

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 22 June 2022

H. Chen
China Telecom
Z. Hu
Huawei Technologies
H. Chen
Futurewei
X. Geng
Huawei Technologies
Y. Liu
China Mobile
G. Mishra
Verizon Inc.
19 December 2021

SRv6 Midpoint Protection
draft-chen-rtgwg-srv6-midpoint-protection-06

Abstract

The current local repair mechanism, e.g., TI-LFA, allows local repair actions on the direct neighbors of the failed node to temporarily route traffic to the destination. This mechanism could not work properly when the failure happens in the destination point or the link connected to the destination. In SRv6 TE, the IPv6 destination address in the outer IPv6 header could be the dedicated endpoint of the TE path rather than the destination of the TE path. When the endpoint fails, local repair couldn't work on the direct neighbor of the failed endpoint either. This document defines midpoint protection for SRv6 TE path, which enables the direct neighbor of the failed endpoint to do the function of the endpoint, replace the IPv6 destination address to the other endpoint, and choose the next hop based on the new destination address.

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.

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 22 June 2022.

Copyright Notice

Copyright (c) 2021 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. SRv6 Midpoint Protection Mechanism	3
3. SRv6 Midpoint Protection Example	3
4. SRv6 Midpoint Protection Behavior	5
4.1. Transit Node as Repair Node	5
4.2. Endpoint Node as Repair Node	6
4.3. Endpoint x Node as Repair Node	6
5. Determining whether the Endpoint could Be Bypassed	7
6. Security Considerations	7
7. IANA Considerations	7
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Authors' Addresses	9

1. Introduction

The current mechanism, e.g., TI-LFA ([I-D.ietf-rtgwg-segment-routing-ti-lfa]), allows local repair actions on the direct neighbors of the failed node to temporarily route traffic to the destination. This mechanism could not work properly when the failure happens in the destination point or the link connected to the destination. In SRv6 TE, the IPv6 destination address in the outer IPv6 header could be the dedicated endpoint of the TE path rather than the destination of the TE path ([RFC8986]). When the endpoint fails, local repair couldn't work on the direct neighbor of the failed endpoint either. This document defines midpoint protection for SRv6 TE path, which enables the direct neighbor of the failed endpoint to do the function of the endpoint, replace the IPv6 destination address to the other endpoint, and choose the next hop based on the new destination address.

2. SRv6 Midpoint Protection Mechanism

When an endpoint node fails, the packet needs to bypass the failed endpoint node and be forwarded to the next endpoint node of the failed endpoint. There are two stages or time periods after an endpoint node fails. The first is the time period from the failure until the IGP converges on the failure. The second is the time period after the IGP converges on the failure.

During the first time period, the packet will be sent to the direct neighbor of the failed endpoint node. After detecting the failure of its interface to the failed endpoint node, the neighbor forwards the packets around the failed endpoint node. It changes the IPv6 destination address with the IPv6 address of the next endpoint node (or the last or other reasonable endpoint node) which could avoid going through the failed endpoint.

During the second time period, the packet of a SRv6 TE path may not be sent to the direct neighbor of the failed endpoint node. There is no route to the failed endpoint node after the IGP converges. When a previous hop node of the failed endpoint node finds out that there is no route to the IPv6 destination address (of the failed endpoint node), it changes the IPv6 destination address with the IPv6 address of the next endpoint node. Note that the previous hop node may not be the direct neighbor of the failed endpoint node.

3. SRv6 Midpoint Protection Example

The topology in Figure 1 illustrates an example of network topology with SRv6 enabled on each node.

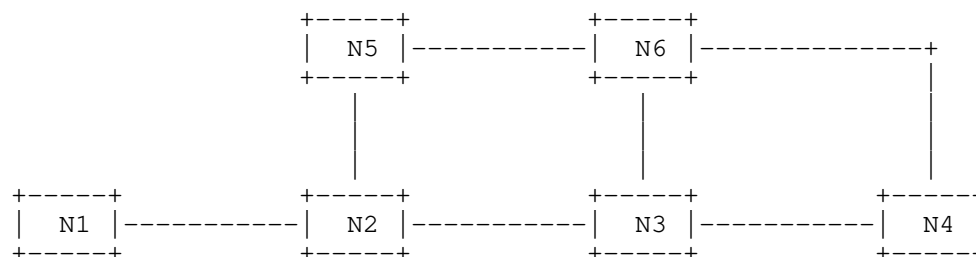


Figure 1: An example of network for midpoint protection

In this document, an end SID at node n with locator block B is represented as $B:n$. An end.x SID at node n towards node k with locator block B is represented as $B:n:k$. A SID list is represented as $\langle S1, S2, S3 \rangle$ where $S1$ is the first SID to visit, $S2$ is the second SID to visit and $S3$ is the last SID to visit along the SRv6 TE path.

In the reference topology, suppose that Node $N1$ is an ingress node of SRv6 TE path going through $N3$ and $N4$. Node $N1$ steers a packet into a segment list $\langle B:3, B:4 \rangle$.

When node $N3$ fails, the packet needs to bypass the failed endpoint node and be forwarded to the next endpoint node after the failed endpoint in the TE path. When outbound interface failure happens in the Repair Node (which is not limited to the previous hop node of the failed endpoint node), it performs the proxy forwarding as follows:

During the first time period (i.e., before the IGP converges), node $N2$ (direct neighbor of $N3$) as a Repair Node forwards the packets around the failed endpoint $N3$ after detecting the failure of the outbound interface to the endpoint $B:3$. It changes the IPv6 destination address with the next sid $B:4$. $N2$ detects the failure of outbound interface to $B:4$ in the current route, it could use the normal Ti-LFA repair path to forward the packet, because it is not directly connected to the node $N4$. $N2$ encapsulates the packet with the segment list $\langle B:5:6 \rangle$ as a repair path.

During the second time period (i.e., after the IGP converges), node $N1$ does not have any route to the failed endpoint $N3$ in its FIB. Node $N1$, as a Repair Node, forwards the packets around the failed endpoint $N3$ to the next endpoint node (e.g., $N4$) directly. There is no need to check whether the failed endpoint node is directly connected to $N1$. $N1$ changes the IPv6 destination address with the next sid $B:4$. Since IGP has completed convergence, it forwards packets directly based on the IGP SPF path

4. SRv6 Midpoint Protection Behavior

A node N protecting the failure of an endpoint node on a SRv6 path may be one of the following types:

- * a transit node: the destination address (DA) of the packet received by N is not N's local SID.
- * an endpoint node: the destination address (DA) of the packet received by N is a N's local END SID.
- * an endpoint x node (i.e., an endpoint with cross-connect node): the destination address (DA) of the packet received by N is a N's local End.X SID with an array of layer 3 adjacencies.

This section describes the behavior of each of these nodes as a repair node for the two time periods after the endpoint node fails.

4.1. Transit Node as Repair Node

When the Repair Node is a transit node, it provides fast protection against the endpoint node failure as follows after looking up the FIB.

```
IF the primary outbound interface used to forward the packet failed
  IF NH = SRH && SL != 0 and
    the failed endpoint is directly connected to Repair Node THEN
    SL decreases*; update the IPv6 DA with SRH[SL];
    FIB lookup on the updated DA;
    forward the packet according to the matched entry;
  ELSE
    forward the packet according to the backup nexthop;
ELSE IF there is no FIB entry for forwarding the packet THEN
  IF NH = SRH && SL != 0 THEN
    SL decreases*; update the IPv6 DA with SRH[SL];
    FIB lookup on the updated DA;
    forward the packet according to the matched entry;
  ELSE
    drop the packet;
ELSE
  forward accordingly to the matched entry;
```

*: SL could be decreased by any dedicated value from [1-N], where N is the current value of SL.

4.2. Endpoint Node as Repair Node

When the Repair Node is an endpoint node, it provides fast protections for the failure through executing the following procedure after looking up the FIB for the updated DA.

```
IF the primary outbound interface used to forward the packet failed
  IF NH = SRH && SL != 0 and
    the failed endpoint is directly connected to Repair Node THEN
    SL decreases; update the IPv6 DA with SRH[SL];
    FIB lookup on the updated DA;
    forward the packet according to the matched entry;
  ELSE
    forward the packet according to the backup nexthop;
ELSE IF there is no FIB entry for forwarding the packet THEN
  IF NH = SRH && SL != 0 THEN
    SL decreases; update the IPv6 DA with SRH[SL];
    FIB lookup on the updated DA;
    forward the packet according to the matched entry;
  ELSE
    drop the packet;
ELSE
  forward accordingly to the matched entry;
```

4.3. Endpoint x Node as Repair Node

When the Repair Node is an endpoint x node, it provides fast protections for the failure through executing the following procedure after updating DA.

```
IF the layer-3 adjacency interface is down THEN
  FIB lookup on the updated DA;
  IF the primary interface used to forward the packet failed THEN
    IF NH = SRH && SL != 0 and
      the failed endpoint directly connected to Repair Node THEN
      SL decreases; update the IPv6 DA with SRH[SL];
      FIB lookup on the updated DA;
      forward the packet according to the matched entry;
    ELSE
      forward the packet according to the backup nexthop;
  ELSE IF there is no FIB entry for forwarding the packet THEN
    IF NH = SRH && SL != 0 THEN
      SL decreases; update the IPv6 DA with SRH[SL];
      FIB lookup on the updated DA;
      forward the packet according to the matched entry;
    ELSE
      drop the packet;
  ELSE
    forward accordingly to the matched entry;
```

5. Determining whether the Endpoint could Be Bypassed

SRv6 Midpoint Protection provides a mechanism to bypass a failed endpoint. But in some scenarios, some important functions may be implemented in the bypassed failed endpoints that should not be bypassed, such as firewall functionality or In-situ Flow Information Telemetry of a specified path. Therefore, a mechanism is needed to indicate whether an endpoint can be bypassed or not. [I-D.li-rtgwg-enhanced-ti-lfa] provides method to determine whether enable SRv6 midpoint protection or not by defining a "no bypass" flag for the SIDs in IGP.

6. Security Considerations

This section reviews security considerations related to SRv6 Midpoint protection processing discussed in this document. To ensure that the Repair node does not modify the SRH header Encapsulated by nodes outside the SRv6 Domain. Only the segment within the SRH is same domain as the repair node. So it is necessary to check the skipped segment have same block as repair node.

7. IANA Considerations

This document makes no request of IANA.

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

8. Acknowledgements

9. References

9.1. Normative References

- [I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over IPv6 Dataplane", Work in Progress, Internet-Draft, draft-ietf-lsr-isis-srv6-extensions-18, 20 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-isis-srv6-extensions-18.txt>>.
- [I-D.ietf-lsr-ospfv3-srv6-extensions]
Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak, "OSPFv3 Extensions for SRv6", Work in Progress, Internet-Draft, draft-ietf-lsr-ospfv3-srv6-extensions-03, 19 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-ospfv3-srv6-extensions-03.txt>>.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [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>>.
- [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>>.

9.2. Informative References

- [I-D.hu-spring-segment-routing-proxy-forwarding]
Hu, Z., Chen, H., Yao, J., Bowers, C., Yongqing, and Yisong, "SR-TE Path Midpoint Restoration", Work in Progress, Internet-Draft, draft-hu-spring-segment-routing-

proxy-forwarding-15, 24 October 2021,
<<https://www.ietf.org/archive/id/draft-hu-spring-segment-routing-proxy-forwarding-15.txt>>.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>>.

[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-14, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-14.txt>>.

[I-D.li-rtgwg-enhanced-ti-lfa]
Li, C., Hu, Z., Zhu, Y., and S. Hegde, "Enhanced Topology Independent Loop-free Alternate Fast Re-route", Work in Progress, Internet-Draft, draft-li-rtgwg-enhanced-ti-lfa-05, 21 October 2021, <<https://www.ietf.org/archive/id/draft-li-rtgwg-enhanced-ti-lfa-05.txt>>.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", Work in Progress, Internet-Draft, draft-sivabalan-pce-binding-label-sid-07, 8 July 2019, <<https://www.ietf.org/archive/id/draft-sivabalan-pce-binding-label-sid-07.txt>>.

[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>>.

Authors' Addresses

Huanan Chen
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou
510000
China

Email: chenhuan6@chinatelecom.cn

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China

Email: huzhibo@huawei.com

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Xuesong Geng
Huawei Technologies

Email: gengxuesong@huawei.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America

Phone: 301 502-1347

Email: gyan.s.mishra@verizon.com

Spring
Internet-Draft
Intended status: Standards Track
Expires: June 30, 2022

D. Lu
M. Chen
Li. Su
China Mobile
Wei. Pan
Cheng. Li
Huawei Technologies
Dec 27, 2021

SRH and IP header protection
draft-chen-spring-srv6-srh-security-02

Abstract

This document proposes a method to protect SRH and IP header using signature which stored in the TLV, this scheme can apply to SRv6 and G-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 June 30, 2022.

Copyright Notice

Copyright (c) 2021 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	2
2. Terminology	3
3. New TLV Type for Signature	3
4. SRH protection used in SRv6 and G-SRv6	4
5. signing and verifying process	7
6. verifying optimization process	8
7. Security Considerations	8
8. IANA Considerations	9
9. Acknowledgement	9
10. Normative References	9
Authors' Addresses	9

1. Introduction

SRv6 is a protocol for forwarding IPv6 packets over a network based on the concept of source routing. By inserting a Segment Routing Header (SRH) into the IPv6 packet, an explicit IPv6 address stack is pressed into the SRH, and the destination address and offset address stack are constantly updated by the intermediate node to complete hop-by-hop forwarding, SRH is defined in RFC8754 [RFC8754]

G-SRv6 is generalized Segment Routing over IPv6 which can reduce the overhead of SRv6 by encoding the Generalized SIDs in SID list, the compression solution is designed in the draft [I-D.cl-spring-generalized-srv6-for-cmpr].

As an emerging source routing protocol, SRv6 is confronted with various threat of source routing attacks. By defining SRH, attackers can construct various source routing attacks, such as bypassing key detection nodes of network and constructing malicious loops.

SRv6 networks generally define SRv6 trust domains for basic security protection, which is also mentioned in the draft [I-D.li-spring-srv6-security-consideration] and RFC 8754 [RFC8754]. Firstly, the address space in the SRv6 trust domain is defined to avoid SRv6 trust domain address leakage. Then ACL filtering is enabled at the boundary of the trust domain, and packets whose destination address is SRv6 trust domain are discarded to avoid source routing attack on SRv6 trust domain by attacking packets.

SRv6 trust domains use Segment Binding technology for basic security. RFC8754 defines SRv6 HMAC TLV for IPv6 source address and SRH integrity protection which based on SRv6 trust domain, identity

authentication based on the shared key, to prevent illegal access and tamper header, so as to prevent various source routing attacks. However, there is a problem with this scheme, HMAC verification is based on symmetric key verification, that means all network nodes that need to be verified have to share the same key, there may exist a problems.

Secret key leak problem: when a single point's key was leaked, then all the trust domain was compromised.

In this document we present an alternative method for Segment Routing Header protection.

2. Terminology

This document uses the terminology defined in [RFC8754].

3. New TLV Type for Signature

This section describes how to use the certificate to authenticate the header. The source address field in IP header and several fields in SRH are protected by signature, and the result of signature is stored in TLV, the TLV format is consistent with the HMAC TLV defined in RFC8754, we describe this in Figure 1.

By defining a new type of TLV which the Type is 6 and we call it Auth TLV, indicates that the TLV is used for signature protection based on asymmetric secret keys. Auth TLV is described in Figure 1.

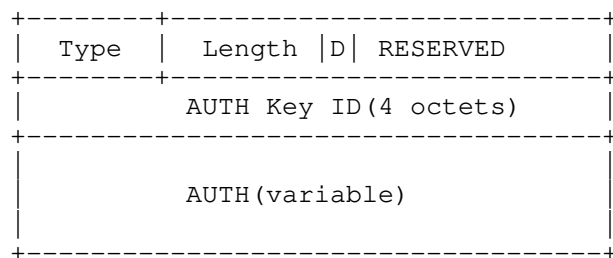


Figure 1: Auth TLV format

Type: 6.

Length: The length of the variable-length data in bytes.

D: 1 bit. 1 indicates that the Destination Address verification is disabled due to use of a reduced Segment List.

RESERVED: 15 bits. MUST be 0 on transmission.

AUTH Key ID: A 4-octet opaque number that uniquely identifies the hash algorithm, signature algorithm, and certificate serial number used for signature authentication.

AUTH: the content of the signature that protects the field, in multiples of 8 octets, at most 32 octets.

The AUTH TLV is used to protect IPv6 source address, SRH header for signature protection. Which fields are in the range of the signature check? they are described in Figure 2 and Figure 3, Figure 2 is for SRv6 and Figure 5 is for G-SRv6.

The AUTH Key ID field is opaque--i.e., it has neither syntax nor semantic except as an identifier of the right combination of hash algorithm, signature algorithm and certificate serial number

Hash Algorithm indicates the hash algorithm used in the header, such as SHA256, and we do not recommend using SHA1.

Signature Algorithm indicates the asymmetric signature algorithm used, such as ECDSA and RAS2048.

Certificate Serial number used to identify certificate that issued by CA, if a custom certificate is used, the Certificate Serial number represents the identity of the custom certificate.

4. SRH protection used in SRv6 and G-SRv6

Segment routing header is defined in RFC8754, when user choose to use the method proposed in this draft, the complete SRv6 header with Auth TLV is show as figure 2, and figure 3 is for G-SRV6.

Version	Traffic class	Flow Label		
Payload Length		Next=43		Hop Limit
Source Address				
Destination Address				
Next Header	Hdr Ext Len	Routing Type=4	Segment Left	
Last Entry	Flags	Tag		
Segment List[0]				
Segment List[1]				
Segment List[2]				
Type=6	Length	D	Reserved	
Auth Key ID				
Auth(variable)				
IPv6 Payload				

Figure 2: Complete SRv6 header with Auth TLV

Figure 5 is the detailed structure for G-SRv6

Version	Traffic class	Flow Label	
Payload Length		Next=43	Hop Limit
Source Address			
Destination Address			
Next Header	Hdr Ext Len	Routing Type=4	Segment Left
Last Entry	Flags	Tag	
G-SID Container[0]			
G-SID Container[1]			
G-SID Container[2]			
Type=6	Length	D	Reserved
Auth Key ID			
Auth(variable)			
IPv6 Payload			

Figure 3: Complete SRv6 header with Auth TLV

Signature check those fields that need to be protected will be signed, the range of signatures includes IPv6 Source address, SRH Last Entry, SRH Flags, SRH Segment List, AUTH TLV D, AUTH TLV Reserved, AUTH TLV Auth Key ID.

what's the difference between this scheme with the AH of the IPv6? In this scheme, the message is protected in the routing extension header with type = 43, and AH uses the extension header with type = 51, they are totally independent. According to the IPv6 protocol, the processing order of AH extension header is lower than that of routing extension header, that is, the AH extension header will not be parsed until the source route forwarding is completed and the routing extension header pops up. AH cannot be directly used to protect the source route attack.

5. signing and verifying process

First, need the CA center to issue a root certificate to the controller that will generate controller's public and private key, or the controller use custom certificate, it depends on the detail implementation. How to preset and update a CA certificate on a device is out of scope in this document. The process described in this document uses CA certificates by default.

SRv6 controller uses the private key of the certificate to hash the SRH and IP header, and encapsulates the digital signature generated by SRv6 header and controller in the SRv6 source node. The signature process is divided into three steps.

Step1: Preset certificates, include private keys and controller certificates on SRv6 controllers, and CA root certificates on key network devices;

Step2: After the secure connection is established between the controller and the network device on the control plane, perform public key certificate distribution and signature algorithm selection, and inform the key node the selection result.

Step3: SRv6 controller uses the private key, the hash algorithm and the asymmetric algorithm selected in the step2 to sign the packet header which generated according to the routing result, and store the signature results in the TLV, finally sends the routing result which include the signature to the source node, the source node wraps and forwards an SRv6 packet with a signature, the SRv6 network structure is described in Figure 4.

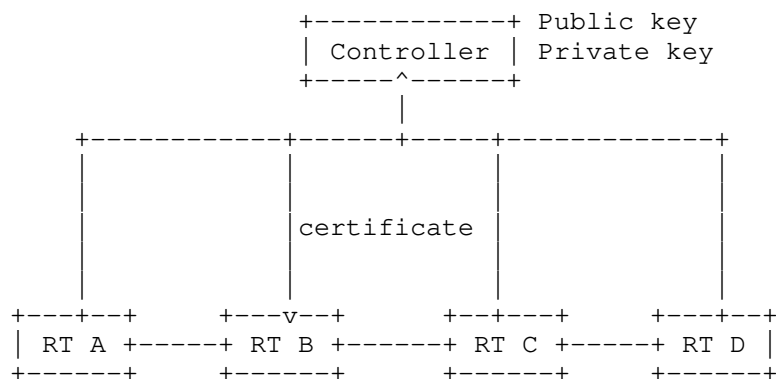


Figure 4: SRv6 network structure

Signature verification is required at key network nodes, it's also divided into three steps.

Step1: Enable signature verification at the key nodes.

Step2: Request a public key certificate from the controller.

Step3: calculate the hash value according to the header, and use the public key to decrypt the signature in the message, compare the decryption result with the hash value, if verify successful, forward the message, otherwise, the message is discarded.

6. verifying optimization process

When asymmetric key is used to verify the signature of the forward message on the data plane, the processing efficiency of the forward message is reduced. An efficient lookup table forwarding mechanism for signature verification can be considered, which verifies the signature of the first packet of the data, and records the hash result and signature of the packet header into the hash table. The subsequent packets can directly find the hash table and compare the signature result, no more need to decrypt, also can divide into three steps.

Step1: When the interface of signature verification is opened and the SRV6 message is received, the hash value of the message header is calculated and finds if the local hash table is hit, the local hash table contains hash value and signature value, and they are bound.

Step2: If the local hash table is not hit, the controller's public key is used to decrypt the signature and compare whether the decrypted result is consistent with the calculated hash value. If not, the message is discarded. If the hash value and decrypted result are consistently then recorded to the local hash table, and the processing packet is forwarded.

Step3: If the local hash table is hit, the signature value in message is compared with hash table's signature value, if yes then forwarded to process the message, if not then discarded.

7. Security Considerations

SRv6 is threatened by various source routing attacks. By defining SRH, an attacker can construct various source routing attacks, such as bypassing the key detection nodes of the network and constructing malicious loops, in this draft we propose a method, it can prevent a single device from being compromised and exposes the network's shared key, then the entire network is under threat.

8. IANA Considerations

This document does not require any action from IANA.

9. Acknowledgement

TBD

10. Normative References

- [I-D.cl-spring-generalized-srv6-for-cmpr]
(editor), W. C., Li, Z., (editor), C. L., Clad, F., Liu,
A., Xie, C., Liu, Y., and S. Zadok, "Generalized SRv6
Network Programming for SRv6 Compression", draft-cl-
spring-generalized-srv6-for-cmpr-04 (work in progress),
October 2021.
- [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>>.

Authors' Addresses

Dongjie Lu
China Mobile
32, Xuanwumen West
BeiJing, BeiJing 100053
China

Email: ludongjie@chinamobile.com

Meiling Chen
China Mobile
32, Xuanwumen West
BeiJing, BeiJing 100053
China

Email: chenmeiling@chinamobile.com

Li Su
China Mobile
32, Xuanwumen West
BeiJing 100053
China

Email: suli@chinamobile.com

Wei Pan
Huawei Technologies
101 Software Avenue, Yuhuatai District
Nanjing
China

Email: william.panwei@huawei.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: c.l@huawei.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 7, 2022

G. Fioccola
T. Zhou
Huawei
M. Cociglio
Telecom Italia
February 3, 2022

Segment Routing Header encapsulation for Alternate Marking Method
draft-fz-spring-srv6-alt-mark-02

Abstract

This document describes how the Alternate Marking Method can be used as the passive performance measurement tool in an SRv6 network. It defines how Alternate Marking data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header.

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 7, 2022.

Copyright Notice

Copyright (c) 2022 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	2
2. Application of the Alternate Marking to SRv6	3
2.1. Controlled Domain	4
3. Definition of the SRH AltMark TLV	4
3.1. Data Fields Format	4
4. Use of the SRH AltMark TLV	6
5. Alternate Marking Method Operation	7
6. Security Considerations	7
7. IANA Considerations	7
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Authors' Addresses	9

1. Introduction

[RFC8321] and [RFC8889] describe a passive performance measurement method, which can be used to measure packet loss, latency and jitter on live traffic. Since this method is based on marking consecutive batches of packets, the method is often referred as Alternate Marking Method.

This document defines how the Alternate Marking Method ([RFC8321]) can be used to measure packet loss and delay metrics for Segment Routing with IPv6 data plane (SRv6).

[RFC8754] defines the Segment Routing Header (SRH) and how it is used by nodes that are Segment Routing (SR) capable.

[I-D.fioccola-v6ops-ipv6-alt-mark] reported a summary on the possible implementation options for the application of the Alternate Marking Method in an IPv6 domain. [I-D.ietf-6man-ipv6-alt-mark] defines a new TLV that can be encoded in the Option Headers (both Hop-by-hop or Destination) for the purpose of the Alternate Marking Method application in an IPv6 domain.

This document defines how Alternate Marking data is carried as SRH TLV, that can be piggybacked in the packet and transported as part of the SRH. The usage of SRH TLV is introduced in [RFC8754].

2. Application of the Alternate Marking to SRv6

The Alternate Marking Method requires a marking field. A possibility is already offered by [I-D.ietf-6man-ipv6-alt-mark] while the use of a new TLV to be encoded in the SRH is defined in this document.

Since [I-D.ietf-6man-ipv6-alt-mark] defines the IPv6 Application of the Alternate Marking Method through both Hop-by-Hop and Destination Options Header, it is applicable also to SRv6 network. Indeed the use of Destination Option Header carrying Alternate Marking bits coupled with SRH allows to monitor every node along the SR path.

This document introduces the SRH TLV carrying Alternate Marking bits and this can be a preferred approach in case of SRv6 network since it does not rely on the use of Destination Option Header.

The optimization of both implementation and scaling of the Alternate Marking Method is also considered and a way to identify flows is required. The Flow Monitoring Identification field (FlowMonID), as introduced in the next sections, goes in this direction and it is used to identify a monitored flow.

Note that the FlowMonID is different from the Flow Label field of the IPv6 Header ([RFC8200]). Flow Label is used for application service, like load-balancing/equal cost multi-path (LB/ECMP) and QoS. Instead, FlowMonID is only used to identify the monitored flow. The reuse of flow label field for identifying monitored flows is not considered since it may change the application intent and forwarding behaviour. Furthermore the flow label may be changed en route and this may also violate the measurement task. Those reasons make the definition of the FlowMonID necessary for IPv6. Flow Label and FlowMonID within the same packet have different scope, identify different flows, and associate different uses.

An important point that will also be discussed in this document is the uniqueness of the FlowMonID and how to allow disambiguation of the FlowMonID in case of collision.

The following section highlights an important requirement for the application of the Alternate Marking to IPv6 and SRv6. The concept of the controlled domain is explained and it is considered an essential precondition.

2.1. Controlled Domain

[RFC8799] introduces the concept of specific limited domain solutions and, in this regard, it is reported the Application of the Alternate Marking Method as an example.

IPv6 has much more flexibility than IPv4 and innovative applications have been proposed, but for a number of reasons, such as the policies, options supported, the style of network management and security requirements, it is suggested to limit some of these applications to a controlled domain. This is also the case of the Alternate Marking application to SRv6 as assumed hereinafter.

Therefore, the application of the Alternate Marking Method to SRv6 MUST NOT be deployed outside a controlled domain. It is RECOMMENDED that an implementation can be able to reject packets that carry Alternate Marking data and are entering or leaving the controlled domains.

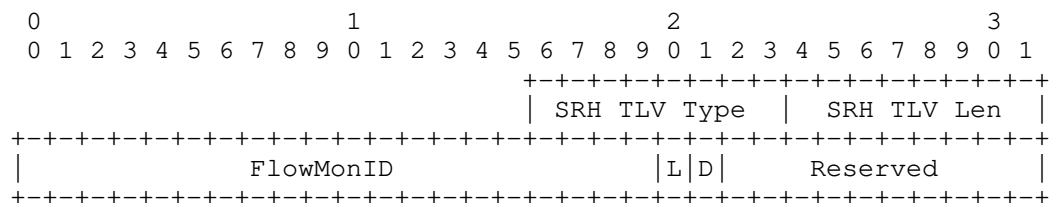
3. Definition of the SRH AltMark TLV

A new TLV carrying the data fields dedicated to the alternate marking method can be defined for the SRH extension headers.

This enables the Alternate Marking Method to take advantage of the network programmability capability of SRv6 ([I-D.ietf-spring-srv6-network-programming]). Specifically, the ability for an SRv6 endpoint to determine whether to process or ignore some specific SRH TLVs is based on the SID function. The nodes that are not capable of supporting the Alternate Marking functionality do not have to look or process the SRH AltMark TLV and can simply ignore it. This also enables collection of Alternate Marking data only from the supporting segment endpoints.

3.1. Data Fields Format

The following figure shows the data fields format for enhanced alternate marking TLV. This AltMark data is expected to be encapsulated as SRH TLV.



where:

- o SRH TLV Type: 8 bit identifier of the type of Option/TLV that needs to be allocated. Unrecognised Types MUST be ignored on receipt.
- o SRH TLV Len: The length of the Data Fields of this TLV in bytes.
- o FlowMonID: 20 bits unsigned integer. The FlowMon identifier is described hereinafter.
- o L: Loss flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];
- o D: Delay flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];
- o Reserved: is reserved for future use. These bits MUST be set to zero on transmission and ignored on receipt.

The Flow Monitoring Identification (FlowMonID) is required for some general reasons:

First, it helps to reduce the per node configuration. Otherwise, each node needs to configure an access-control list (ACL) for each of the monitored flows. Moreover, using a flow identifier allows a flexible granularity for the flow definition.

Second, it simplifies the counters handling. Hardware processing of flow tuples (and ACL matching) is challenging and often incurs into performance issues, especially in tunnel interfaces.

Third, it eases the data export encapsulation and correlation for the collectors.

The FlowMon identifier field is to uniquely identify a monitored flow within the measurement domain. The field is set at the source node. The FlowMonID can be uniformly assigned by the central controller or algorithmically generated by the source node. The latter approach cannot guarantee the uniqueness of FlowMonID but it may be preferred for local or private network, where the conflict probability is small due to the large FlowMonID space.

It is important to note that if the 20 bit FlowMonID is set independently and pseudo randomly there is a chance of collision. So, in some cases, FlowMonID could not be sufficient for uniqueness.

This issue is more visible when the FlowMonID is pseudo randomly generated by the source node and there needs to tag it with additional flow information to allow disambiguation. While, in case of a centralized controller, the controller should set FlowMonID by considering these aspects and instruct the nodes properly in order to guarantee its uniqueness.

4. Use of the SRH AltMark TLV

SRv6 leverages the Segment Routing header which consists of a new type of routing header. Like any other use case of IPv6, Hop-by-Hop and Destination Options are useable when SRv6 header is present. Because SRv6 is a routing header, destination options before the routing header are processed by each destination in the route list.

SRH TLV can also be used to encode the AltMark Data Fields for SRv6 and to monitor every node along the SR path. For SRv6, it may be preferred to use the SRH TLV, while for all the other cases with IPv6 data plane the use of the Hop-by-Hop and Destination Option to carry AltMark data fields (as described in [I-D.ietf-6man-ipv6-alt-mark]) is the best choice.

It is to be noted that the SR nodes implementing the Alternate Marking functionality follows the MTU and other considerations outlined in [I-D.voyer-6man-extension-header-insertion]. Furthermore, in a SRv6 network, the intermediated nodes that are not in the SID list do not consider the SRH, therefore they cannot support and dig into the SRH TLV.

It is possible to summarize the procedure for AltMark data encapsulation in SRv6 SRH:

- * 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 AltMark TLV in the SRH of the data packet, if it supports AltMark functionality and based on local configuration.

- * Intermediate SR Node: The intermediate SR node is any node receiving an IPv6 packet where the destination address of that packet is a local SID. If an intermediate SR node is not capable of processing AltMark TLV, it simply ignores it. While, if an intermediate SR node is capable of processing AltMark TLV, it checks if SRH AltMark TLV is present in the packet using procedures defined in [RFC8754] and process it.

- * Egress Node: The Egress node is the last node in the segment-list of the SRH. The processing of AltMark TLV at the Egress node

is similar to the processing of AltMark TLV at the Intermediate SR Nodes.

5. Alternate Marking Method Operation

[RFC8321], [RFC8889] describe the Alternate Marking Method in general. While [I-D.ietf-6man-ipv6-alt-mark] describe in detail the application and the Operation of the methodology for IPv6.

6. Security Considerations

The security considerations of SRv6 are discussed in [RFC8754] and [I-D.ietf-spring-srv6-network-programming], and the security considerations of Alternate Marking in general and its application to IPv6 are discussed in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark].

The Alternate Marking application to IPv6, defined in [I-D.ietf-6man-ipv6-alt-mark], analyzes different security concerns and related solutions. These aspects are valid and applicable also to this document. In particular the fundamental security requirement is that Alternate Marking **MUST** be applied in a specific limited domain, as also mentioned in [RFC8799].

Alternate Marking is a feature applied to a trusted domain, where one or several operators decide on leveraging and configuring Alternate Marking according to their needs. Additionally, operators need to properly secure the Alternate Marking domain to avoid malicious configuration and attacks, which could include injecting malicious packets into a domain. So the implementation of Alternate Marking is applied within a controlled domain where the network nodes are locally administered. A limited administrative domain provides the network administrator with the means to select, monitor and control the access to the network.

7. IANA Considerations

The SRH TLV Type should be assigned in IANA's Segment Routing Header TLVs Registry.

This draft requests to allocate a SRH TLV Type for Alternate Marking TLV data fields under registry name "Segment Routing Header TLVs" requested by [RFC8754].

SRH TLV Type	Description	Reference
TBD	AltMark Data Fields TLV	This document

8. Acknowledgements

TBD

9. References

9.1. 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>>.

9.2. Informative References

[I-D.fioccola-v6ops-ipv6-alt-mark]
Fioccola, G., Velde, G. V. D., Cociglio, M., and P. Muley, "IPv6 Performance Measurement with Alternate Marking Method", draft-fioccola-v6ops-ipv6-alt-mark-01 (work in progress), June 2018.

[I-D.ietf-6man-ipv6-alt-mark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-12 (work in progress), October 2021.

[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-16 (work in progress), January 2022.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Garvia, P. C., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.voyer-6man-extension-header-insertion]
Voyer, D., Filsfils, C., Dukes, D., Matsushima, S., Leddy, J., Li, Z., and J. Guichard, "Deployments With Insertion of IPv6 Segment Routing Headers", draft-voyer-6man-extension-header-insertion-10 (work in progress), November 2020.

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [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>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8889] Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <<https://www.rfc-editor.org/info/rfc8889>>.

Authors' Addresses

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 February 2022

X. Geng
M. Chen
F. Yang, Ed.
Huawei Technologies
P. Camarillo
Cisco Systems, Inc.
G. Mishra
Verizon Inc.
2 August 2021

SRv6 for Redundancy Protection
draft-geng-spring-sr-redundancy-protection-05

Abstract

Redundancy Protection is a generalized protection mechanism to achieve the high reliability of service transmission in Segment Routing network. The mechanism inherits the "Live-Live" methodology, targeting to enhance the functionalities of Segment Routing over IPv6. Inspired by DetNet Packet Replication and Packet Elimination functions, two new Segments are introduced to provide replication and elimination functions on specific network nodes by leveraging SRv6 Segment programming capabilities.

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 .

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 3 February 2022.

Copyright Notice

Copyright (c) 2021 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	2
2. Terminology	3
2.1. Requirements Language	3
2.2. Terminology and Conventions	3
3. Redundancy Protection in Segment Routing Scenario	4
4. Segment to Support Redundancy Protection	5
4.1. Redundancy Segment	5
4.2. Merging Segment	6
5. Meta Data to Support Redundancy Protection	7
6. Segment Routing Policy to Support Redundancy Protection	7
7. IANA Considerations	7
8. Security Considerations	8
9. Acknowledgements	8
10. References	8
10.1. Normative References	8
10.2. Informative References	8
Authors' Addresses	8

1. Introduction

Redundancy Protection is a generalized protection mechanism to achieve the high reliability of service transmission in Segment Routing network. Specifically, packets of flows are replicated at a network node into two or more copies, which are transported via different paths in parallel. When copies of packets are received and merged at one network node, the redundant packets are determined and further eliminated to guarantee only one copy of packets is transmitted. The mechanism inherits the "Live-Live" methodology, targeting to enhance the functionalities of Segment Routing over IPv6 [RFC8986]. Inspired by DetNet [RFC8655] Packet Replication and Packet Elimination Functions, two new Segments are introduced to provide the replication and elimination functions on specific network

nodes by leveraging SRv6 Segment programming capabilities. As it is unnecessary to perform switchover between different paths triggered by failure detection, 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. In this document, redundancy protection is applied to point-to-point service. The mechanism for P2MP service stays out of the scope of this document.

Segment Routing (SR) leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions, called "segments". A segment can be associated to an arbitrary processing of the packet in the node identified by the segment.

This document extends the Segment Routing capabilities to support the redundancy protection in an SRv6 environment, including the definitions of two new Segments, meta data encapsulation, and a variation of Segment Routing Policy.

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

URLLC: Ultra-Reliable Low-Latency Communication

VR: Virtual Reality

Red Node: Redundancy Node

Mer Node: Merging Node

FID: Flow IDentification

SN: Sequence Number

3. Redundancy Protection in Segment Routing Scenario

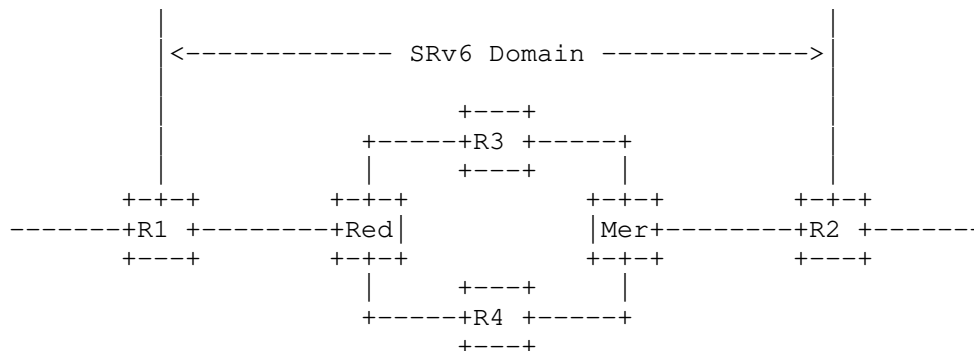


Figure 1: Example Scenario of Redundancy Protection in SRv6 Domain

This figure shows an example of redundancy protection used in SRv6 domain. R1, R2, R3, R4, Red and Mer are SR-capable nodes. When a flow is sent into 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 Red node, known as Redundancy Node, 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.
- 3) Meta data information such as flow identification (FID) and sequence number (SN) is used to facilitate the packet elimination on Merging node (Mer). Flow identification identifies the specific flow, and sequence number distinguishes the packet sequence of a flow. Meta data is either carried in the packet before it arrives at Red node, or added to each of the replicas at Red node.
- 4) The multiple replicas go through different paths until the Mer node. The first received packet of the flow is transmitted from Merging Node to R2, and the redundant packets are eliminated.
- 5) When there is any failures or packet loss in one path, the service continues undisrupted through the other path without break.

6) Sometimes, the packet will arrive out of order because of redundancy protection, the function of reordering may be also necessary on Merging Node. In such case the Merging node may include a reordering function, which is implementation specific and out of the scope of this document.

In this example, service protection is supported by utilizing two packet flows transmitted over two forwarding paths. It is noted that there is no limitation of the number of replicas. For a unidirectional flow, Red node supports replication function, and Mer node supports elimination function. Reordering function MAY be required in combination of elimination function on merging node. To minimize the jitter caused by random packet loss, the disjoint paths are recommended to have similar path forwarding delay.

4. Segment to Support Redundancy Protection

To achieve the packet replication and elimination functions, Redundancy Segment and Merging Segment, as well as the related SRv6 Endpoint Behavior are introduced.

4.1. Redundancy Segment

Redundancy Segment is the identifier of packets which need the replication function on redundancy node. It is also a variation of Binding SID, and associated with a Redundancy Policy to provide segment lists of disjoint paths. Thus, Redundancy segment is associated with service instructions, indicating the following operations:

- * Steers the packet into the corresponding redundancy policy
- * Encapsulates flow identification and sequence number in packets if the two information is not carried in packets
- * Packet replication and segment encapsulation based on the information of redundancy policy, e.g., the number of replication copies, an ordered list of segments with a topological instruction

In the case of SRv6, 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 in the encapsulation mode. The End.R behavior in the insertion mode is for further study.

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.     Decrement IPv6 Hop Limit by 1
S04.     Decrement Segments Left by 1
S05.     Update IPv6 DA with Segment List[Segments Left]
S06.     Add flow identification and sequence number if indicated*
S07.     Duplicate the packets (as number of active SID lists in B)
S08.     Push the new IPv6 headers to each replica. The IPv6 header
        contains an SRH with the SID list in B
S09.     Set the outer IPv6 SA to A
S10.     Set the outer IPv6 DA to the first SID of new SRH SL
S11.     Set the outer Payload Length, Traffic Class, Flow Label,
        Hop Limit and Next-Header fields
S12.     Submit the packet to the egress IPv6 FIB lookup
        for transmission to the new destination
S13.   }
S14. }
```

* Adding flow identification and sequence number is an optional behavior for Redundancy Segment. The instruction execution is determined and explicitly indicated by SR policy or Segment itself.

4.2. Merging Segment

Merging Segment is associated with service instructions, indicates the following operations:

- * Packet merging and elimination: forward the first received packets and eliminate the redundant packets

In order to eliminate the redundant packet of a flow, merging 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 merging node needs to maintain the state of flows, a centralized controller should have a knowledge of merging nodes capability, and never provision the redundancy policy to redundancy node when the computation result goes beyond the flow recovery capability of merging node. The capability advertisement of merging node will be specified separately elsewhere, which is not within the scope of this document.

In the case of SRv6, a new behavior End.M for Merging Segment is defined.

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 Merging SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left > or == 0)   {
S03.     Acquire the sequence number of received packet and
        look it up in table
S04.     If (this sequence number does not exist in the table) {
S05.       Store this sequence number in table
S06.       Remove the outer IPv6+SRH header
S07.       Decrement IPv6 Hop Limit by 1 in inner SRH
S08.       Decrement Segments Left by 1 in inner SRH
S09.       Update IPv6 DA with Segment List[Segments Left] in inner SRH
S10.       Submit the packet to the egress IPv6 FIB lookup and transmit
S11.     }
S12.   ELSE {
S13.     Drop the packet
S14.   }
S15. }
S16. }
```

5. Meta Data to Support Redundancy Protection

To support the redundancy protection function, flow identification and sequence number are required. Flow identification identifies one specific flow of redundancy protection, and is usually allocated from centralized controller to the SR ingress node or redundancy node in SR network. Sequence number distinguishes the packets within a flow by specifying the order of packets. It is usually generated at SR ingress node. If necessary, redundancy node can also facilitate to add sequence number if required. Thus, encapsulations of flow identification and sequence number should be specified accordingly.

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 to include more than one 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 the copies of flow into different disjoint paths.

7. IANA Considerations

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

8. Security Considerations

TBD

9. Acknowledgements

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

10. References

10.1. 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>>.
- [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>>.
- [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>>.

10.2. 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

Fan Yang
Huawei Technologies
China

Email: shirley.yangfan@huawei.com

Pablo Camarillo Garvia
Cisco Systems, Inc.
Spain

Email: pcamaril@cisco.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 October 2022

Z. Hu
Huawei Technologies
H. Chen
Futurewei
J. Yao
Huawei Technologies
C. Bowers
Juniper Networks
Y. Zhu
China Telecom
Y. Liu
China Mobile
11 April 2022

SR-TE Path Midpoint Restoration
draft-hu-spring-segment-routing-proxy-forwarding-19

Abstract

Segment Routing Traffic Engineering (SR-TE) supports explicit paths using segment lists containing adjacency-SIDs, node-SIDs and binding-SIDs. The current SR FRR such as TI-LFA provides fast re-route protection for the failure of a node along a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node. This document describes a mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment of the path. With the restoration of the routes to the failure, the traffic is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

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.

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 October 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
2. Proxy Forwarding	4
3. Protocol Extensions/Re-uses for Proxy Forwarding	4
3.1. Advertising Binding Segment	4
3.2. Advertising Proxy Forwarding	5
4. Proxy Forwarding Example	6
4.1. Advertising Proxy Forwarding	8
4.2. Building Proxy Forwarding Table	8
4.3. Proxy Forwarding for Binding Segment	9
5. Security Considerations	10
6. Acknowledgements	10
7. References	10
7.1. Normative References	10
7.2. Informative References	11
Appendix A. Proxy Forwarding for Adjacency and Node Segment . .	11
A.1. Next Segment is an Adjacency Segment	11
A.2. Next Segment is a Node Segment	12
Authors' Addresses	13

1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using a segment list. SR-TE supports the creation of explicit paths using adjacency-SIDs, node-SIDs, anycast-SIDs, and binding-SIDs. A node-SID in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When the node fails, the network may no longer be able to properly forward traffic on that SR-TE path.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes an SR FRR mechanism that provides fast re-route protection for the failure of a node on a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node and drop the traffic.

To solve this problem,

[I-D.ietf-spring-segment-protection-sr-te-paths] proposes that a hold timer should be configured on every router in a network. After the IGP converges on the event of a node failure, if the node-SID of the failed node becomes unreachable, the forwarding changes should not be communicated to the forwarding planes on all configured routers (including PLRs for the failed node) until the hold timer expires. This solution may not work for some cases such as some of nodes in the network not supporting this solution.

This document describes a proxy forwarding mechanism for the restoration of the routes to the failure of a SR-MPLS TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment on a failed node along the path. With the restoration of the routes to the failure, the traffic for the SR-MPLS TE path is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

1.1. Terminology

SR: Segment Routing.

PLR: Point of Local Repair.

LSP: Link State Protocol Data Unit (PDU) in IS-IS.

LSA: Link State Advertisement in OSPF.

LS: Link State, which is LSP or LSA.

2. Proxy Forwarding

In the proxy forwarding mechanism, each neighbor of a possible failed node advertises its SR proxy forwarding capability in its network domain when it has the capability. This capability indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from the neighbors of a failed node will send traffic using the node-SID of the failed node to the nearest Proxy Forwarder after the IGP converges on the event of the failure.

Once the affected traffic reaches a Proxy Forwarder, it sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list.

For a binding segment of a possible failed node, the node advertises the information about the binding segment, including the binding SID and the list of SIDs/segments associated with the binding SID, to its direct neighbors only. Note that the information is not advertised in the network domain.

After the node fails and the IGP converges on the failure, the traffic with the binding SID of the failed node will reach its neighbor having SR Proxy Forwarding capability. Once receiving the traffic, the neighbor swaps the binding SID with the list of SIDs/segments associated with the binding SID and sends the traffic along the post-failure shortest path to the first node in the segment list.

3. Protocol Extensions/Re-uses for Proxy Forwarding

This section describes the semantic of protocol extensions/re-uses for advertising the information about each binding segment (including its binding SID and the list of SIDs/segments associated with the binding SID) of a node to its direct neighbors and the SR proxy forwarding capability of a node in a network domain.

3.1. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of SIDs/segments, node A advertises an LS containing the binding (i.e., the binding SID and the list of the SIDs/segments) in a binding segment TLV. The LS is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LS is a opaque LSA of LS type 9 (i.e., a link local scope LSA). For IS-IS, the TLV is advertised in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356].

Alternatively, when a protocol (such as PCE or BGP running on a controller) supports sending a binding on a node A to A, this protocol may be extended to send the binding with node A to A's neighbors if the controller knows the neighbors and there are protocol (PCE or BGP) sessions between the controller and the neighbors.

Note: how to send bindings of node A to A's neighbors via which protocol is out of the scope of this document.

3.2. Advertising Proxy Forwarding

When a node P is able to do SR proxy forwarding for its neighboring nodes for protecting the failures of these nodes, P advertises its SR proxy forwarding capability for these nodes. The mirror SID [RFC8402] for a node N (Neighbor of P) advertised by P using IS-IS extensions [RFC8667] indicates the capability of P for N.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P. The route to the proxy prefix/node SID is through proxy forwarding capable nodes.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/node SID of node N. Thus node X will forward the traffic targeting to the prefix/node SID of node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forward the traffic towards its final destination without going through node N.

Note that the behaviors of normal IP forwarding and routing convergences in a network are not changed at all by the SR proxy forwarding. For example, the next hop used by BGP is an IP address (or prefix). The IGP and BGP converge in normal ways for changes in the network. The packet with its IP destination to this next hop is forwarded according to the IP forwarding table (FIB) derived from IGP and BGP routes.

Similar to IS-IS [RFC8667], OSPF should be extended for advertising mirror SID to indicate the capability. Note that OSPF extensions is out of the scope of this document.

4. Proxy Forwarding Example

This section illustrates the proxy forwarding for a binding SID through an example. The proxy forwarding for a node SID and an adjacency SID can refer to [I-D.ietf-spring-segment-protection-sr-te-paths] or Appendix. Figure 1 is an example network topology used to illustrate the proxy forwarding mechanism for a binding SID. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Label Stack 1 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 1, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.

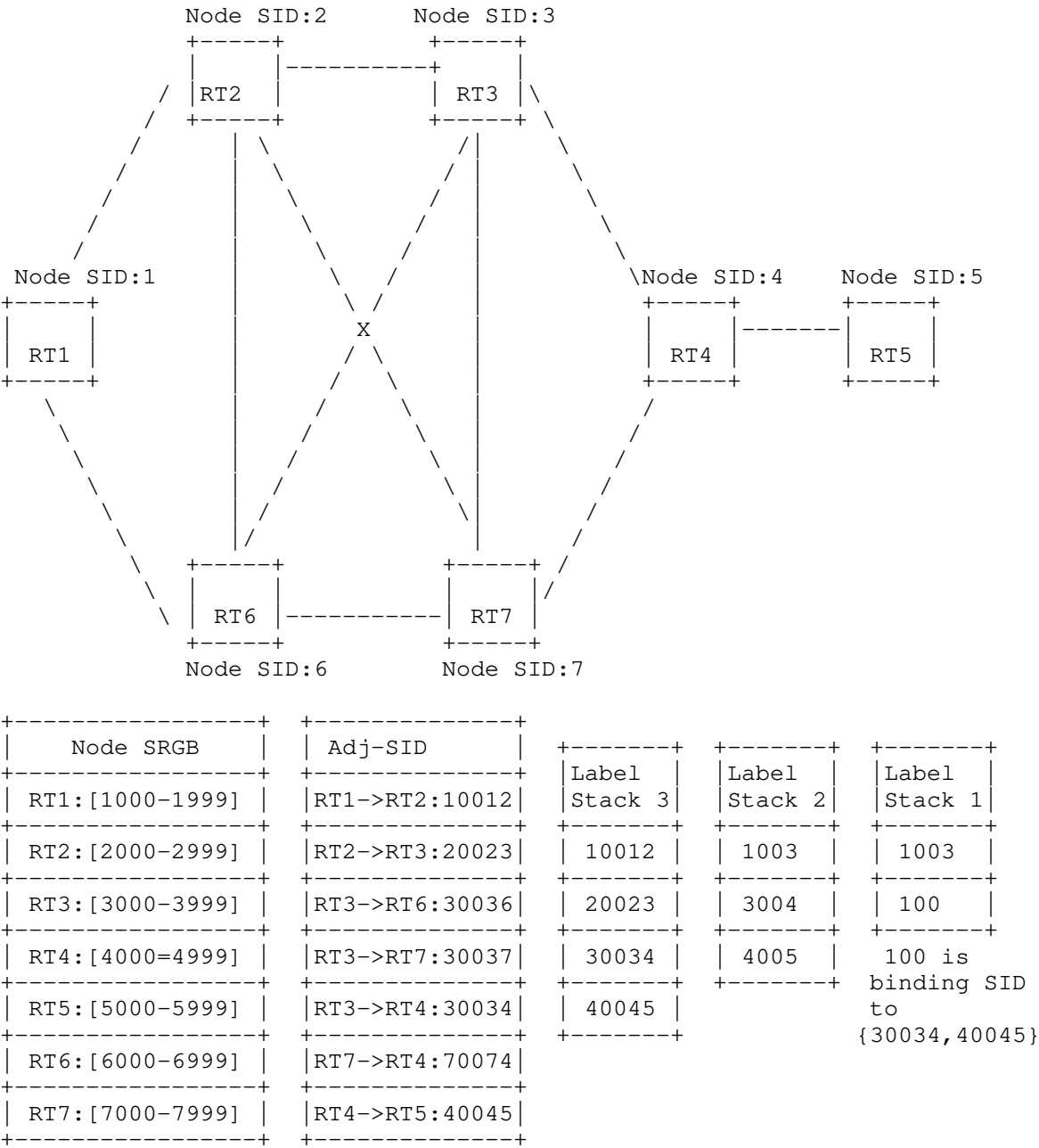


Figure 1: Topology of SR-TE Path

4.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do SR proxy forwarding for its neighboring nodes such as RT3, it must advertise this capability. When RT3 fails, RT2 needs to maintain its SR proxy forwarding capability for a period of time. When the proxy forwarding table corresponding to the fault node is deleted, the capability is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node SID advertised by RT3 and the proxy forwarding capability for RT3 advertised by RT2. When RT3 is normal, the nodes prefer prefix/node SID. When the RT3 fails, the proxy prefix/node SIDs of RT3 for RT2 is preferred.

For binding-SID 100, which is associated with segment list {30034, 40045}, RT3 advertises the binding (i.e., 100 bond to {30034, 40045}) to its neighbors RT2, RT4 and RT7. RT2 as PLR uses the binding to build an entry for proxy forwarding for binding-SID 100 in its Proxy Forwarding Table for RT3. The entry is used when RT3 fails.

4.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

- 1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;
- 2: Every adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.
- 3: Every binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adjacency-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. RT2 calculates the proxy forwarding table for RT3, as shown in Figure 2.

In-label	SRGBDiffValue	Next Label	Action	Map Label
2003	-1000	30034	Fwd to RT4	2004
		30036	Fwd to RT6	2006
		30037	Fwd to RT7	2007
		100	Swap to { 30034, 40045 }	

Figure 2: RT2's Proxy Forwarding Table for RT3

4.3. Proxy Forwarding for Binding Segment

This Section shows through example how a proxy node uses the SR proxy forwarding mechanism to forward traffic to the destination node when a node fails and the next segment of label stack is a binding-SID.

As shown in Figure 1, Label Stack 1 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-SID, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.
- c. RT2 uses Binding-SID:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.
- d. RT2 swaps Binding-SID:100 to Segment list {30034, 40045}, and uses the 30034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.
- e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.

f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

5. Security Considerations

The extensions to OSPF and IS-IS described in this document result in two types of behaviors in data plane when a node in a network fails. One is that for a node, which is a upstream (except for the direct upstream) node of the failed node along a SR-TE path, it continues to send the traffic to the failed node along the SR-TE path for an extended period of time. The other is that for a node, which is the direct upstream node of the failed node, it fast re-routes the traffic around the failed node to the direct downstream node of the failed node along the SR-TE path. These behaviors are internal to a network and should not cause extra security issues.

6. Acknowledgements

The authors would like to thank Peter Psenak, Acee Lindem, Les Ginsberg, Bruno Decraene and Jeff Tantsura for their comments to this work.

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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [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>>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.

7.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-08, 21 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-08.txt>>.

[I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", Work in Progress, Internet-Draft, draft-ietf-spring-segment-protection-sr-te-paths-03, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-protection-sr-te-paths-03.txt>>.

[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://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-22.txt>>.

Appendix A. Proxy Forwarding for Adjacency and Node Segment

This Section shows through example how a proxy node forward traffic to the destination node when a node fails and the next segment of label stack is an adjacency-SID or node-SID.

A.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 3 {10012, 20023, 30034, 40045} uses only adjacency-SIDs and represents the SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adjacency-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 pops top adjacency-SID 10012, and forwards the packet to RT2;
- b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The query result is 2004. RT2 uses 2004 as the incoming label to query the label forwarding table. The next hop is RT7, and the incoming label is changed to 7004.
- c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT7 forwards it to RT4, where the original path is rejoined.
- d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

A.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

- a. RT1 swaps label 1003 to out-label 2003 to RT3.
- b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure, RT2 pops label 2003.
- c. RT2 uses 3004 as the in-label to lookup Proxy Forwarding table, The value of Map Label calculated based on SRGBDiffValue is 2004. and the query result is forwarding the packet to RT4.
- d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.
- e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.
- f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China
Email: huzhibo@huawei.com

Huaimo Chen
Futurewei
Boston, MA,
United States of America
Email: Huaimo.chen@futurewei.com

Junda Yao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing
100095
China
Email: yaojunda@huawei.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA, 94089
United States of America
Email: cbowers@juniper.net

Yongqing
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou
510000
China
Email: zhuyq8@chinatelecom.cn

Yisong
China Mobile
510000
China

Email: liuyisong@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2022

Z. Li
T. Sun
China Mobile
W. Cheng
J. Wang
Centec
July 8, 2021

S-BFD over SRv6
draft-li-sbfd-over-srv6-00

Abstract

Bidirectional Forwarding Detection (BFD) can be used to monitor paths between node. Seamless BFD (S-BFD) provides a simplified mechanism which is suitable for monitoring of paths that are setup dynamically and on a large scale network. This draft describes a method to simplify the implementation of S-BFD over SRv6 by using SRH.flag to instruct the S-BFD peer node to do reverse operation of SRv6 SID list.

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 .

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2022.

Copyright Notice

Copyright (c) 2021 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	2
2. Motivation for Proposing S-BFD over SRv6	2
3. The benefits of S-BFD over SRv6	4
4. Future Considerations and Enhancements of S-BFD over SRv6 . .	5
5. Security Considerations	5
6. IANA Considerations	5
Authors' Addresses	5

1. Introduction

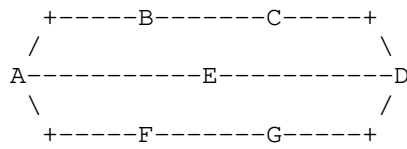
With the increasing adoption of segment routing (SR) technology, ISPs have upgraded their networks seamlessly from MPLS to SR MPLS, and their next goal might be the overall upgrading of the IPv6 underlay network forwarding plane.

We hope to implement BFD over SRv6 while retaining the bidirectional detection capabilities of traditional BFD, rather than using asymmetrical path detection only. Another problem relates to the bidirectional detection mechanism in BFD over SRv6, Using SR Policy or using TLV to carry the return path brings extra load to the message parsing depth on existing SRv6 device.

In order to accelerate applying BFD in SRv6 networks, this paper proposed a S-BFD over SRv6 implementation solution.

2. Motivation for Proposing S-BFD over SRv6

As shown in the figure below, the BFD initiator is A and the peer node is D, while bfd packets forwarding from A to D via the path: A->B->C->D, and return via the path: D->C->B->A.



Forward Paths: A-B-C-D

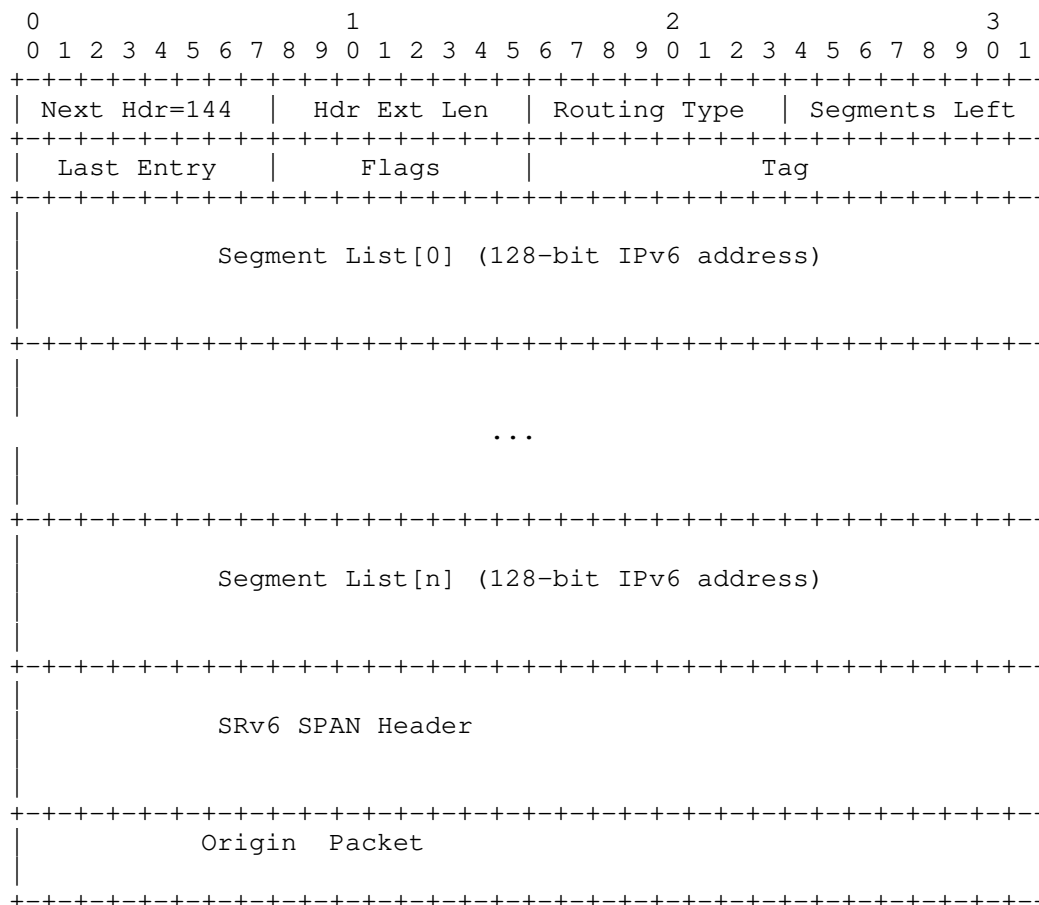
Return Paths: D-C-B-A

Traditional BFD in SRv6 Data Plane

SRv6 SID operations on the initial node A: The SRv6 SID list {A, B, C, D} is pushed into Node A.

SRv6 SID operations on the terminal node D: The SRv6 SID list {A, B, C, D} is swapped in Node D, and the updated SRv6 SID list is : {D, C, B, A}, and the Last Entry, Segment Left, and other fields are updated. Return Path: D->C->B->A.

As shown in the figure below, the length of the flags field in the SRH header is 8-bit. This draft uses the left third bit (0|0|R|0|0|0|0|0) to represent the reverse operation of the SRv6 SID list.



The reverse operations for S-BFD of SRv6 Flag

BFD peer node D check if SRH.Flags[5] == 1, it means that this device requires the reverse operation of the SRv6 SID list.

3. The benefits of S-BFD over SRv6

This solution does not need to use the SRv6 Policy to add length of the SID list or to carry the SID list of the return path by TLV. It only needs to support reverse SRv6 SID in the reflector node to solve the issue of S-BFD over SRv6 described in the previous.

4. Future Considerations and Enhancements of S-BFD over SRv6

In future versions of this paper, we will also consider the compatibility of using compressed IDs in SRv6, such as seamlessly merging S-BFD over G-SRv6. Furthermore, there will be no effect on intermediate nodes within the SRv6 network and it only requires S-BFD reflector support the SID reverse operation.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

Authors' Addresses

Zhiqiang Li
China Mobile
Beijing 100053
China

Email: lizhiqiangyjy@chinamobile.com

Tao Sun
China Mobile
Beijing 100053
China

Email: suntao@chinamobile.com

Wei Cheng
Centec
Suzhou 215000
China

Email: chengw@centecnetworks.com

Junjie Wang
Centec
Suzhou 21500
China

Email: wangjj@centecnetworks.com

IDR WG
Internet-Draft
Intended status: Standards Track
Expires: 22 June 2022

Y. Liu
S. Peng
ZTE
19 December 2021

BGP Extensions of SR Policy for Path Protection
draft-lp-idr-sr-path-protection-02

Abstract

This document proposes extensions of BGP to provide protection information of segment lists within a candidate path when delivering SR policy. And it also extends BGP-LS to provide some extra information of the segment list in the advertisement.

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 22 June 2022.

Copyright Notice

Copyright (c) 2021 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. BGP Extensions for Advertising Segment List	3
2.1. Extensions of Segment List sub-TLV	3
2.2. List Identifier Sub-TLV	4
2.2.1. List Protection Sub-TLV	4
3. BGP-LS Extensions for Distributing Segment List States	7
4. IANA Considerations	7
4.1. New Registry: Flag Field of Segment List sub-TLV	7
4.2. Existing Registry: BGP Tunnel Encapsulation Attribute sub-TLVs	7
4.3. New Registry: List Identifier Sub-TLVs	8
4.4. Existing Registry: Flag Field of SR Segment List TLV . .	8
5. Security Considerations	8
6. References	8
6.1. Normative References	8
6.2. Informative References	9
Authors' Addresses	9

1. Introduction

Segment Routing [RFC8402] allows a headend node to steer a packet flow along any path. [I-D.ietf-spring-segment-routing-policy] details the concept of SR Policy and steering into an SR Policy. An SR Policy is a set of candidate paths, each consisting of one or more segment lists. The headend of an SR Policy may learn multiple candidate paths for an SR Policy.

Candidate path can be used for path protection, that is, the lower preference candidate path may be designated as the backup for a specific or all (active) candidate path(s). Backup candidate path provide protection only when all the segment lists in the active CP are invalid.

If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with weight for weighted load balancing.

The protection mechanism for SR Policy is not flexible enough. For example, there're three segment lists(SL1, SL2, SL3) in candidate path 1, it may be desired that SL1 and SL2 are the primary path, SL3 are the backup path for SL1 and will be active only when SL1 fails.

[I-D.ietf-pce-multipath] proposes extensions to PCEP to specify the protection relationship between segment lists in the candidate path.

[I-D.ietf-idr-segment-routing-te-policy] specifies BGP extensions for the advertisement of SR Policies and each candidate path is carried in an NLRI. This document proposes extensions of BGP in order to provide protection information of segment lists when delivering SR policy.

[I-D.ietf-idr-te-lsp-distribution] describes a mechanism to collect the SR policy information that is locally available in a node and advertise it into BGP Link State (BGP-LS) updates. This document also extends it to provide some extra information of the segment list in a candidate path in the BGP-LS advertisement.

1.1. 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].

2. BGP Extensions for Advertising Segment List

2.1. Extensions of Segment List sub-TLV

Segment List sub-TLV is introduced in [I-D.ietf-idr-segment-routing-te-policy] and it includes the elements of the paths (i.e., segments).

This document introduces a one-bit flag in the RESERVED field.

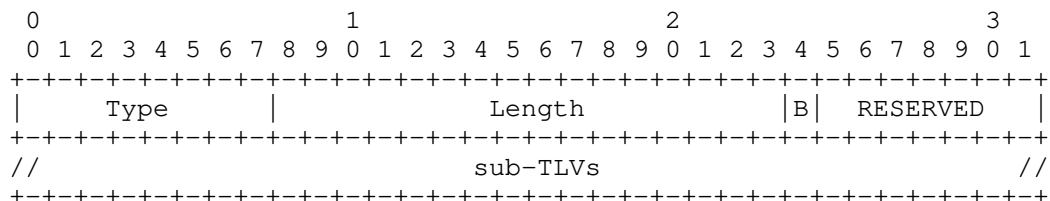


Figure 1: Segment List sub-TLV

B-Flag(Backup Flag): one bit. When set to 0, it indicates that the segment list acts as the active member in the candidate path. When set to 1, it indicates that the segment list acts as the backup path in the candidate path.

Using segment lists for path protection can be compatible with using candidate paths. When a path fails, the backup segment list within the same candidate path is used preferentially for path protection. If the backup list is also invalid, then other candidate path can be enabled for protection.

2.2. List Identifier Sub-TLV

This document introduces a new sub-sub-tlv of Segment List sub-TLV, where,

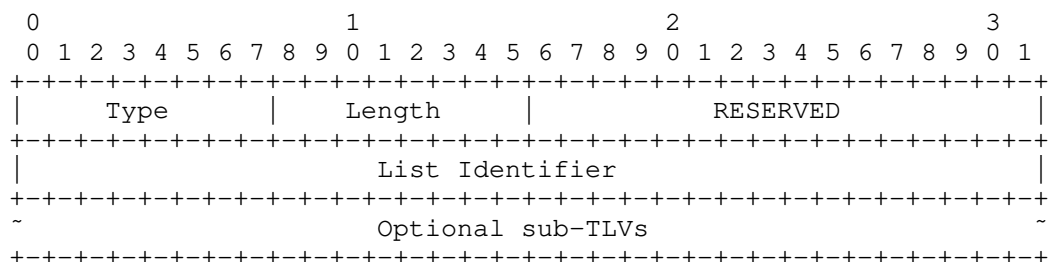


Figure 2: List Identifier Sub-TLV

- * Type: 1 octet. TBD.
- * Length: 1 octet, specifies the length of the value field not including Type and Length fields.
- * RESERVED: 2 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- * List Identifier: 4 octets. It is the identifier of the corresponding segment list, so that the segment list can be operated according to the specified Segment List identifier.
- * This sub-TLV is optional and it MUST NOT appear more than once inside the Segment List sub-TLV.

2.2.1. List Protection Sub-TLV

The List Protection Info sub-TLV is an optional sub-TLV of List Identifier sub-TLV, where:

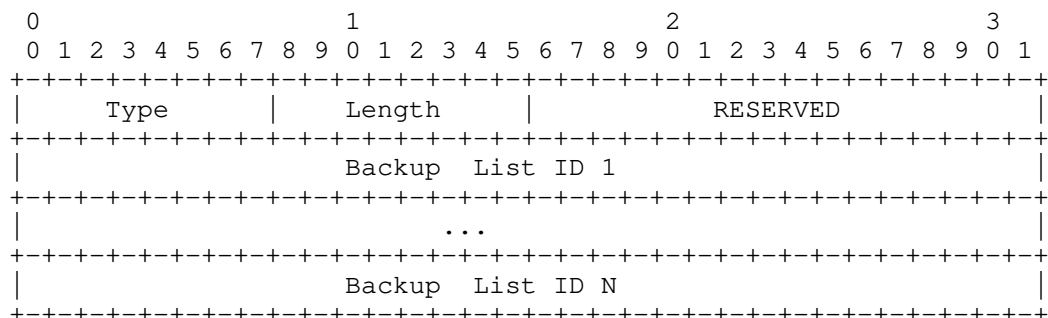


Figure 3: List Protection Info Sub-TLV

- * Type: 1 octet. TBD.
- * Length: 1 octet, specifies the length of the value field not including Type and Length fields.
- * RESERVED: 2 octet of reserved bits. SHOULD be unset on transmission and MUST be ignored on receipt.
- * Backup List ID: 4 octets. It is the List Identifier of the backup segment list that protects this segment list. If there're multiple backup paths, the list ID of each path should be included in the TLV.

As defined in [I-D.ietf-idr-segment-routing-te-policy], the SR Policy encoding structure is as follows:

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:

- Tunnel Encaps Attribute (23)
 - Tunnel Type: SR Policy
 - Binding SID
 - Preference
 - Priority
 - Policy Name
 - Explicit NULL Label Policy (ENLP)
 - Segment List
 - Weight
 - Segment
 - Segment
 - ...
 - Segment List
 - ...
 - ...

The new SR Policy encoding structure with List Identifier sub-TLV is shown as below:

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
Tunnel Encaps Attribute (23)

- Tunnel Type: SR Policy
- Binding SID
- SRv6 Binding SID
- Preference
- Priority
- Policy Name
- Policy Candidate Path Name
- Explicit NULL Label Policy (ENLP)
- Segment List
 - List Identifier
 - List Protection Info
 - Weight
 - Segment
 - Segment
 - ...
- Segment List
- ...
- ...

3. BGP-LS Extensions for Distributing Segment List States

[I-D.ietf-idr-te-lsp-distribution] describes a mechanism to collect the SR Policy information that is locally available in a node and advertise it into BGP Link State (BGP-LS) updates. The SR Policy information includes status of the candidate path, e.g, whether the candidate path is administrative shut or not.

SR Segment List TLV is defined in [I-D.ietf-idr-te-lsp-distribution] to report the SID-List(s) of a candidate path. Figure 4 shows the flags in SR Segment List TLV.

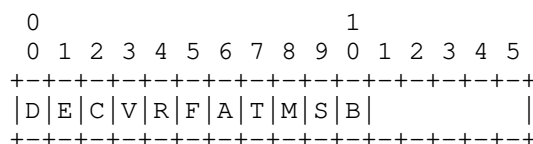


Figure 4: Flag Field of SR Segment List TLV

The D,E,C,V,R,F,A,M flags are defined in [I-D.ietf-idr-te-lsp-distribution].

This document introduces two new flags, where,

- * S-Flag : Indicates the segment list is in administrative shut state when set.
- * B-Flag : Indicates the segment list is the backup path within the candidate path when set, otherwise it is the active path.

4. IANA Considerations

4.1. New Registry: Flag Field of Segment List sub-TLV

This document introduces a one-bit flag field in the Segment List sub-TLV [I-D.ietf-idr-segment-routing-te-policy] for the Backup Flag (B-Flag).

4.2. Existing Registry: BGP Tunnel Encapsulation Attribute sub-TLVs

This document defines a new sub-TLV in the registry "SR Policy List Sub-TLVs" [I-D.ietf-idr-segment-routing-te-policy] to be assigned by IANA:

Codepoint	Description	Reference
TBD	List Identifier Sub-TLV	This document

4.3. New Registry: List Identifier Sub-TLVs

This document requests the creation of a new registry called "List Identifier Sub-TLVs" under the "BGP Tunnel Encapsulation" registry. Following initial Sub-TLV codepoint are assigned by this document.

Codepoint	Description	Reference
TBD	List Protection Sub-TLV	This document

4.4. Existing Registry: Flag Field of SR Segment List TLV

This document requests bit 9 and bit 10 in the flag field of "SR Segment List TLV" [I-D.ietf-idr-te-lsp-distribution] under the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" registry.

Bit	Description	Reference
9	Administrative Shut State Flag(S-Flag)	This document
10	Backup Path State Flag(B-Flag)	This document

5. Security Considerations

Procedures and protocol extensions defined in this document do not affect the security considerations discussed in [I-D.ietf-idr-segment-routing-te-policy] and [I-D.ietf-idr-te-lsp-distribution].

6. References

6.1. Normative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filisfilis, C., Talaulikar, K., Mattes, P., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-segment-routing-te-policy-14, 10 November 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-segment-routing-te-policy-14>>.

[I-D.ietf-idr-te-lsp-distribution]
Previdi, S., Talaulikar, K., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", Work in Progress, Internet-Draft, draft-ietf-idr-te-lsp-distribution-16, 22 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-te-lsp-distribution-16>>.

[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-14, 25 October 2021,
<[https://datatracker.ietf.org/doc/html/draft-ietf-spring-
segment-routing-policy-14](https://datatracker.ietf.org/doc/html/draft-ietf-spring-segment-routing-policy-14)>.

[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-pce-multipath]
Koldychev, M., Sivabalan, S., Saad, T., Beeram, V. P.,
Bidgoli, H., Yadav, B., and S. Peng, "PCEP Extensions for
Signaling Multipath Information", Work in Progress,
Internet-Draft, draft-ietf-pce-multipath-03, 25 October
2021, <[https://datatracker.ietf.org/doc/html/draft-ietf-
pce-multipath-03](https://datatracker.ietf.org/doc/html/draft-ietf-pce-multipath-03)>.

[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

Yao Liu
ZTE
Nanjing
China

Email: liu.yao71@zte.com.cn

Shaofu Peng
ZTE
Nanjing
China

Email: peng.shaofu@zte.com.cn

SPRING
Internet-Draft
Intended status: Informational
Expires: January 9, 2022

R. Bonica
Juniper
W. Cheng
China Mobile
D. Dukes, Ed.
Cisco Systems
W. Henderickx
Nokia
C. Li
Huawei
P. Shaofu
ZTE
C. Xie
China Telecom
July 08, 2021

Compressed SRv6 SID List Analysis
draft-srcompdt-spring-compression-analysis-02

Abstract

Several mechanisms have been proposed to compress the SRv6 SID list. This document analyzes each mechanism with regard to the requirements stated in the companion requirements document.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2022.

Copyright Notice

Copyright (c) 2021 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	3
2. SRv6 Compression Requirements	3
2.1. Encapsulation Header Size	4
2.1.1. Reference Scenarios	4
2.2. Forwarding Efficiency	5
2.2.1. Headers Parsed	5
2.2.2. Lookups Performed (LKU)	7
2.3. State Efficiency	8
3. SRv6 Specific Requirements	10
3.1. SRv6 Based	10
3.2. Functional Requirements	11
3.2.1. SRv6 Functionality	11
3.2.2. Heterogeneous SID Lists	14
3.2.3. SID List Length	15
3.2.4. SID Summarization	15
3.3. Operational Requirements	15
3.3.1. Lossless Compression	16
3.3.2. Preservation of non-routing information	16
3.3.3. Address Planning	16
3.4. Scalability Requirements	17
3.4.1. Compression Levels	18
4. Protocol Design Requirements	18
4.1. SRv6 Base Coexistence	18
5. Security Requirements	19
5.1. Security Mechanisms	19
5.2. SR Domain Protection	19
6. Conclusions	19
7. Normative References	21
Appendix A. Encapsulation analysis	24
A.1. CRH note	24
A.2. Analysis results	25
Authors' Addresses	27

1. Introduction

The following mechanisms are proposed to compress the SRv6 SID list:

- o CSID - [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] - Describes two new SRv6 SID flavors, a combination of SID flavors from [I-D.filsfils-spring-net-pgm-extension-srv6-usid] and [I-D.cl-spring-generalized-srv6-for-cmpr]
- o CRH - [I-D.bonica-6man-comp-rtg-hdr] - Requires two new routing header types and a label mapping technique.
- o VSID - [I-D.decreaene-spring-srv6-vlsid] - Defines a set of SID behaviors to access smaller SIDs within the SR header.
- o UIDSr - [I-D.mirsky-6man-unified-id-sr] - Extends the SRH to carry MPLS labels or IPv6 addresses.

This document analyzes each mechanism against the requirements stated in [I-D.srcompdt-spring-compression-requirement]. Each section of this document corresponds to a similarly named section in [I-D.srcompdt-spring-compression-requirement]. Each section reiterates corresponding requirements and analyzes each proposal against the those requirements.

The terms compression mechanism, compression solution, and compression proposal are used interchangeably within this document.

2. SRv6 Compression Requirements

An SR domain consisting of 3 sub-domains is shown to illustrate the scenarios associated with encapsulation header size, forwarding efficiency and state efficiency.

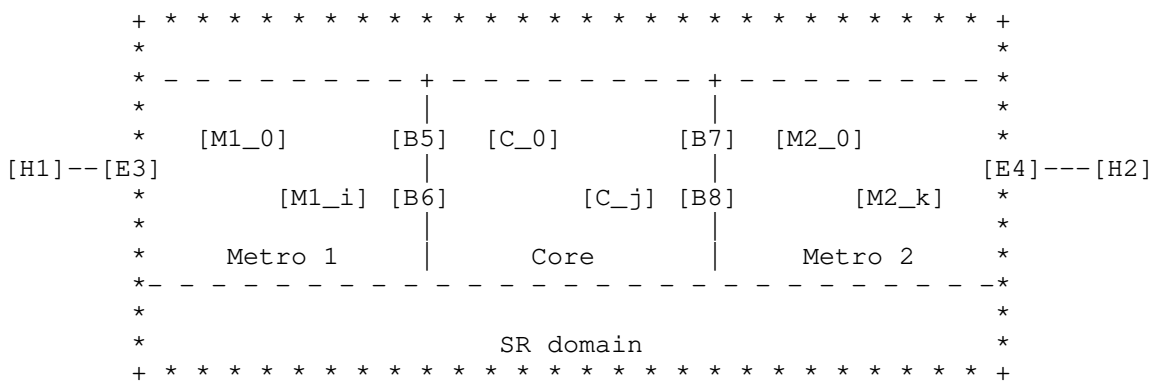


Figure 1: Sample SR Domain

- o H1 and H2 are hosts outside the SR domain

- o E3 and E4 are SR domain edge routers
- o Metro 1, Core and Metro 2 are sub-domains with independent IGP instances
- o B5 and B6 are border routers between the Metro 1 and Core
- o B7 and B8 are border routers between the Metro 2 and Core
- o M1_1..M1_i are routers in Metro 1
- o C_1..C_j are routers in Core
- o M2_1..M2_k are routers in Metro 2
- o If Metro and Core are different AS's the border routers (B5 to B8) may be replaced by pairs of ASBRs
- o Flexible algorithms may be deployed within each sub-domain

2.1. Encapsulation Header Size

The compression proposal MUST reduce the size of the SRv6 encapsulation header.

Encapsulation header size is evaluated against a set of reference scenarios.

2.1.1. Reference Scenarios

A service provider offers a VPN service with underlay optimization in the SR domain.

- o Hosts H1 and H2 are located in two different sites of a VPN customer.
- o Edge nodes E3 and E4 encapsulate/decapsulate traffic between H1 and H2 to provide the VPN service.
- o The encapsulation consists of a VPN SID (V) (eg END.DT etc) and an SR policy with between 0 and 15 transport segments (T) (eg END or END.X)
- o The SR domain has a block size (B) of 48 bits
- o These independent variables are used to uniquely identify each scenario. For example

- * A scenario with 48bit block size, 3 transport segments and a VPN segment is named 48B.3T.V

Proposals are evaluated against the set of scenarios to calculate the encapsulation in octets (E) and the encapsulation savings (ES) as a fraction of the SRv6 base encapsulation in octets.

E and ES were evaluated for:

- o each proposal in two variants
 - * 16-bit SID

- * 32-bit SID
- o 48-bit SRv6 block, 0 to 15 transport segments and a VPN segment (expressed in short form as 48B.0-15T.V)

The average encapsulation savings for each proposal is shown below. The complete analysis is recorded in Appendix:

16-bit SIDs	CSID	CRH	CRH+TPF	VSID	UIDSR
Average ES	54.3%	54.2%	50.4%	51.6%	49.2%

Table 1: Average ES, 16-bit SIDs, 48B.0-15T.V

32-bit SIDs	CSID	CRH	CRH+TPF	VSID	UIDSR
Average ES	42.5%	45.5%	43.2%	45.5%	42.5%

Table 2: Average ES, 32-bit SIDs, 48B.0-15T.V

E and ES are also evaluated for 32bit and 64bit SRv6 block sizes. The CSID 16-bit ES averages 57.4% for 32-bit blocks and 49.9% for 64-bit blocks, other proposals are unchanged.

Conclusion: All proposals meet the requirement to reduce the size of the SRv6 encapsulation header. Variances between proposals are negligible.

2.2. Forwarding Efficiency

The compression proposal SHOULD minimize the number of required hardware resources accessed to process a segment.

2.2.1. Headers Parsed

Forwarding efficiency is calculated against the reference scenarios above, recording and summarizing the differences in header parsing for different segment lists.

The following tables indicate the number of headers parsed for each proposal.

16-bit	CSID	CRH	CRH+TPF	VSID	UIDSR
PRS (48B.0T) .V)	IPv6	IPv6	IPv6	IPv6	IPv6
PRS (48B.1-4T) .V)	IPv6	IPv6 CRH	IPv6 CRH	IPv6 SRH	IPv6 SRH
PRS (48B.5-15T) .V)	IPv6 SRH	IPv6 CRH	IPv6 CRH	IPv6 SRH	IPv6 SRH

Table 3: Headers parsed on non-decapsulating SR segment endpoint nodes, 16-bit SIDs, 48B.0-15T.V

16-bit	CSID	CRH	CRH+TPF	VSID	UIDSR
PRS (48B.0T) .V)	IPv6	IPv6	IPv6	IPv6	IPv6
PRS (48B.1-4T) .V)	IPv6	IPv6 CRH	IPv6 CRH TPF	IPv6 SRH	IPv6 SRH
PRS (48B.5-15T) .V)	IPv6 SRH	IPv6 CRH	IPv6 CRH TPF	IPv6 SRH	IPv6 SRH

Table 4: Headers parsed on decapsulating SR segment endpoint nodes, 16-bit SIDs, 48B.0-15T.V

32-bit	CSID	CRH	CRH+TPF	VSID	UIDSR
PRS (48B.0T.V)	IPv6	IPv6	IPv6	IPv6	IPv6
PRS (48B.1-15T.V)	IPv6 SRH	IPv6 CRH	IPv6 CRH	IPv6 SRH	IPv6 SRH

Table 5: Headers parsed on non-decapsulating SR segment endpoint nodes, 32-bit SIDs, 48B.0-15T.V

32-bit	CSID	CRH	CRH+TPF	VSID	UIDSR
PRS (48B.0T.V)	IPv6	IPv6	IPv6	IPv6	IPv6
PRS (48B.1-15T.V)	IPv6 SRH	IPv6 CRH	IPv6 CRH TPF	IPv6 SRH	IPv6 SRH

Table 6: Headers parsed on decapsulating SR segment endpoint nodes, 32-bit SIDs, 48B.0-15T.V

Conclusion: Overall, the CSID parses the fewest headers. When per packet state is processed per segment, CSID, VSID and UIDSR proposals may include it in the routing header, CRH may include it in a destination option preceding the CRH.

2.2.2. Lookups Performed (LKU)

Some proposals require a different number of lookups per packet, depending on the active segment in a segment list.

An implementation may perform lookups as longest prefix match (LPM) or exact match (EM). CSID, VSID and UIDSR describe SRv6 SID lookup from the IPv6 destination address as an LPM, however an implementation may use either an LPM or EM lookup for SRv6 SIDs. CRH implementations must always uses an exact match for CRH SID lookups.

The following table describes the number of lookups per proposal per segment type.

	CSID	CRH	VSID	UIDSR
Adjacency and VPN Segments	LPM (a)	LPM (a) EM (b) EM (b,c)	LPM (a)	LPM (a)
Prefix Segments	LPM (a) LPM (d)	LPM (a) EM (b)	LPM (a) LPM (d)	LPM (a) LPM (d)

Table 7: Lookups

- o [a] On active SID, appearing in the IPv6 Destination address
- o [b] On SID in CRH header

- o [c] This lookup is required only when the IPv6 next hop node is not non-CRH aware
- o [d] On next SID, appearing in the IPv6 destination address

Note: [I-D.filsfils-spring-net-pgm-extension-srv6-usid] Section 5 describes an optional local implementation to reduce CSID 16-bit lookups, in some cases, by adding local forwarding state. The analysis of this implementation option is not included in this version of the document.

Conclusion: CSID, VSID, and UIDSR require a single lookup to process an adjacency or VPN segment. CRH always requires 2 lookups for VPN segments, and 2 and sometimes 3 lookups for adjacency segments. All proposals require two lookups to process a prefix segment and the next segment.

2.3. State Efficiency

The compression proposal SHOULD minimize the amount of additional forwarding state stored at a node.

State efficiency is analyzed in a sub-domain of the SR domain, with the following parameters:

- o N: the number of SRv6 nodes in the sub-domain
- o I: the number of IGP algorithms [I-D.ietf-lsr-flex-algo] configured
- o A: the number of local adjacency SIDs at a node
- o D: the number of attached SR sub-domains at a border node
- o V: the number of VPN services at edge nodes

For a sub-domain consisting of:

- o 1000 SRv6 nodes (N=1000) with some number of non-SRv6 nodes
- o 2 IGP algorithms (I=2)
- o 100 adjacencies per SRv6 node (A=100)
- o up to 10 attached sub-domains per border node (D=10)
- o 1000 VPN service segments per edge (V=1000)

The number of forwarding entries at a node is calculated for any node, a border node, and an edge node.

UIDSR, CSID and VSID require the following entries:

- o a FIB entry for the node's prefix segment (1), per algorithm (I=2).
- o a FIB entry per local adjacency SID (A=100) **Notel
- o At border nodes (or any SRv6 nodes) either:

- * A.1) a FIB entry per domain ($D=10$) to swap the IPv6 destination address prefix.
- * A.2) no additional FIB entries, and the SR source places a 128-bit SID in the segment list of a packet if needed.
- o At edge nodes, a FIB entry per VPN segment ($V=1000$)

CRH requires:

- o a CFIB entry per CRH node per IGP algorithm for local and remote prefix segments ($N*I=2000$)
- o a CFIB entry per local adjacency segment ($A=100$) **Notel
- * When non-CRH adjacent nodes are present, additional state is required for CRH as per [I-D.bonica-6man-comp-rtg-hdr] Appendix B (note, only the second option in the appendix is considered feasible due to state explosion)
 - + B.1) Up to one CFIB entry per next endpoint and an additional CFIB entry per adjacency to support non-CRH adjacent endpoints, assuming IP flex algo is not implemented on non-CRH nodes ($I=1$) ($(N+A)*I=1200$).
- o At border nodes, assuming two inter-domain links per adjacent domain for redundancy, additional state is required as per [I-D.bonica-6man-comp-rtg-hdr] Appendix B (note, only the second option in the appendix is considered feasible due to state explosion):
 - * C.1) In a common CRH network topology, the remote sub-domain borders support CRH: a CFIB entry per CRH node per IGP algorithm for local and remote prefix segments ($N*I$) plus a CFIB entry per local adjacency segment (A) plus a CFIB entry per connected remote border router (20) ($N*I+A+20=2120$).
 - * C.2) In a poorly designed CRH network topology, the remote sub-domain borders do not support CRH: a CFIB entry per unique endpoint ($N*D*I$), plus a CFIB entry per local adjacency segment (A), assuming IP flex algo is not implemented on non-CRH border domain ($I=1$), plus inter-domain adjacency (20) ($N*D*I+2=10120$).
- o At edge nodes, $V=1000$ entries for SRv6 based VPN SIDs and another $V=1000$ entries for CFIB and TPF VPN SIDs.

**Notel: there may be additional adjacency SIDs for protected, unprotected, and per algorithm adjacencies, resulting in some multiple of A . This is common for all compression proposals.

16-bit and 32-bit	CSID	CRH	VSID	UIDSR
S(N1000,I2,A100,D10)	102 A.1:112 A.2:102	2100 B.1:3300 C.1:2120 C.2:10120	102 A.1:112 A.2:102	102 A.1:112 A.2:102
S(V1000)	1000	2000	1000	1000

Table 8: Forwarding State Maintained

Conclusion: CSID, VSID and UIDSR minimize forwarding state stored at a node. CRH moves per segment state from the packet to the FIB.

3. SRv6 Specific Requirements

3.1. SRv6 Based

A solution to compress SRv6 SID Lists SHOULD be based on the SRv6 architecture, control plane and data plane. The compression solution MAY be based on a different data plane and control plane, provided that it derives sufficient benefit.

This section records the use of SRv6 standards for compression.

	CSID	CRH	VSID	UIDSR
U.RFC8402	Yes	Yes - update required for SRv6 data plane	Yes	Yes
U.RFC8754	Yes	No	Yes - update required for segments left	Yes - update for flags and segments left
U.PGM	Yes	No	Yes - update required for SID behaviors	Yes
U.IGP	Yes	No	Yes	Yes - additional extensions
U.BGP	Yes	No	Yes	Yes
U.POL	Yes	No	Yes	Yes
U.BLS	Yes	No	Yes	Yes - additional extensions
U.SVC	Yes	No	Yes	Yes
U.ALG	Yes	Yes - Adds IP flex Algo	Yes	Yes
U.OAM	Yes	No	Yes	Yes

Table 9: SRv6 Based

Conclusion: CSID is SRv6 based, requiring no updates to existing SRv6 standards, VSID and UIDSR require updates. CRH is not strictly based on SRv6 but is able to provide equivalent functionality.

3.2. Functional Requirements

3.2.1. SRv6 Functionality

A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement ensures no SRv6 functionality is lost. It is particularly important to understand how a proposal, as evaluated in section "SRv6 Based", provides this functionality.

Functional requirements and the drafts defining how a proposal provides the functionality are documented in the table below.

Draft reference Abbreviations	
RFC8986:	[RFC8986]
SRV6POL:	[I-D.ietf-spring-segment-routing-policy]
SRV6EXT:	[I-D.ietf-lsr-isis-srv6-extensions]
SRV6BGPSVC:	[I-D.ietf-bess-srv6-services]
SRV6BGPLS:	[I-D.ietf-idr-bgpls-srv6-ext]
SRV6SVCP:	[I-D.ietf-spring-sr-service-programming]
SRV6OAM:	[I-D.ietf-6man-spring-srv6-oam]
SRV6FLEXALG:	[I-D.ietf-lsr-flex-algo]
SRV6TILFA:	[I-D.ietf-rtgwg-segment-routing-ti-lfa]
RFC8402:	[RFC8402]
RFC8754:	[RFC8754]
CRH:	[I-D.bonica-6man-comp-rtg-hdr]
VSID:	[I-D.decraene-spring-srv6-vlsid]
UIDSR:	[I-D.mirsky-6man-unified-id-sr]
IPFLEXALG:	[I-D.ietf-lsr-ip-flexalgo]
CRHEXT:	[I-D.bonica-lsr-crh-isis-extensions]
SRM6BGPSVC:	[I-D.ssangli-bess-bgp-vpn-srm6]
CSID:	[I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc]

Abbreviations

	CSID	CRH	VSID	UIDSR
F.SID	RFC8402	CRH	RFC8402	RFC8402 1
F.Scop e	RFC8402	CRH	RFC8402	RFC8402 1
F.PFX	RFC8402, RFC8986, CSID adds an END SID flavor	CRH	RFC8402, RFC8986, VSID updates the End behavior	RFC8402, RFC8986 with new flavor 1
F.ADJ	RFC8402, RFC8986, CSID adds an END.X flavor	CRH	RFC8402, RFC8986, VSID updates the End.X behavior	RFC8402, RFC8986 with new flavor 1
F.BIND	RFC8402, RFC8986	CRH	RFC8402, RFC8986, VSID updates the End.B	RFC8402, RFC8986 with new flavor 1

F.PEER	RFC8402, RFC8986, CSID adds an END.X. flavor	CRH	behaviors RFC8402, RFC8986, VSID updates the End.X	RFC8402, RFC8986 with new flavor 1,2
F.SVC	RFC8986	CRH	behaviors RFC8986, VSID updates the service segment	RFC8986 1
F.ALG F.TILF A	SRV6FLEXALG SRV6TILFA	IPFLEXALG SRV6TILFA	behaviors SRV6FLEXALG SRV6TILFA	SRV6FLEXALG SRV6TILFA 3
F.SEC F.IGP F.BGP	RFC8754 SRV6EXT SRV6BGPSVC	CRH CRH-EXT SRM6BGPSVC	RFC8754 SRV6EXT SRV6BGPSVC	RFC8754 SRV6EXT 1,4 SRV6BGPSVC 1
F.POL	SRV6SRPOL	SRV6SRPOL update required	SRV6SRPOL	SRV6SRPOL
F.BLS	SRV6BGPLS	(specification required)	SRV6BGPLS and addition for VSID Length	SRV6BGPLS 5
F.SFC F.PING	SRV6SVCP SRV6OAM	CRH CRH	SRV6SVC SRV6OAM	SRV6SVCP 1 SRv6OAM

Table 10: SRv6 Functionality

1. UIDSR with Global Container SID + local index enhancement
2. draft-peng-spring-truncates-sid-inter-domain
3. For protections described in section 6.1.2.1, 6.1.2.2, and 6.2, to get next-next SID from SRH with the help of draft-pl-spring-compr-path-recover.
4. Need more extensions to advertise the capability of U-SID compression (32bits, 16bits, etc.). Note: Global Container SID + local index enhancement.
5. IGP extensions

Conclusion: CSID supports SRv6 functionality. CRH VSID and UID support SRv6 functionality or equivalent with some new specifications.

3.2.2. Heterogeneous SID Lists

The compression proposal SHOULD support a combination of compressed and non-compressed segments in a single path. As an example, a solution may satisfy this requirement without being SRv6 based by using a binding SID to impose an additional SRv6 header (IPv6 header plus optional SRH) with non-compressed SID.

	CSID	CRH	VSID	UIDSR
Heterogeneous SID Lists	Yes	Yes	Yes	Yes

Heterogeneous SID Lists

VSID require a binding SID with an additional SRv6 encapsulation to encode non-compressed segments in a single path. VSID changes the interpretation of the SRH Segments Left field, which makes it capable of carrying only compressed segments.

The CRH can include a binding SID that imposes a new IPv6 header with an SRH. This is required when the next segment endpoint in the path can process the SRH, but not the CRH. The next segment endpoint or a subsequent endpoint can execute decapsulation, removing the new IPv6 header and exposing the old one with its CRH. This is required because an IPv6 packet can carry only one routing header.

CSID and UIDSR permit the encoding of, and processing of, any combination of compressed or non-compressed segments in a segment list of an SRH.

CSID makes use of the SRH, without modification, to encode CSIDs as 128 bits, supporting the use of non-compressed segments within the SRH.

UIDSR modifies the interpretation of the SRH Segments Left field at segment endpoint nodes to allow variable segment lengths within a segment list.

Conclusion: All proposals support heterogeneous SID lists. CSID and UIDSR support heterogeneous SID lists in the SRH, while CRH and VSID require installation of binding SIDs at midpoint nodes.

3.2.3. SID List Length

The compression proposal MUST be able to represent SR paths that contain up to 16 segments.

	CSID	CRH	VSID	UIDSR
16 Segments	Yes	Yes	Yes	Yes

SID List Length

Conclusion: All proposals support segment lists of at least 16 segments.

3.2.4. SID Summarization

The solution MUST be compatible with segment summarization.

In inter sub-domain deployments with summarization:

- o Any node can reach any other node in another sub-domain via a prefix segment.
- o Prefixes are summarized for advertisement between domains.

Without summarization, border router SIDs must be leaked:

- o An additional global prefix segment is required for each domain border to be traversed.

	CSID	CRH	VSID	UIDSR
SID Summarization	Yes	No	Yes	Yes

SID Summarization

Conclusion: CSID, VSID and UIDSR support segment summarization, CRH does not.

3.3. Operational Requirements

3.3.1. Lossless Compression

A path traversed using a compressed SID list MUST always be the same as the path traversed using the uncompressed SID list if no compression was applied.

	CSID	CRH	VSID	UIDSR
Lossless Compression	Yes	Yes	Yes	Yes

Lossless Compression

Conclusion: All proposals provide lossless compression.

3.3.2. Preservation of non-routing information

The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node

	CSID	CRH	VSID	UIDSR
Preserves Non-Routing Information	Complies	Complies	Complies	Complies

Preservation of non-routing information

Conclusion: All proposals preserve non-routing information.

3.3.3. Address Planning

Description: Network operators require addressing plan flexibility, The compression mechanism MUST support flexible IPv6 address planning, it MUST support deployment by using GUA from different address blocks.

	CSID	CRH	VSID	UIDSR
Flexible Address Planning	Yes	Yes	Yes	Yes

Address Planning

All compression mechanisms provide the encapsulation savings described in Tables 1 and 2. CRH provides these encapsulation savings regardless of the IPv6 addressing scheme. CSID adds a CSID container, or one compressed SID (END.X with XPS behavior), for each change in locator block in a segment list. VSID (via XPS behavior) and UIDSR add one compressed SID for each change in locator block in the segment list.

The XPS behavior draws the new address block from the control plane. At the time of publication, this control plane behavior is undefined. Therefore XPS impact on the control plane is not entirely understood. While it may be possible to define these mechanisms without impacting the control plane, specifications are not yet available.

Conclusion: All proposals support flexible IPv6 planning.

3.4. Scalability Requirements

The compression proposal MUST be capable of representing 65000 adjacency segments per node.

The compression proposal MUST be capable of representing 1 million prefix segments per SID numbering space.

The compression proposal MUST be capable of representing 1 million services per node.

	CSID	CRH	VSID	UIDSR
Adjacency Segment Scale 65000	Yes	Yes	Yes	Yes
Prefix Segment Scale 1000000	Yes	Yes	Yes	Yes
Service Scale 1000000	Yes	Yes	Yes	Yes

Table 11: Scale Requirements

The 32-bit variants of all proposals support this scale of prefix, adjacency and services at a node.

Each proposals 16-bit variant supports a lesser scale. All proposals can encode 2^{16} prefix, adjacency and service segments. However, each proposal has various ways of supporting some larger scale per node if required.

CRH 16-bit proposes the encoding of the ultimate segment in a TPF destination option instead of the CRH. This supports 2^{32} service segments per node.

VSID proposes the combination of multiple vSIDs, by copying multiple SIDs to a destination address or looking up the next segment in the segment list. This supports more than 2^{16} adjacency and service segments per node.

CSID 16-bit variant uses a LIB for adjacency and service segments, the LIB allows local definition of SIDs longer than 16-bits when needed. This supports more than 2^{16} adjacency and service segments per node.

UIDSR defines a segment type that modifies the value of SRH segments left field to support variable segment sizes within the segment list. This supports 2^{32} adjacency and service segments per node.

Conclusion: All proposals meet scalability requirements.

3.4.1. Compression Levels

The compression proposal SHOULD be able to support multiple levels of compression.

	CSID	CRH	VSID	UIDSR
Multiple compression Levels	Yes	Yes	Yes	Yes

Compression Levels

Conclusion: All proposals support 16-bit and 32-bit SID variants.

4. Protocol Design Requirements

4.1. SRv6 Base Coexistence

The compression proposal MUST support deployment in SRv6 networks.

	CSID	CRH	VSID	UIDSR
SRv6 Base Coexistence	Yes	Yes	Yes	Yes

SRv6 Base Coexistence

Conclusion: All proposals can be deployed simultaneously with the SRv6 base solution.

5. Security Requirements

5.1. Security Mechanisms

The compression solution SHOULD be able to address security issues that it introduces, using existing security mechanisms.

	CSID	CRH	VSID	UIDSR
Security Mechanisms	Yes	Yes	Yes	Yes

Security Mechanisms

Conclusion: All proposals address security issues they may introduce with existing security mechanisms.

5.2. SR Domain Protection

A compression solution must not require nodes outside the SR domain to know SID values within the SR domain, and it must provide the ability to block nodes outside an SR domain from accessing SIDs.

	CSID	CRH	VSID	UIDSR
SR Domain Protection	Yes	Yes	Yes	Yes

SR Domain Protection

Conclusion: All proposals protect SIDs within the SR domain.

6. Conclusions

Encapsulation Header Size

- o All proposals meet the requirement to reduce the size of the SRv6 encapsulation header. Variances between proposals are negligible.

Forwarding Efficiency

- o Overall, the CSID parses the fewest headers. When per packet state is processed per segment, CSID, VSID and UIDSR proposals may include it in the routing header, CRH may include it in a destination option preceding the CRH.

- o CSID, VSID, and UIDSR require a single lookup to process an adjacency or VPN segment. CRH always requires 2 lookups for VPN segments, and 2 and sometimes 3 lookups for adjacency segments. All proposals require two lookups to process a prefix segment and the next segment.

State Efficiency

- o CSID, VSID and UIDSR minimize forwarding state stored at a node. CRH moves per segment state from the packet to the FIB.

SRv6 Based

- o CSID is SRv6 based, requiring no updates to existing SRv6 standards, VSID and UIDSR require updates. CRH is not strictly based on SRv6 but is able to provide equivalent functionality.

SRv6 Functionality

- o CSID supports SRv6 functionality. CRH VSID and UID support SRv6 functionality or equivalent with some new specifications.

Heterogeneous SID lists

- o All proposals support heterogeneous SID lists. CSID and UIDSR support heterogeneous SID lists in the SRH, while CRH and VSID require installation of binding SIDs at midpoint nodes.

SID List Length

- o All proposals support segment lists of at least 16 segments.

SID Summarization

- o VSID, CSID and UIDSR support segment summarization, CRH does not.

Operational Requirements

- o All proposals provide lossless compression.
- o All proposals preserve non-routing information.
- o All proposals support flexible IPv6 planning.

Scalability Requirements

- o All proposals meet scalability requirements.
- o All proposals support 16-bit and 32-bit SID variants.

Protocol Design Requirements

- o All proposals can be deployed simultaneously with the SRv6 base solution.

Security Requirements

- o All proposals address security issues they may introduce with existing security mechanisms.
- o All proposals protect SIDs within the SR domain.

7. Normative References

[I-D.bonica-6man-comp-rtg-hdr]

Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-24 (work in progress), January 2021.

[I-D.bonica-6man-vpn-dest-opt]

Bonica, R., Kamite, Y., Jalil, L., Zhou, Y., and G. Chen, "The IPv6 Tunnel Payload Forwarding (TPF) Option", draft-bonica-6man-vpn-dest-opt-15 (work in progress), February 2021.

[I-D.bonica-lsr-crh-isis-extensions]

Kaneriya, P., Shetty, R., Hegde, S., and R. Bonica, "IS-IS Extensions To Support The IPv6 Compressed Routing Header (CRH)", draft-bonica-lsr-crh-isis-extensions-04 (work in progress), March 2021.

[I-D.cl-spring-generalized-srv6-for-cmpr]

Cheng, W., Li, Z., Li, C., Clad, F., Liu, A., Xie, C., Liu, Y., and S. Zadok, "Generalized SRv6 Network Programming for SRv6 Compression", draft-cl-spring-generalized-srv6-for-cmpr-03 (work in progress), April 2021.

[I-D.decraene-spring-srv6-vlsid]

Decraene, B., Raszuk, R., Li, Z., and C. Li, "SRv6 vSID: Network Programming extension for variable length SIDs", draft-decraene-spring-srv6-vlsid-05 (work in progress), February 2021.

- [I-D.filsfils-spring-net-pgm-extension-srv6-usid]
Filsfils, C., Garvia, P. C., Cai, D., Voyer, D., Meilik, I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman, D., Liu, Y., and J. Guichard, "Network Programming extension: SRv6 uSID instruction", draft-filsfils-spring-net-pgm-extension-srv6-usid-10 (work in progress), March 2021.
- [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc]
Cheng, W., Filsfils, C., Li, Z., Cai, D., Voyer, D., Clad, F., Zadok, S., Guichard, J. N., and L. Aihua, "Compressed SRv6 Segment List Encoding in SRH", draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-02 (work in progress), November 2020.
- [I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-10 (work in progress), April 2021.
- [I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", draft-ietf-bess-srv6-services-07 (work in progress), April 2021.
- [I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-07 (work in progress), March 2021.
- [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-15 (work in progress), April 2021.
- [I-D.ietf-lsr-ip-flexalgo]
Britto, W., Hegde, S., Kaneriya, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", draft-ietf-lsr-ip-flexalgo-02 (work in progress), April 2021.

- [I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-14 (work in progress), April 2021.
- [I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-06 (work in progress), February 2021.
- [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-11 (work in progress), April 2021.
- [I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, 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-04 (work in progress), March 2021.
- [I-D.mirsky-6man-unified-id-sr]
Weiqiang, C., Mirsky, G., Shaofu, P., Aihua, L., and G. S. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-09 (work in progress), March 2021.
- [I-D.srcompdt-spring-compression-requirement]
Cheng, W., Xie, C., Bonica, R., Dukes, D., Li, C., Shaofu, P., and W. Henderickx, "Compressed SRv6 SID List Requirements", draft-srcompdt-spring-compression-requirement-06 (work in progress), March 2021.
- [I-D.ssangli-bess-bgp-vpn-srm6]
Sangli, S. and R. Bonica, "BGP based Virtual Private Network (VPN) Services over SRm6 enabled IPv6 networks", draft-ssangli-bess-bgp-vpn-srm6-02 (work in progress), September 2020.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

- [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>>.
- [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>>.

Appendix A. Encapsulation analysis

A.1. CRH note

CRH compression efficiency statistics are derived as follows:

If an SR path contains no transport segments and a VPN segment, the SR path is encoded in a single IPv6 header (40 bytes). The destination address in the IPv6 header is a classic SRv6 SID (e.g., END.DT4, END.DT6).

If the SR path contains T transport segments and a VPN segment, and T is greater than 0, the SR path can be encoded:

- o With an IPv6 Tunnel Payload Function (TPF) Option [I-D.bonica-6man-vpn-dest-opt]
- o Without a TPF Option

If the SR path is encoded with a TPF Option, the packet includes a single IPv6 Header (40 bytes), a CRH (variable length), and a Destination Options header (8 bytes). The destination address in the IPv6 header represents the IPv6 address of an interface on the first transport segment endpoint. The CRH must be large enough to contain the subsequent T segments.

If the SR path is encoded without a TPF Option, the packet includes a single IPv6 Header (40 bytes) plus a CRH (variable length). The destination address in the IPv6 header represents the IPv6 address of an interface on the first transport segment endpoint. The CRH must be large enough to contain T+1 segments. In the CRH, SID[1] maps to the IPv6 address of the PE router. SID[0] maps to a classic SRv6 SID (e.g., END.DT4) that is instantiated on the PE router.

In some deployment scenarios, each encoding strategy yields better compression.

A.2. Analysis results

The detailed encapsulation and encapsulation savings per proposal with one VPN segment and "T" transport segments:

T	CSID	CRH	CRH+TPF	VSID	UIDSR
0	40	40	40	40	40
1	40	48	56	56	64
2	40	56	56	56	64
3	40	56	64	56	64
4	64	56	64	64	64
5	64	56	64	64	64
6	64	64	64	64	64
7	64	64	72	64	64
8	64	64	72	72	64
9	80	64	72	72	80
10	80	72	72	72	80
11	80	72	80	72	80
12	80	72	80	80	80
13	80	72	80	80	80
14	96	80	80	80	80
15	96	80	88	80	80

Table 12: Encapsulation (E) octets, 16bit SIDS, 48B.0-15T.V

T	CSID	CRH	CRH+TPF	VSID	UIDSR
0	0.0%	0.0%	0.0%	0.0%	0.0%
1	37.5%	25.0%	12.5%	12.5%	0.0%
2	50.0%	30.0%	30.0%	30.0%	20.0%
3	58.3%	41.7%	33.3%	41.7%	33.3%
4	42.9%	50.0%	42.9%	42.9%	42.9%
5	50.0%	56.3%	50.0%	50.0%	50.0%
6	55.6%	55.6%	55.6%	55.6%	55.6%
7	60.0%	60.0%	55.0%	60.0%	60.0%
8	63.6%	63.6%	59.1%	59.1%	63.6%
9	58.3%	66.7%	62.5%	62.5%	58.3%
10	61.5%	65.4%	65.4%	65.4%	61.5%
11	64.3%	67.9%	64.3%	67.9%	64.3%
12	66.7%	70.0%	66.7%	66.7%	66.7%
13	68.8%	71.9%	68.8%	68.8%	68.8%
14	64.7%	70.6%	70.6%	70.6%	70.6%
15	66.7%	72.2%	69.4%	72.2%	72.2%

Table 13: Encapsulation Savings (ES), 16bit SIDS, 48B.0-15T.V

T	CSID	CRH	CRH+TPF	VSID	UIDSR
0	40	40	40	40	40
1	64	56	56	56	64
2	64	56	64	56	64
3	64	64	64	64	64
4	64	64	72	64	64
5	80	72	72	72	80
6	80	72	80	72	80
7	80	80	80	80	80
8	80	80	88	80	80
9	96	88	88	88	96
10	96	88	96	88	96
11	96	96	96	96	96
12	96	96	104	96	96
13	112	104	104	104	112
14	112	104	112	104	112
15	112	112	112	112	112

Table 14: Encapsulation (E) octets, 32bit SIDS, 48B.0-15T.V

T	CSID	CRH	CRH+TPF	VSID	UIDSR
0	0.0%	0.0%	0.0%	0.0%	0.0%
1	0.0%	12.5%	12.5%	12.5%	0.0%
2	20.0%	30.0%	20.0%	30.0%	20.0%
3	33.3%	33.3%	33.3%	33.3%	33.3%
4	42.9%	42.9%	35.7%	42.9%	42.9%
5	37.5%	43.8%	43.8%	43.8%	37.5%
6	44.4%	50.0%	44.4%	50.0%	44.4%
7	50.0%	50.0%	50.0%	50.0%	50.0%
8	54.5%	54.5%	50.0%	54.5%	54.5%
9	50.0%	54.2%	54.2%	54.2%	50.0%
10	53.8%	57.7%	53.8%	57.7%	53.8%
11	57.1%	57.1%	57.1%	57.1%	57.1%
12	60.0%	60.0%	56.7%	60.0%	60.0%
13	56.3%	59.4%	59.4%	59.4%	56.3%
14	58.8%	61.8%	58.8%	61.8%	58.8%
15	61.1%	61.1%	61.1%	61.1%	61.1%

Table 15: Encapsulation Savings (ES), 32bit SIDS, 48B.0-15T.V

Authors' Addresses

Ron Bonica
Juniper

Email: rbonica@juniper.net

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Darren Dukes (editor)
Cisco Systems

Email: ddukes@cisco.com

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

SPRING
Internet-Draft
Intended status: Informational
Expires: January 10, 2022

W. Cheng
China Mobile
C. Xie
China Telecom
R. Bonica
Juniper
D. Dukes
Cisco Systems
C. Li
Huawei
P. Shaofu
ZTE
W. Henderickx
Nokia
July 09, 2021

Compressed SRv6 SID List Requirements
draft-srcompdt-spring-compression-requirement-07

Abstract

This document specifies requirements for solutions to compress SRv6 SID lists.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 10, 2022.

Copyright Notice

Copyright (c) 2021 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	3
2. Conventions used in this document	4
2.1. Requirements Language	4
2.2. Terminology	4
3. SRv6 SID List Compression Requirements	4
3.1. Dataplane Efficiency and Performance Requirements	4
3.1.1. Encapsulation Header Size	5
3.1.2. Forwarding Efficiency	5
3.1.3. State Efficiency	6
4. SRv6 Specific Requirements	6
4.1. SRv6 Based	6
4.2. Functional Requirements	7
4.2.1. SRv6 Functionality	7
4.2.2. Heterogeneous SID lists	8
4.2.3. SID list length	8
4.2.4. SID summarization	8
4.3. Operational Requirements	9
4.3.1. Lossless Compression	9
4.3.2. Preservation of non-routing information	9
4.3.3. Address Planning	10
4.4. Scalability Requirements	10
4.4.1. Adjacency segment scale	10
4.4.2. Prefix segment scale	11
4.4.3. Service Scale	11
4.4.4. Compression Levels	11
5. Protocol Design Requirements	11
5.1. SRv6 Base Coexistence	11
6. Security Requirements	12
6.1. Security Mechanisms	12
6.2. SR Domain Protection	12
7. IANA Considerations	12
8. Security Considerations	13
9. Contributors	13
10. Normative References	13
Appendix A. Proposed Requirements	14
A.1. IPv6 Based	14

A.2. Point to Multipoint	15
A.3. Parsability	15
Authors' Addresses	15

1. Introduction

The SPRING working group defined SRv6, with [RFC8402] describing how the Segment Routing (SR) architecture is instantiated on two data-planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). SRv6 uses a routing header called the SR Header (SRH) [RFC8754] and defines SRv6 SID behaviors and a registry for identifying them in [RFC8986]. SRv6 is a proposed standard and is deployed today.

The SPRING working group has observed that some use cases, such as strict path TE, may require long SRv6 SID lists. There are several proposed methods to reduce the resulting SRv6 encapsulation size by compressing the SID list.

The SPRING working group formed a design team to define requirements for, and analyze proposals to, compress SRv6 SID lists.

It is a goal of the design team to identify solutions to SRv6 SID list compression that are based on the SRv6 standards. As such, this document provides requirements for SRv6 SID list compression solutions that utilize the existing SRv6 data plane and control plane.

It is also a goal of the design team to consider proposals that are not based on the SRv6 data plane and control plane. As such, this document includes requirements to evaluate whether a compression proposal provides all the functionality of SRv6 (section "SRv6 Functionality") in addition to satisfying compression specific requirements.

For each requirement, a description, rationale and metrics are described.

The design team will produce a separate document to analyze the proposals.

This document is a draft; additional requirements are under review, additional requirements will be added, and current requirements may change. Appendix A contains a subset of requirements without unanimous consensus. Additional requirements without unanimous consensus are not in the appendix.

2. Conventions used in this document

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

SR: Segment Routing

SRH: Segment Routing Header

MPLS: Multiprotocol Label Switching

SR-MPLS: Segment Routing over MPLS data plane

SID: Segment Identifier

SRv6: Segment Routing over IPv6

SRv6 SID List: A list of SRv6 SIDs

Compression proposal: A proposal to compress SRv6 SID lists

SRv6 base: SRv6 as defined in [RFC8402], [RFC8754], [RFC8986]

SID numbering space: may be implemented as

- o a single IGP instance
- o a single IGP level or area
- o two or more autonomous systems that coordinate SID numbering space
- o two or more IGP instances that coordinate SID numbering space

SRv6 Encapsulation Header: The IPv6 header, and any extension headers preceding a payload, used to implement a SRv6 base or compression proposal.

3. SRv6 SID List Compression Requirements

3.1. Dataplane Efficiency and Performance Requirements

3.1.1. Encapsulation Header Size

Description: The compression proposal MUST reduce the size of the SRv6 encapsulation header.

Rationale: A smaller SRv6 encapsulation results in better MTU efficiency.

Metric: Compression is the ratio of the IPv6 encapsulation size of SRv6 as defined in [RFC8402], [RFC8754], [RFC8986] vs the IPv6 encapsulation size of a given proposal. The encapsulation savings of a compression proposal vs the SRv6 base is a useful measurement to compare proposals.

The encapsulation metric (E) records the number of bytes required for a proposal to encapsulate a packet given a specific segment list.

o $E(\text{proposal}, \text{segment list})$.

The encapsulation savings (ES) records the encapsulation savings for a proposal to encapsulate a packet given a specific segment list.

o $ES(\text{proposal}, \text{segment list}) = 1 - E(\text{proposal}, \text{segment list})/E(\text{SRv6 base}, \text{segment list})$.

3.1.2. Forwarding Efficiency

Description: The compression proposal SHOULD minimize the number of required hardware resources accessed to process a segment.

Rationale: Efficiency in bits on the wire and processing efficiency are both important. Optimizing one at the expense of the other may lead to significant performance impact.

Metric: The data plane efficiency metric (D) records the data plane forwarding efficiency of the proposed solution. Two metrics are used and recorded at each segment endpoint:

- o $D.PRS(\text{segment list})$: number of headers parsed during processing of the segment list, starting from and including the IPv6 header.
- o $D.LKU(\text{segment list})$: number of FIB lookups during processing of the segment list. The type of lookup is also recorded as longest prefix match (LPM) or exact match (EM)

3.1.3. State Efficiency

Description: The compression proposal SHOULD minimize the amount of additional forwarding state stored at a node.

Rationale: Additional state increases the complexity of the control plane and data plane. It can also result in an increase in memory usage.

Metric: The state efficiency metric (S) records the amount of additional forwarding state required by the proposed solution.

- o S(node parameters): the number of additional forwarding states that need to be stored at a node, given a set of node parameters consisting of the number of nodes in the network, number of local interfaces, number of adjacencies. The forwarding state is counted as entries required in a Forwarding Information Base (FIB) at a node.

4. SRv6 Specific Requirements

4.1. SRv6 Based

Description: A solution to compress SRv6 SID Lists SHOULD be based on the SRv6 architecture, control plane and data plane. The compression solution MAY be based on a different data plane and control plane, provided that it derives sufficient benefit.

Rationale: A compression proposal built on existing IETF standards is preferable to creating new standards with equivalent functionality and performance.

Metric: The utilization metric (U) records whether a proposal utilizes the SRv6 specifications.

Utilization is recorded in a table, with a column per proposal and rows consisting of the following metrics:

- o U.RFC8402: utilizes [RFC8402].
- o U.RFC8754: utilizes [RFC8754].
- o U.PGM: utilizes [RFC8986].
- o U.IGP: utilizes [I-D.ietf-lsr-isis-srv6-extensions].
- o U.BGP: utilizes [I-D.ietf-bess-srv6-services].
- o U.POL: utilizes [I-D.ietf-spring-segment-routing-policy].
- o U.BLS: utilizes [I-D.ietf-idr-bgppls-srv6-ext].
- o U.SVC: utilizes [I-D.ietf-spring-sr-service-programming].
- o U.OAM: utilizes [I-D.ietf-6man-spring-srv6-oam].
- o U.ALG: utilizes [I-D.ietf-lsr-flex-algo].

Each cell contains "yes" for utilizes, or "no" for does not utilize.

4.2. Functional Requirements

4.2.1. SRv6 Functionality

Description: A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement ensures no SRv6 functionality is lost. It is particularly important to understand how a proposal, as evaluated in section "SRv6 Based", provides this functionality.

Rationale: Operators require SRv6 functionality. Evaluating the extent to which a proposal supports SRv6 functionality is important for operators and implementors to understand the impact on network operations.

Metric: The Functionality metric (F) records whether a proposal supports SRv6 functionality and how the functionality is provided.

Functionality is recorded in a table with columns for each proposal and rows consisting of the following metrics:

- o F.SID: Supports SRv6 SID functionality as described in [RFC8402]
- o F.SCOPE: Supports globally and locally scoped SID functionality as described in [RFC8402]
- o F.PFX: Supports prefix SID functionality as described in [RFC8402] and [RFC8986]
- o F.ADJ: Supports adjacency SID functionality as described in [RFC8402] and [RFC8986]
- o F.BIND: Supports binding SID functionality as described in [RFC8402] and [RFC8986]
- o F.PEER: Supports BGP peering SID functionality as described in [RFC8402] and [RFC8986]
- o F.SVC: Supports L3 and L2 VPN service SID functionality as described in [RFC8986]
- o F.ALG: Supports flexible algorithms functionality as described in [I-D.ietf-lsr-flex-algo]
- o F.TILFA: Supports TI-LFA functionality as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- o F.SEC: Supports securing an SR domain with ingress filtering as functionally defined in [RFC8754]
- o F.IGP: Supports distributing topological SIDs and behaviors via ISIS as functionally described in [I-D.ietf-lsr-isis-srv6-extensions]
- o F.BGP: Supports BGP VPNs as functionally described in [I-D.ietf-bess-srv6-services]

- o F.POL: Supports SR policies and steering traffic over those policies as functionally described in [I-D.ietf-spring-segment-routing-policy]
- o F.BLS: Supports Link State distribution via BGP as functionally described in [I-D.ietf-idr-bgpls-srv6-ext]
- o F.SFC: Supports stateless service programming as functionally described in [I-D.ietf-spring-sr-service-programming]
- o F.PING: Supports pinging a SID to verify the SID is implemented as functionally described in [I-D.ietf-6man-spring-srv6-oam]

Each cell contains the specification name documenting the functionality.

4.2.2. Heterogeneous SID lists

Description: The compression proposal SHOULD support a combination of compressed and non-compressed segments in a single path. As an example, a solution may satisfy this requirement without being SRv6 based by using a binding SID to impose an additional SRv6 header (IPv6 header plus optional SRH) with non-compressed SID.

Rationale: Support of SID lists with compressed and non-compressed SIDs reduces encapsulation size when not all SRv6 nodes deploy the compression proposal or 128-bit SIDs are required.

Metric: A compliant compression proposal supports both:

- o classic 128-bit SRv6 SIDs in the IPv6 Destination Address field
- o segment lists (i.e., paths) with both compressed and 128-bit SRv6 SIDs.

4.2.3. SID list length

Description: The compression proposal MUST be able to represent SR paths that contain up to 16 segments.

Rationale: Strict TE paths require SID list lengths proportional to the diameter of the SR domain.

Metric: The compression proposal must be able to steer a packet through an SR path that contains up to sixteen segments.

4.2.4. SID summarization

Description: The solution MUST be compatible with segment summarization.

Rationale: Summarization of segments is a key benefit of SRv6 vs SR MPLS. In interdomain deployments, any node can reach any other node via a single prefix segment. Without summarization, border router SIDs must be leaked, and an additional global prefix segment is required for each domain border to be traversed.

Metric: A solution supports summarization when segments can be summarized for advertisement into other IGP domains or levels.

4.3. Operational Requirements

4.3.1. Lossless Compression

Description: A path traversed using a compressed SID list MUST always be the same as the path traversed using the uncompressed SID list if no compression was applied.

Rationale: In SRv6, we can represent a path to meet certain objectives. A compression proposal needs to support the objectives with the same path.

Metric: Information present in the pre-compression segment list MUST also be present in the post-compression SID list.

4.3.2. Preservation of non-routing information

Description: The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node

Rationale: SRv6 ingress nodes encode non-routing information in the IPv6 header chain. This information can be encoded in the following fields:

- o Hop Count
- o DSCP bits
- o ECN bits
- o Flow label
- o HBH Options Extension header
- o Fragment Extension header
- o Authentication Extension header
- o Encrypted Security Payload Extension header
- o Destination Options Extension header

Some of these fields are mutable (e.g., Hop Count) while others are immutable (e.g., Fragment Extension Header).

Some of these fields contain information that is required by every node along a packet's delivery path (e.g., Hop Count). Others contain information that is required only by the packet's ultimate destination (e.g., Fragment Extension Header).

Therefore, the compression mechanism MUST NOT prevent this information from being delivered, in an IPv6 header chain, to any node that needs it.

Metric: The SR source node encapsulates its payload (e.g., Ethernet, IP, TCP) in an IPv6 header. The SRv6 header contains both routing and non-routing information. The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node.

4.3.3. Address Planning

Description: Network operators require addressing plan flexibility, The compression mechanism MUST support flexible IPv6 address planning, it MUST support deployment by using GUA from different address blocks.

Rationale: The address planning of the network may vary based on the addressing scheme of the operator, so the solution MUST support a flexible addressing scheme. Operators need to deploy the solution based on their own address planning.

Metric: The compression proposal supports locators drawn from different prefixes with the solutions analysis indicating efficiency.

4.4. Scalability Requirements

4.4.1. Adjacency segment scale

Description: The compression proposal MUST be capable of representing 65000 adjacency segments per node

Rationale: Typically, network operators deploy networks with tens or hundreds of adjacency segments per node, but some network operators may deploy networks that use more adjacency segments per node.

Metric: A proposal that allows 65000 adjacency segments per node satisfies this requirement.

4.4.2. Prefix segment scale

Description: The compression proposal MUST be capable of representing 1 million prefix segments per SID numbering space.

Rationale: Typically, network operators deploy networks with thousands of prefix segments per SID numbering space, but some network operators may deploy networks that use more prefix segments per SID numbering space.

Metric: A proposal that allows 1 million prefix segments per SID numbering space satisfies this requirement.

4.4.3. Service Scale

Description: The compression proposal MUST be capable of representing 1 million services per node.

Rationale: Typically, network operators deploy networks with tens to hundreds of thousands of services per node, but some network operators may deploy networks that use more services per node.

Metric: A proposal that allows 1 million services per node satisfies this requirement.

4.4.4. Compression Levels

Description: The compression proposal SHOULD be able to support multiple levels of compression.

Rationale: The compression proposal will be deployed in networks of varying size with SID numbering spaces of varying size. Network and service scale can directly impact SID length and the ability of a proposal to compress the SID list.

Metric: A compression proposal that supports relatively better compression for smaller SID numbering spaces and service scale satisfies this requirement.

5. Protocol Design Requirements

5.1. SRv6 Base Coexistence

Description: The compression proposal MUST support simultaneous deployment with SRv6 networks.

Rationale: SRv6 is deployed today. A compression proposal that interoperates well with SRv6, as deployed, will reduce the overhead

and simplify operations. For Network operators who would migrate to compressed SRv6 SID lists, the migration is expected to gradually occur over a period of time as they upgrade networks, domains, device families and software instances.

Metric: A compliant compression proposal provides the following

- o Supports simultaneous deployment at a node with current SRv6 SIDs.
- o Supports simultaneous deployment at a node with current SRv6 control plane.
- o Supports simultaneous operation of current SRv6 paths with compressed paths.
- o Supports the behaviors in [RFC8986].
- o Does not require removal of existing IPv6 address planning.

6. Security Requirements

6.1. Security Mechanisms

Description: The compression solution SHOULD be able to address security issues that it introduces, using existing security mechanisms.

Rationale: It is important to identify security issues and how to address them in any specification.

Metric: A compression solution that does not introduce unresolved security issues meets this requirement.

6.2. SR Domain Protection

Description: A compression solution must not require nodes outside the SR domain to know SID values within the SR domain, and it must provide the ability to block nodes outside an SR domain from accessing SIDs.

Rationale: The unauthorized use of SIDs within the SR domain by nodes outside the domain can disrupt an operators' network.

Metric: A compliant solution describes how access to SIDs within the SR domain is denied to nodes outside the SR domain.

7. IANA Considerations

This document has no requests to IANA.

8. Security Considerations

TBD

9. Contributors

The following individuals contributed to this draft

Sanders Steffann, SJM Steffann Consultancy, sander@steffann.nl

10. Normative References

[I-D.ietf-6man-spring-srv6-oam]

Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-10 (work in progress), April 2021.

[I-D.ietf-bess-srv6-services]

Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", draft-ietf-bess-srv6-services-07 (work in progress), April 2021.

[I-D.ietf-idr-bgppls-srv6-ext]

Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgppls-srv6-ext-07 (work in progress), March 2021.

[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-15 (work in progress), April 2021.

[I-D.ietf-lsr-isis-srv6-extensions]

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-14 (work in progress), April 2021.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-06 (work in progress), February 2021.

- [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-11 (work in progress), April 2021.
- [I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, 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-04 (work in progress), March 2021.
- [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>>.
- [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>>.

Appendix A. Proposed Requirements

This appendix contains requirements that the design team discussed but could not be agreed upon.

A.1. IPv6 Based

Description: The compression mechanism requires every node along the packet's delivery path to be IPv6-capable. It MUST not require any

node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

Rational: According to RFC 8402, SRv6 is an instantiation of the SR Architecture over the IPv6 data plane.

Metric: A compliant solution requires every node along the packet's delivery path to be IPv6-capable. It does not require any node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

A.2. Point to Multipoint

Description: The compression mechanism SHOULD support point-to-multipoint SR paths.

Rationale: Many VPN services require point-to-multipoint SR paths.

Metric: A compliant proposal can encode a multicast address in the ultimate segment of the segment list.

A.3. Parsability

Description: The compression mechanism MUST be parsable. That is, the node that consumes the compressed SID list must be able to decode the active and next segment. Parsing information MAY be conveyed in either the forwarding or control plane.

Rationale: Failure to parse the compressed SID list leads to undesired behaviors.

Metric: In the nominal case the producer and consumer of the SID list agree on the active segment and next segment. In foreseeable failure modes it is possible to determine why the producer and consumer don't agree.

Authors' Addresses

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

Ron Bonica
Juniper

Email: rbonica@juniper.net

Darren Dukes
Cisco Systems

Email: ddukes@cisco.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com