint Behavior | Reference | Change Controller |
|---|---|---|---|---|
| TBD1 | xTBD1 | End.R | [This.I-D] | IETF |

## 8. Security Considerations

The introduction of Redundancy Segments and Merging Segments in Segment Routing networks introduces new vectors for security threats that must be carefully mitigated.

### 8.1. Packet Duplication

Redundancy protection intentionally replicates packets across multiple paths. Without proper admission control or policy enforcement, an attacker could exploit this mechanism to amplify traffic, overwhelming downstream links or merging nodes.

The use of redundancy protection SHOULD be restricted to trusted applications and provisioned via authenticated and authorized controllers (e.g., using BGP with RPKI or PCEP with TLS). Rate-limiting and flow admission control at the ingress SHOULD be employed to prevent abuse.

### 8.2. Sequence Number Spoofing

The merging node relies on sequence numbers to de-duplicate packets. An attacker that can inject or manipulate these sequence numbers could cause legitimate packets to be dropped or reordered.

Redundancy Segments MUST be deployed only within trusted SR domains.

### 8.3. Information Disclosure

Redundancy protection may involve topology-specific path selections that reveal operational characteristics of the network (e.g., availability of disjoint paths).

Such information SHOULD NOT be exposed outside the trusted SR domain. Control-plane interactions involving Redundancy Segments SHOULD be encrypted and authenticated (e.g., BGP with TCP-AO, PCEP over TLS).

### 8.4. State Exhaustion at Redundancy Node

Redundancy nodes with elimination functionality need to maintain state (e.g., sequence windows, buffering) for each redundancy-protected flow. An attacker might attempt to create many such flows to exhaust memory or processing capacity.

Redundancy nodes SHOULD limit the number of concurrent redundancy flows per source. Idle timeout mechanisms MUST be implemented to garbage-collect stale state.

## 9. Contributors

Fan Yang
Huawei
China
Email: shirley.yangfan@huawei.com

## 10. Acknowledgements

The authors would like to thank Bruno Decraene, Ron Bonica, James Guichard, Jeffrey Zhang, Balazs Varga, Adrian Farrel for their valuable comments and discussions.

## 11. Appendix A. Example

This appendix shows how the described End.R mechanisms can be used in an SRv6 network.

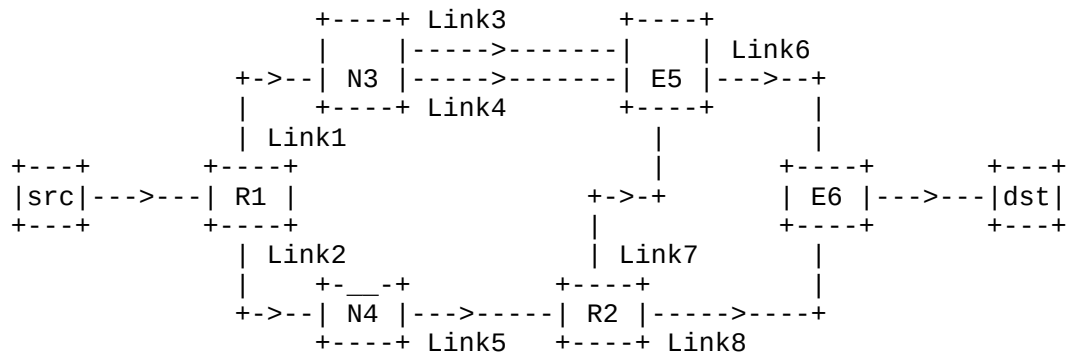


Figure 3: Example Topology

In the reference topology:

* Nodes N3, R1, R2, E5 and E6 are SRv6-capable nodes.

* Nodes R1, R2, E5 and E6 are Redundacy nodes.

* Nodes N4 is an IPv6 node that is not SRv6-capable.

* Node j has an IPv6 loopback address 2001:db8:L:j::/128.

* A SID at node j with locator block 2001:db8:K::/48 and function U is represented by 2001:db8:K:j:U::.

* 2001:db8:K:j:P:: is explicitly allocated as the End.R SID at node j. For example, 2001:db8:K:2:P:: represents End.R at node R2.

* 2001:db8:K:j:Xin:: is explicitly allocated as the End.X SID at node j towards neighbor node i via the nth link between nodes i and j. For example, 2001:db8:K:3:X51:: represents End.X at node N3 towards node E5 via link3 (the first link between nodes N3 and E5). Similarly, 2001:db8:K:3:X52:: represents the End.X at node N3 towards node E5 via link4 (the second link between nodes N3 and E5).

If the src node sends a packet to the dst node for which per packet redundancy is configured, then the nodes with Redundancy functions provide the required replication or elimination functions. For instance, in the example in Figure 3:

* Node src sends a UDP packet as follows: (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node R1, which is an SRv6-capable Redundancy node, identifies the flow the packet belongs to. As replication is configured for the given flow, R1 performs the replication action and intends to send the packet to the next Redundancy nodes (E5 and R2). These nodes are reachable via SRv6, so R1 performs H.Encaps.R(.Red) on the replicas with a path specific SRH. The argument part of the End.R SID involves the Flow-ID and the SeqNum. Specifically, one replica is sent on link-1 towards E5 (2001:db8:L:1::, 2001:db8:K:3:X51::) (2001:db8:K:5:P:arg::, 2001:db8:K:3:X51::, SL=1, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload) and the other replica is sent on link-2 towards R2 (2001:db8:L:1::, 2001:db8:K:2:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node N3, which is an SRv6-capable node, performs the standard SRH processing. Specifically, it executes the End.X behavior indicated by the 2001:db8:K:3:X51:: SID and forwards the packet on link3 to node E5.

* Node N4, which is a non-SRv6-capable node, performs the standard IPv6 processing. Specifically, it forwards the UDP packet based on DA 2001:db8:K:2:P:arg:: in the IPv6 header towards node R2.

* Node R2, which is an SRv6-capable Redundancy node, identifies the packet as targeted to the local Redundancy function. R2 performs the decapsulation and forwards the exposed payload and the ARG part to the redundancy functionality. The redundancy function identifies the flow the packet belongs to. As replication is configured for the given flow, R2 performs the replication action and intends to send the packet to the next redundancy nodes (E5 and E6). These nodes are reachable via SRv6, so R2 performs H.Encaps.R(.Red) on the replicas with a path specific SRH. The argument part of the End.R SID involves the Flow-ID and the SeqNum. Specifically, one replica is sent on link-7 towards E5 (2001:db8:L:2::, 2001:db8:K:5:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload) and the other replica is sent on link-8 towards E6 (2001:db8:L:2::, 2001:db8:K:6:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node E5, which is an SRv6-capable Redundancy node, identifies the packets as targeted to the local redundancy function. E5 performs the decapsulation and forwards the payload and the ARG part to the redundancy functionality. The redundancy function identifies the flow the packet belongs to. As elimination is configured for the given flow, the elimination action is performed on the packets received over Link3 and Link7. E5 intends to send the packet to the next redundancy node (E6), which is reachable via SRv6, so E6 performs H.Encaps.R(.Red) with a path specific SRH. The argument part of the End.R SID involves the Flow-ID and the SeqNum. Specifically, the replica received first is sent on link-6 towards E6 (2001:db8:L:5::, 2001:db8:K:6:P:arg::, NH = IPv6) (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

* Node E6, which is an SRv6-capable redundancy node, identifies the packets as targeted to the local redundancy function. It performs the decapsulation and forwards the payload and the ARG part to the redundancy functionality. The redundancy function identifies the flow the packet belongs to. As elimination is configured for the given flow, the elimination action is performed on the packets received over Link6 and Link8. E6 is the last redundancy node, so after the redundancy function it send the UDP packet towrds the destination. Specifically, the replica received first is sent towards the destination (2001:db8:src::1, 2001:db8:dst::1, NH = UDP)(UDP payload).

The example topology shown in Figure 3 is constructed to show the usage of RSID. Note that any of the links can be replaced with an SRv6 network segment. The above described principles are applicable to such more complex network topologies as well.

# 12. References

## 12.1. Normative References

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-22, 22 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-segment-routing-policy-22>.

[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>.

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

[RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <https://www.rfc-editor.org/info/rfc8964>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

## 12.2. Informative References

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

## Authors' Addresses

Xuesong Geng
Huawei Technologies
China
Email: gengxuesong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
China
Email: mach.chen@huawei.com

Pablo Camarillo Garvia (editor)
Cisco Systems
Spain
Email: pcamaril@cisco.com

Gyan Mishra
Verizon Inc.
Email: gyan.s.mishra@verizon.com

Balazs Varga
Ericsson
Email: balazs.a.varga@ericsson.com

Ferenc Fejes
Ericsson
Email: ferenc.fejes@ericsson.com