RTGWG
Internet-Draft
Intended status: Standards Track
Expires: 20 December 2025

F. Yang
China Mobile
C. Lin
Y. Qiu
New H3C Technologies
X. Zhang
China Mobile

18 June 2025

Reliability Framework for SRv6 Service Function Chaining
draft-yang-rtgwg-srv6-sfc-reliability-framework-04

Abstract

Segment routing (SR) [RFC8402] is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node. SR can be instantiated on the MPLS data plane (MPLS-SR) and the IPv6 data plane (SRv6).

On the MPLS-SR data plane, a segment is encoded as an MPLS label, and an ordered list of segments is encoded as a stack of labels. On the SRv6 data plane, a segment is encoded as an IPv6 address (SRv6 SID) [RFC8986], and an ordered list of segments is encoded as an ordered list of SRv6 SIDs in the SR header (SRH) [RFC8754]. The ingress node steers packets into a specific path according to the ordered list of segments (SR Policy) as defined in [RFC9256]. Service Function Chaining (SFC) defines an ordered set of service functions and subsequent "steering" of traffic through them. The architecture of SFC is defined in [RFC7665].

This document describes the common failure scenarios and protection mechanisms of service function chains in SR networks. Then implementation recommendations for protection of service function chains are proposed.

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/.

Table of Contents

1. Introduction

Segment routing (SR) [RFC8402] is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node. SR can be instantiated on the MPLS data plane (MPLS-SR) and the IPv6 data plane (SRv6).

On the MPLS-SR data plane, a segment is encoded as an MPLS label, and an ordered list of segments is encoded as a stack of labels. On the SRv6 data plane, a segment is encoded as an IPv6 address (SRv6 SID) [RFC8986], and an ordered list of segments is encoded as an ordered list of SRv6 SIDs in the SR header (SRH) [RFC8754]. The ingress node steers packets into a specific path according to the ordered list of segments (SR Policy) as defined in [RFC9256].

Service Function Chaining (SFC) defines an ordered set of service functions and subsequent "steering" of traffic through them. The architecture of SFC is defined in [RFC7665].

This document describes the common failure scenarios and protection mechanisms of service function chains in SR networks. Then implementation recommendations for protection of service function chains are proposed.

1.1. Requirements Language

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

1.2. Terminology

The terms in this document are defined in [RFC7665], [RFC8459], [RFC8986] and [I-D.ietf-spring-sr-service-programming].

The following lists widely used terms in this document.

SR: Segment Routing

SRv6: Segment Routing over IPv6

CF: Classifier

SF: Service Function

SFF: Service Function Forwarder

SFC: Service Function Chaining

2. Background Requirements

In SRv6 networks, programming of forwarding paths can be achieved through the combination of SIDs. If the programming paths sequentially pass through the specified SF, SFC is achieved, which is the main idea of stateless SRv6 SFC solution.

The architecture of the Stateless SRv6 SFC solution is shown in Figure 1, which includes the following key components:

SRv6-aware SF: SF nodes that support SRv6 can be directly connected to SFF.

SRv6-unaware SF: SF nodes that do not support SRv6 need to deploy an SRv6 proxy between SFF and SRv6-unaware SF to complete the processing of SRv6 packets.

SRv6 Proxy: The proxy for SRv6, which forwards packets from the SRv6 network to the SRv6 unaware SF and returns them from the SRv6-unaware SF to the SRv6 network.

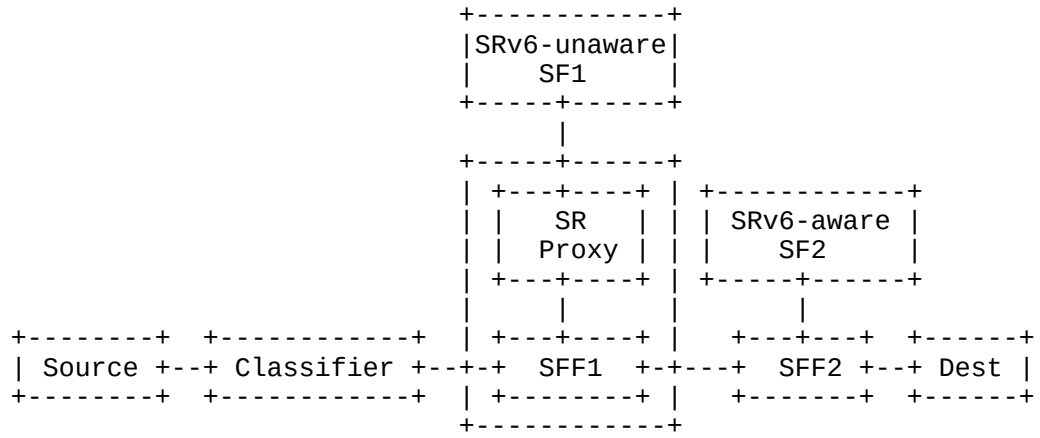

Figure 1

The SRv6 proxy caches the mapping relationship between SRH and the virtual interface connecting SF, which is used to recover SRv6 packets based on the virtual interface when packets return from SF.

Currently, during the deployment of SFC, we have encountered some reliability issues with SF. Taking the simple networking model in Figure 2 as an example, any of the following faults will cause the service message to be discarded.

Fault 1: SF fault.

Fault 2: SFF fault.

Fault 3: Link failure or unreachable routing between SFF and SF.

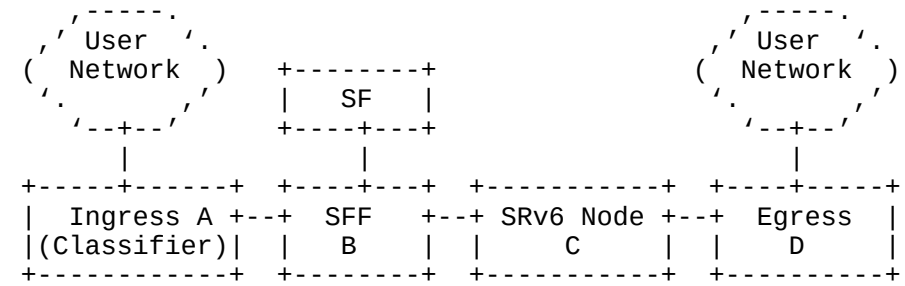

Figure 2

In order to improve the reliability of SFC, we propose the following three reliability protection mechanisms:

* SFF redundant backup protection method

* SF redundant backup protection method

* SF bypass forwarding method

It should be noted that OAM packets involve relatively complex state processing. Discarding or altering these packets can lead to compatibility issues for the entire connection. Therefore, it is advisable to avoid diverting OAM traffic into the SFC processing.

3. SFC Reliability Protection Mechanisms

3.1. SFF Redundant Backup Protection Method

Deploy primary and backup SFFs for SF. SF is dual-homed connected to both SFFs simultaneously. When the routing between the primary SFF and SF is unreachable or primary SFF fails, the service message is switched to the backup SFF and forwarded to SF via the backup SFF.

The following describes the working process of SFF redundant backup protection mechanism based on SF’s SRv6 capabilities and SRv6 Proxy behavior.

3.1.1. Static SR Proxy

As shown in Figure 3, SFF1 and SFF2 are two Service Function Forwarders that are backed up to each other. SF is connected to both SFF1 and SFF2.

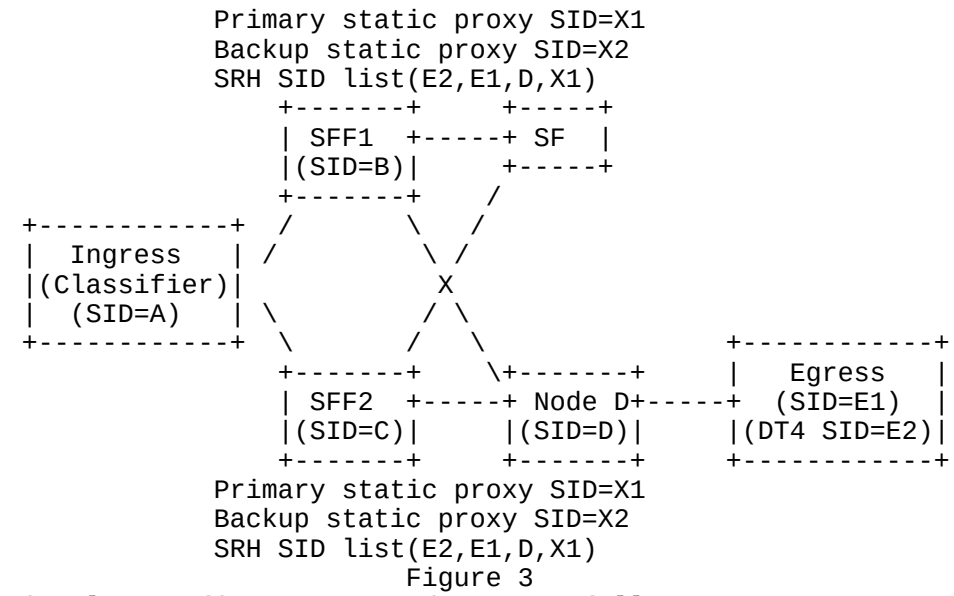

Figure 3

Firstly, configure SFF1 and SFF2 as follows:

* Configure the mapping relationship cache entries for the same SRH and interface connecting SF on two SFFs.

* Two SFFs use the same Locator segment to assign their respective static proxy SID.

* Two SFFs with the same configuration as the primary static proxy SID and backup static proxy SID. Both specify SFF1 as the primary and SFF2 as the backup.

Normally, the Ingress node encapsulates the SRH header and fills the SID list with <E1, E2, D, X1>, indicating that the message is forwarded along the path Ingress->SFF1->SF->SFF1->NodeD->Egress. After receiving the message, SFF1 matches it to the local primary static proxy SID based on the destination address X1 of the message. SFF1 performs the processing of End.AS behavior, removes the IPv6 header and SRH, and sends the original message to SF through the interface corresponding to the service SID.

When SFF1 detects that the route from SFF1 to SF is unreachable, it encapsulates a new SRv6 header to the message based on the configured backup SFF. There are two encapsulation options for the SRv6 header:

* Option 1: With SRH, forward the payload to the backup SFF through the SRv6 TE path.

- The SRH SID list contains the segment IDs of the nodes that need to be passed through for backup SFF.

- SID[0] is backup static proxy SID.

* Option 2: Forward the payload to the backup SFF through the SRv6 BE path.

- The IPv6 DA is backup static proxy SID.

SFF1 searches the routing table and forwards the updated message to SFF2.

After receiving the message, if the route from SFF2 to SF is reachable, SFF2 removes the outer SRv6 header, and forwards the payload to SF. If the route from SFF2 to SF is also unreachable, discard the message.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 4.

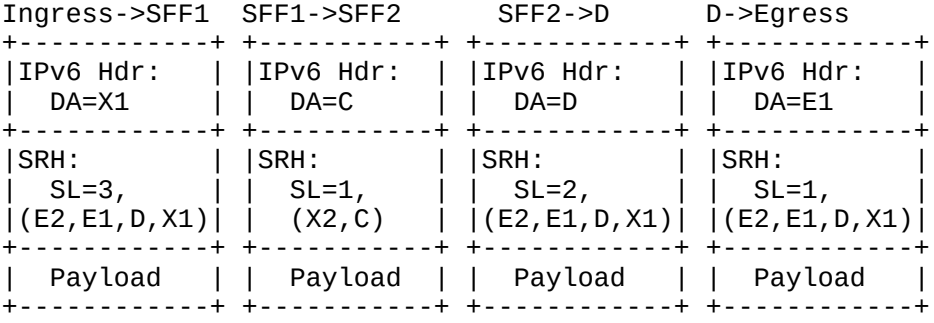

Figure 4

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 5.

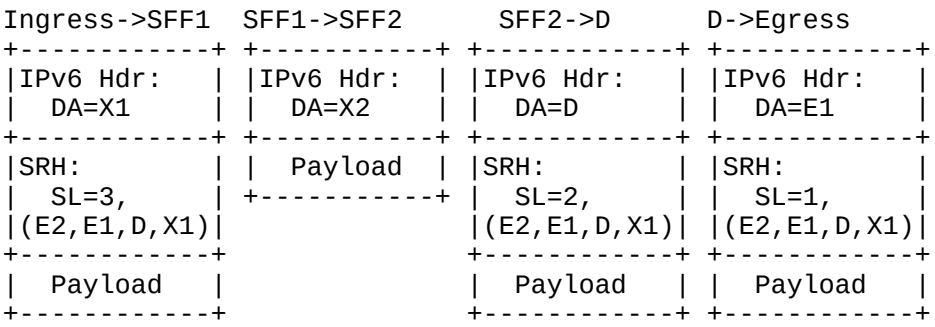

Figure 5

3.1.2. Dynamic SR Proxy

Compared to static SR Proxy, dynamic SR Proxy has the ability to learn mapping relationship cache entries. The mapping relationship cache between SRH and the interface connecting SF is dynamically generated based on the SRH of the messages received by SFF.

After enabling SFF redundant backup protection function, when the primary SFF senses that it is unreachable to SF, the SRH are not removed. Instead, SFF1 needs to carry the original SRH to the backup SFF.

As shown in Figure 6, SFF1 and SFF2 are two Service Function Forwarders that are backed up to each other, and SF is connected to both SFF1 and SFF2.

Firstly, configure SFF1 and SFF2 as follows:

* Two SFFs use the same Locator segment to assign their respective dynamic proxy SID.

* Two SFFs with the same configuration as the primary dynamic proxy SID and backup dynamic proxy SID. Both specify SFF1 as the primary and SFF2 as the backup.

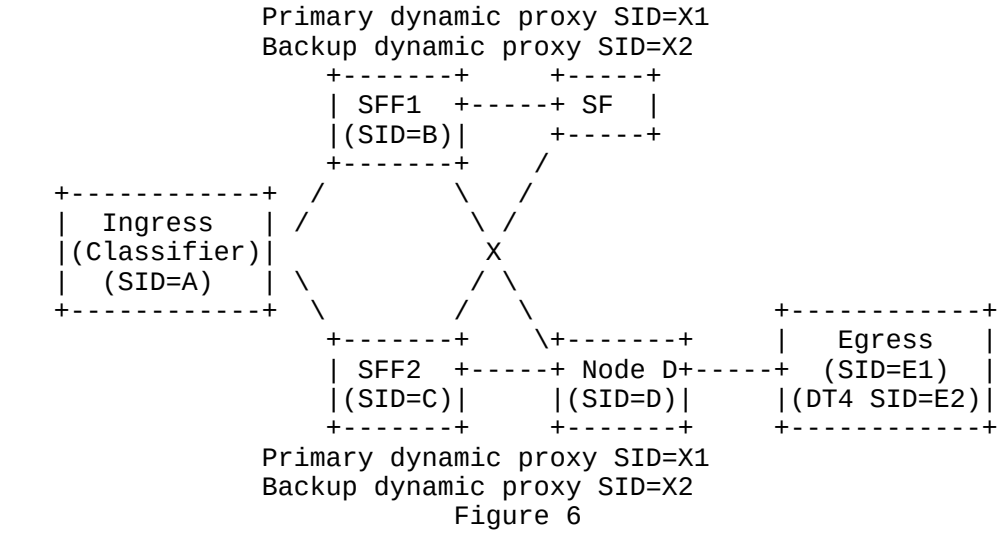

Figure 6

Normally, the Ingress node encapsulates the SRH header and fills the SID list with <E1, E2, D, X1>, indicating that the message is forwarded along the path Ingress->SFF1->SF->SFF1->NodeD->Egress. After receiving the message, SFF1 matches it to the local dynamic proxy SID based on the destination address X1 of the message.

SFF1 performs the processing of End.AD behavior, decapsulates the message, removes the IPv6 header and SRH, and sends the payload to SF through the interface corresponding to the service SID. At the same time, SFF1 records the mapping relationship cache between SRH and the interface connecting SF, in order to recover the SRH when the message returns from SF.

When SFF1 detects that the route from SFF1 to SF is unreachable, SFF1 does not remove the existing IPv6 header and SRH, and there are two process options for the SRv6 header:

* Option 1: SFF1 adds another SRv6 header to the message.

  - The SRH SID list contains the segment IDs of the nodes that need to be passed through for backup SFF.

  - SID[0] is backup dynamic proxy SID.

* Option 2: SFF1 reuses existing IPv6 headers. SFF1 replaces the IPv6 DA to backup dynamic proxy SID.

SFF1 searches the routing table and forwards the updated message to SFF2. After the message arrives at SFF2, it is matched to the local

backup dynamic proxy SID based on the destination address X2. SFF2 removes the outer IPv6 header(s), sends the payload to SF, and records the mapping relationship cache between SRH and the interface connecting SF.

If the route from SFF2 to SF is unreachable, it is considered that both the primary and backup SFFs are unable to process and the message is discarded.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 7.

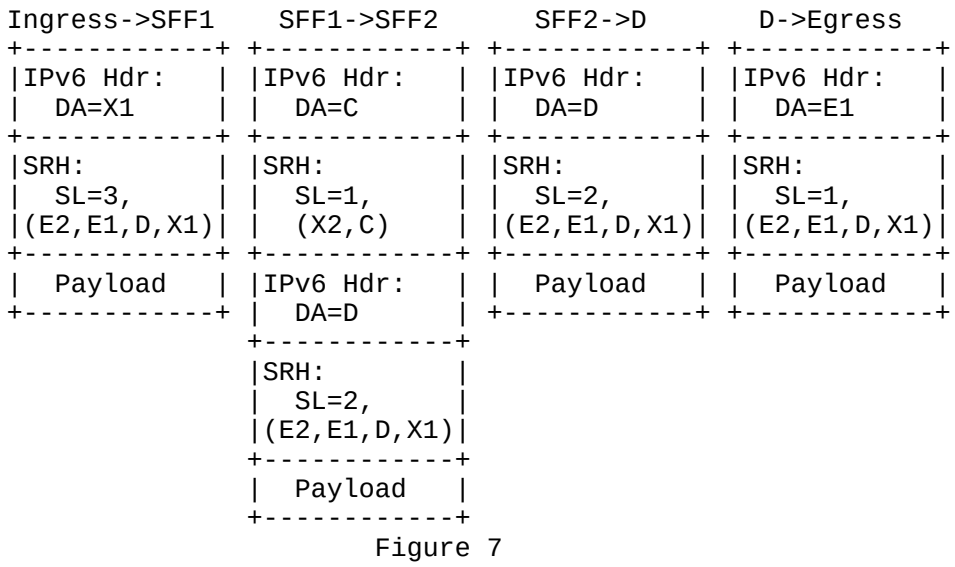

Figure 7

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 8.

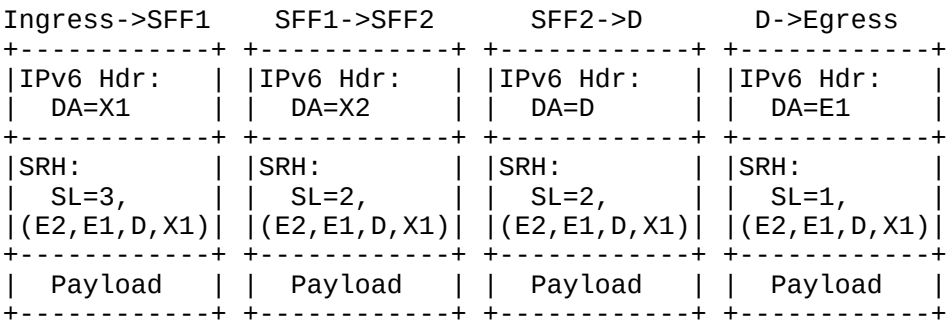

Figure 8

3.1.3. Masquerading SR Proxy

SFC masquerading proxy is suitable for scenarios where SF can recognize SRv6 packets but does not support processing SRH. Before forwarding the message to the SF, the masquerading SR proxy first updates SID[0] of Segment list to DA, and then directly forwards the SRv6 message to SF. After SF completes the service processing, it does not process the SRv6 SRH and forwards the message back to SFF. After receiving the message returned from SF, SFF updates DA to SID[SL] of Segment list and continues forwarding.

Because the SRH has always been carried in the message without modification, SFF does not need to re encapsulate the SRH after receiving the message returned from SF, and can directly use the SRH inside the message.

In the scenario of using masquerading SR proxy, after enabling SFF redundant backup protection function, when the primary SFF senses that the route with SF is unreachable, the primary SFF does not remove the IPv6 header and SRH. Instead, SFF1 needs to carry the SRH to the backup SFF.

Taking Figure 9 as an example, the detailed protection processing process is as follows.

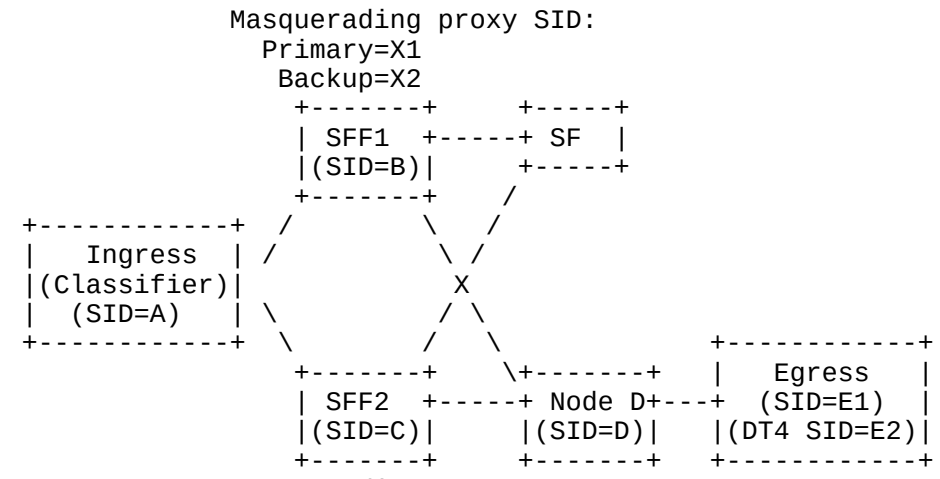

Figure 9

Configure SFF1 and SFF2 as follows:

* Two SFFs use the same Locator segment to assign their respective masquerading proxy SID.

* Two SFFs with the same configuration as the primary masquerading proxy SID and Backup masquerading proxy SID. Both specify SFF1 as the primary and SFF2 as the backup.

Normally, the message is forwarded along the path Ingress->SFF1->SF->SFF1->NodeD->Egress. After receiving the message, SFF1 matches it to the local masquerading proxy SID based on the DA of the message. SFF1 performs the processing of the End.AM behavior, changing DA to SID[0] of Segment list, and then sending the message to SF through the interface connecting to SF.

When SFF1 detects that the route from SFF1 to SF is unreachable, there are two options:

* Option 1: SFF-1 replaces the IPv6 DA with the backup masquerading proxy SID.

  After the message arrives at SFF2, SFF2 changes the DA to SID[0] of the Segment list, and then sending the message to SF through the interface connecting to SF.

* Option 2: SFF-1 adds another SRv6 header to the message. The SRH[0] is backup masquerading proxy SID.

  After the message arrives at SFF2, SFF2 first removes the outer SRv6 header and restores the inner SRv6 message. Then, use the SID[0] of the SRH Segment List as the destination address to send the message to SF.

If the route from SFF2 to SF is unreachable, it is considered that both the primary and backup SFFs are unable to process and the message is discarded.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 10.

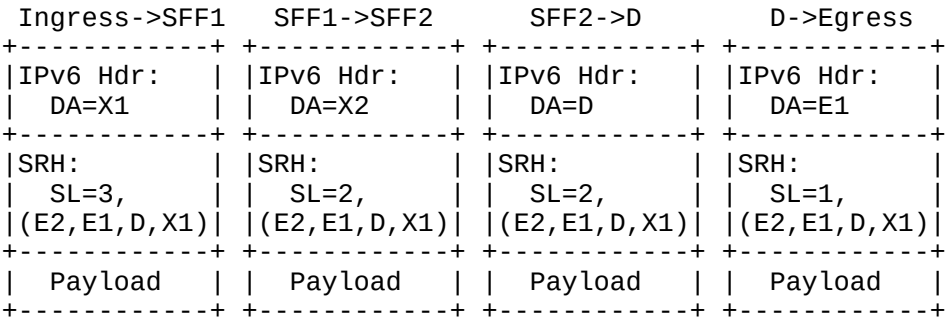

Figure 10

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 11.

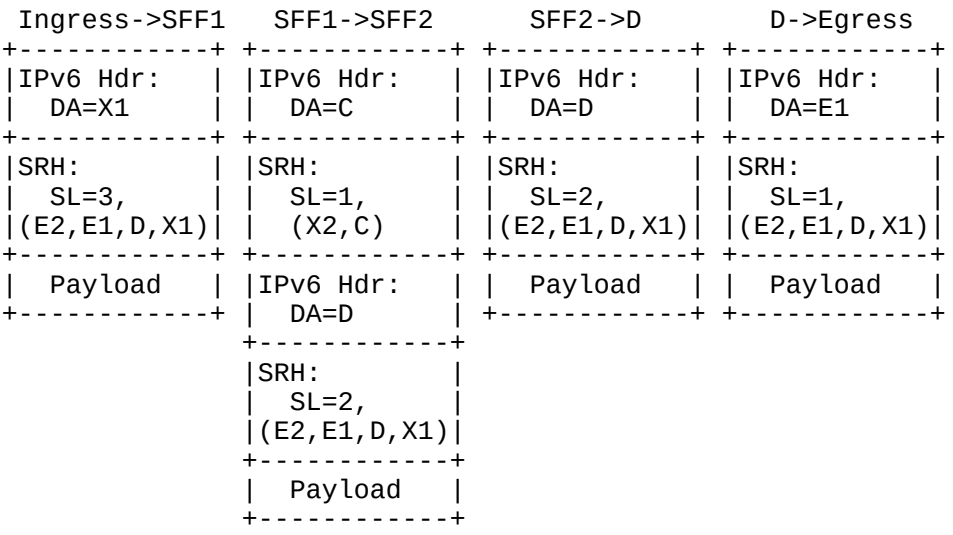

Figure 11

3.1.4. SRv6-aware SF

SF, which supports SRv6 functionality, serves as a SRv6 endpoint node and is arranged on the path of SRv6 Policy forwarding. Its reliability protection mechanism is identical to that of SRv6 endpoint nodes, and no special handling is required in the SFF redundant backup protection mechanism.

3.2. SF Redundant Backup Protection Method

There are active and standby SFs in the network, which are connected to the same SFF or different SFFs.

SF redundant backup protection method is the process of bypassing the faulty primary SF and continuing to send messages to backup SF for processing. The following describes the working processes of the SF redundant backup protection method based on SF’s SRv6 capability and SRv6 Proxy’s behavior.

3.2.1. Static SR Proxy

As shown in Figure 12, there is a backup service function node SF2 for primary service function node SF1 in the SFC network. SF1 is connected to SFF1 with a single-homed connection, and SF2 is connected to SFF2 with a single-homed connection.

In order to implement the SF redundant backup protection function, it is necessary to enable the SF backup function on SFF1 and configure the bypass proxy connected to the backup SF.

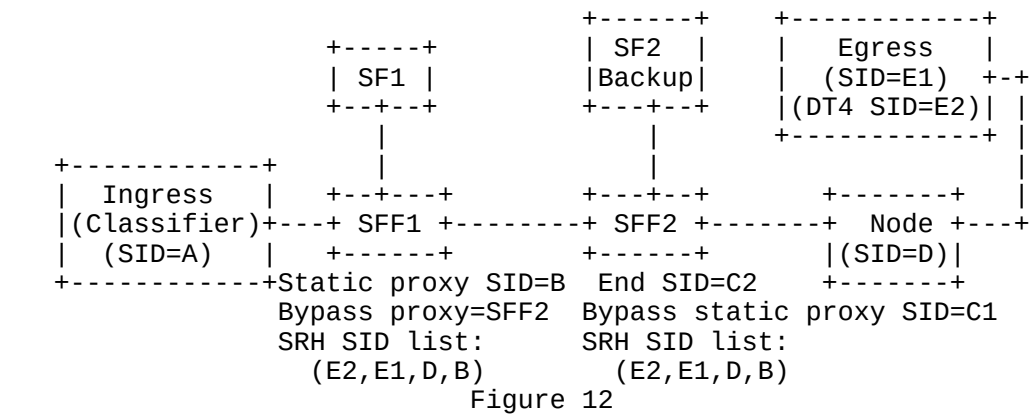

Figure 12

When SFF1 detects that the route from SFF1 to SF1 is unreachable, it encapsulates a new SRv6 header to the message. There are two encapsulation options for the SRv6 header:

* Option 1: With SRH, forward the payload to the bypass SFF through the SRv6 TE path.

  - The SRH SID list contains the segment IDs of the nodes that need to be passed through for the SFF connecting to the backup SF.

  - SRH[0] is the bypass static proxy SID.

* Option 2: Forward the payload to the bypass SFF through the SRv6 BE path.

  - The IPv6 DA is bypass static proxy SID.

SFF1 searches the routing table and forwards the updated message to SFF2. After receiving the message, SFF2 removes the IPv6 header, and sends the payload to SF2. If the route from SFF2 to SF2 is also unreachable, discard the message.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 13.

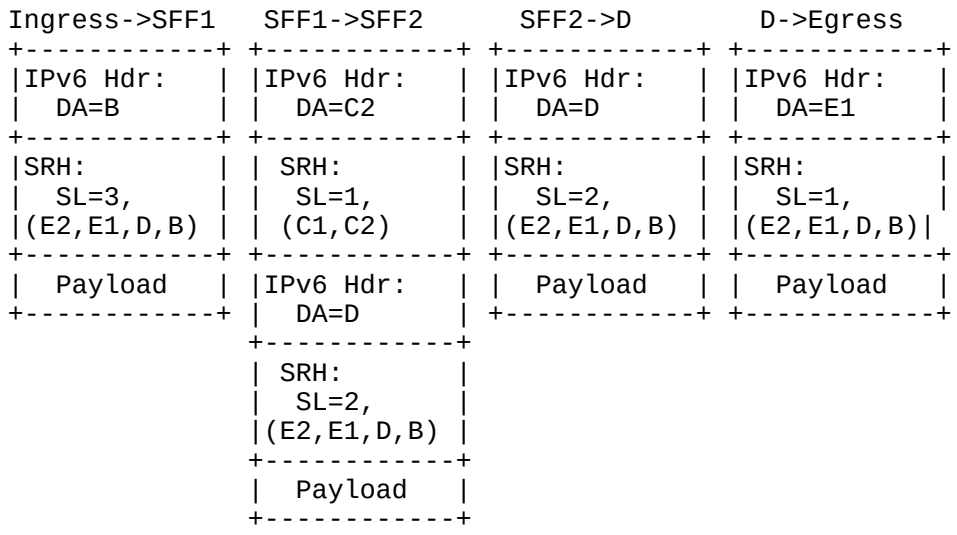

Figure 13

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 14.

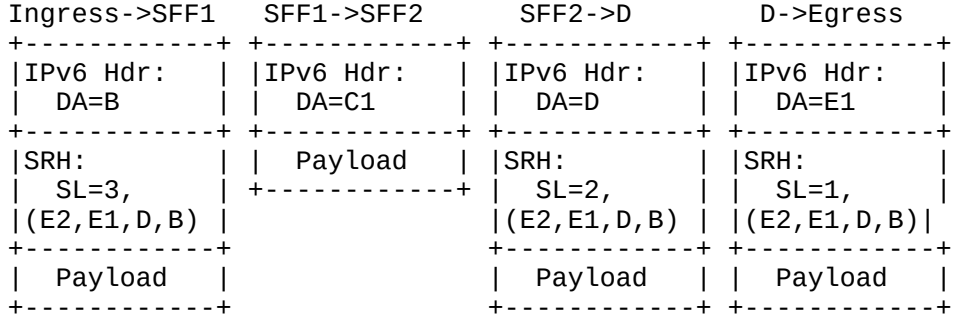

Figure 14

3.2.2. Dynamic SR Proxy

Because the dynamic SR proxy needs to dynamically generate a mapping relationship cache with the virtual interface connecting to SF based on the SRH of the message, after enabling SF redundant backup protection function, when the SFF senses that the route with primary SF is unreachable, it cannot remove IPv6 encapsulation. Instead, a new IPv6 header needs to be added outside the IPv6 header, and the original IPv6 packet is sent as a payload to the bypass SFF connecting to the backup SF.

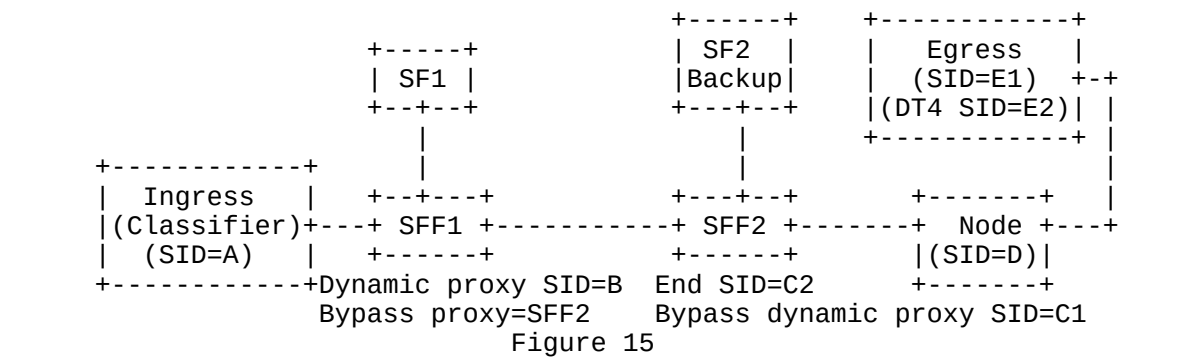

Figure 15

As shown in Figure 15, there is a backup service function node SF2 for SF1. SF1 is connected to SFF1 with a single-homed connection, and SF2 is connected to SFF2.

Enable the SF redundant backup protection function and specify SFF2 as the bypass protection node on SFF1. Configure the bypass dynamic proxy SID corresponding to the backup SF on SFF-2.

Normally, the message is forwarded along the path Ingress->SFF1->SF1->SFF1->NodeD->Egress. After receiving the message, SFF1 matches it to the local dynamic proxy SID based on the DA. SFF1 performs the processing of the dynamic proxy behavior, removes the original IPv6 header and SRH, and sends the payload to SF1 through the virtual interface connecting SF1. At the same time, SFF1 records the mapping relationship cache between SRH and the interface connecting SF1.

When SFF1 detects that the route from SFF1 to SF1 is unreachable, SFF1 does not remove the existing IPv6 header and SRH, and there are two process options:

* Option 1: SFF1 adds another SRv6 header to the message.

  - The SRH SID list contains the segment IDs of the nodes that need to be passed through for the SFF connecting to the backup SF.

  - SID[0] is bypass dynamic proxy SID.

* Option 2: SFF1 reuses existing IPv6 headers. SFF1 replaces the IPv6 DA to bypass dynamic proxy SID.

SFF1 looks up the routing table and forwards the encapsulated message to SFF2.

After the message arrives at SFF2, it is matched to the local bypass dynamic proxy SID based on the DA. SFF-2 removes the outer IPv6 header(s), sends the payload to SF2, and records the mapping relationship cache between SRH and the virtual interface connecting SF2.

If SFF2 to SF2 are also unreachable, discard the message.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 16.

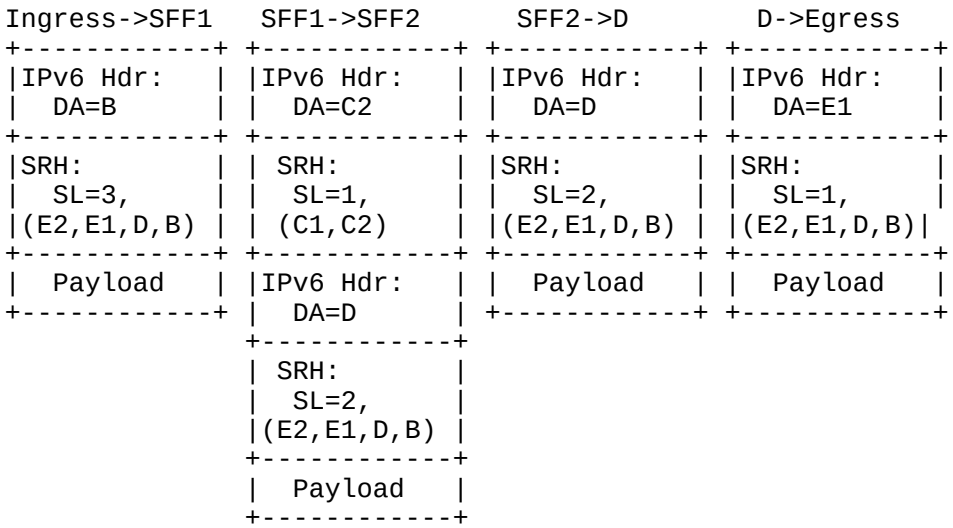

Figure 16

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 17.

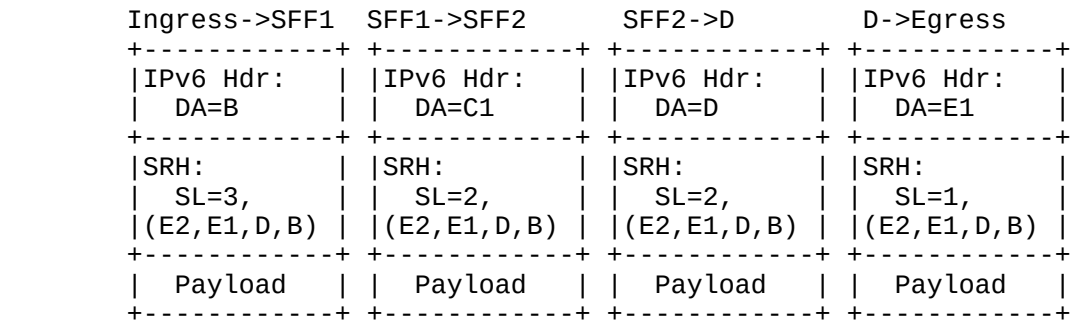

Figure 17

3.2.3. Masquerading SR Proxy

The SF redundant backup protection mechanism of the masquerading SR proxy is basically the same as the SF redundant backup protection method of the dynamic SR proxy, except that the SID of the bypass SFF needs to be with masquerading proxy behavior.

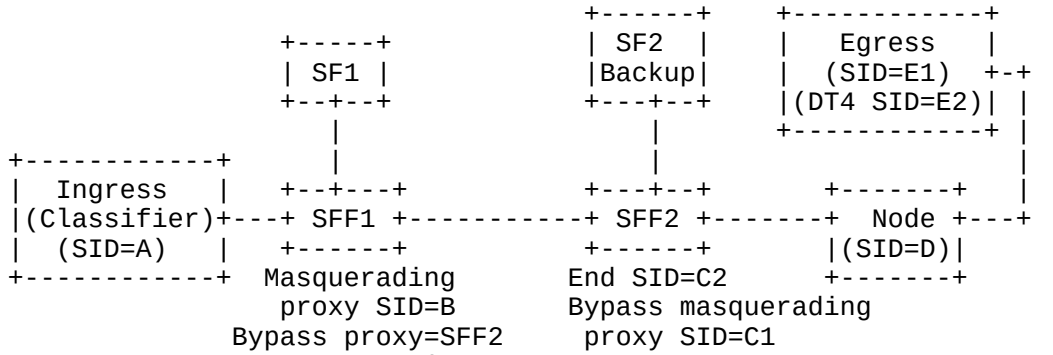

Figure 18

As shown in Figure 18, when SFF1 detects that it is unreachable from SFF1 to SF1, SFF1 does not remove the existing IPv6 header and SRH, and there are two process options:

* Option 1: SFF1 adds another SRv6 header to the message. The SID[0] is bypass masquerading proxy SID.

  After the message arrives at SFF2, SFF2 first removes the outer SRv6 header and restores the inner SRv6 message. Then, use the SID[0] of the SRH Segment List as the destination address to send the message to backup SF.

* Option 2: SFF-1 replaces the IPv6 DA with the bypass masquerading proxy SID.

After the message arrives at SFF2, SFF2 changes the DA to SID[0] of the Segment list, and then sending the message to backup SF through the interface connecting to backup SF.

If SFF2 to SF2 are also unreachable, discard the message.

For option 1, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 19.

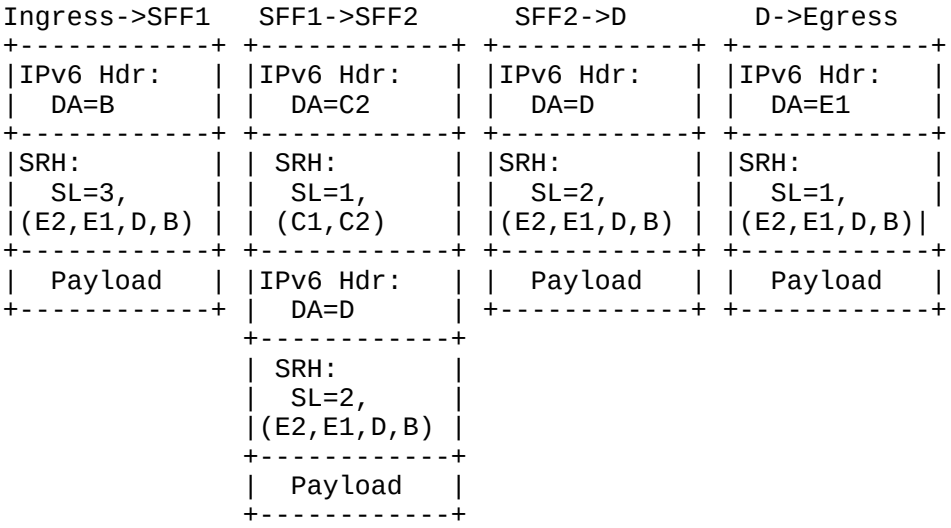

Figure 19

For option 2, the encapsulation of messages transmitted between SRv6 nodes from Ingress to Egress is shown in Figure 20.

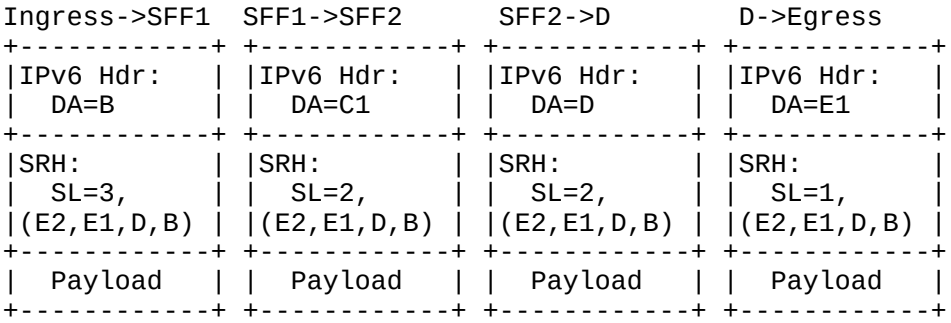

Figure 20

3.2.4. SRv6-aware SF

SF, which supports SRv6 functionality, serves as a SRv6 endpoint node and is arranged on the path of SRv6 Policy forwarding. Its

reliability protection mechanism is identical to that of SRv6 endpoint nodes, and there is no need to add special processing in the SF redundant backup protection mechanism.

3.3. SFF Bypass forwarding Method

The SFF bypass forwarding method is applied to scenarios where backup SFF and backup SF are not deployed for SR Proxy.

As shown in Figure 21, when SFF1 senses an SF fault or the route to SF is unreachable, it skips the service function node and directly forwards packets to downstream nodes.

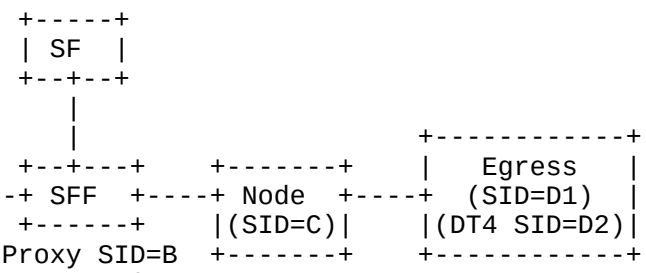

Figure 21

When SFF senses that SF is unreachable, it does not remove the IPv6 header, changes the destination address of the message to SID[SL], looks up the routing table, and forwards the message.

3.4. The Service Flow Affiliation Maintenance Method

In the current SFC scenario, SF can be deployed on multiple VMs/containers, so that a VM/container can switch to another one when it is overloaded or failed. There is no specific configuration for SF, so the user’s business will be distributed to one of the free VMs/containers randomly according to a set algorithm by the load balancer inside the SF.

For some specific users or services, the flow affiliation is to hope that the service flow can always be maintained in a certain VM/container to realize the optimal service, until the current VM/container is overloaded or failed before switching, rather than randomly distributed to different VM/container. Therefore, we need to configure the network device with relevant policies for the VM/container level and implement corresponding processing in the forwarding process to realize the maintenance of flow affiliation and session continuity. For example, we expect to use the VM/container in shared or exclusive mode, or prefer a certain type of VM/container such as GPU.

The service chain head node (SC) can determine a target SRv6 policy based on the service of the target user, and the SRv6 policy includes at least one logical device(VM or container) selection indication information corresponding to the target service. The information contains as follows: preferred VM or container and a selection strategy for this, an exclusive mode or a shared mode for the VM/container, and a selection strategy for the second best VM/container in case of failure of the preferred one.

According to the target SRv6 policy, an IPv6 message for the service to be transmitted is generated, including the SF SID with the VM/container selection indication information. Then SC sends a SRv6 policy message to the next hop node such as SFF, with the Function field and/or Argument field of the SF SID including said VM/container selection indication information, which can be recorded in the Function flavor parameter or the color attribute of the SRv6 Policy.

4. Changes in SR Proxy Behavior

In the SFF bypass forwarding method, when SR Proxy is processing local proxy SIDs, if it is unreachable to SF, the message cannot be discarded. Instead, SFF should modify the destination address of the message to SID[SL], search the routing table, and forward the message.

Among the reliability methods described in Section 3, there are also differences in the processing of SR Proxy SIDs on backup/bypass SFF compared to the normal End.AS, End.AD, and End.AM SID behaviors defined in [I-D.ietf-spring-sr-service-programming].

When the backup proxy or bypass proxy receives a message with two layers of SRv6 header encapsulation sent by the primary SFF, these two layers of SRv6 encapsulation need to be removed.

After receiving an IPv6 packet from the primary SFF, the backup and bypass proxies do not verify the consistency between DA and SID[SL].

There are three options for implementing the above functions:

4.1. Option 1: Add configuration on SR proxy.

The SFC reliability protection methods proposed in this document only need to be processed locally on the SFF node, and can be configured on the SR proxy to distinguish whether the End.AS, End.AD, and End.AM SIDs on the backup/bypass SFF perform the above special processing.

4.2. Option 2: Define new behavior for SR proxy SIDs

Define new behavior for SR proxy SIDs for SFC reliability protection, corresponding to the End.AS, End.AD, and End.AM SIDs in the primary SFF. When sending messages from the primary SFF to the backup/bypass SFF, use an SR proxy SID with backup or bypass behavior.

As described in Section 5.1.

4.3. Option 3: Define new flavors for SR proxy SIDs.

Define new flavors for End.AS, End.AD, and End.AM SIDs to indicate that additional SFF reliability processing is required when DA is these SIDs. When sending messages from the primary SFF to the backup/bypass SFF, use an SR proxy SID with a backup or bypass flavor.

As described in Section 5.2.

5. IANA Considerations

5.1. SRv6 Endpoint behavior

This document requests the IANA to allocate, within the "SRv6 Endpoint Behaviors" sub-registry belonging to the top-level "Segment-routing with IPv6 dataplane (SRv6) Parameters" registry, the following allocations:

Value	Description		Reference
TBA1-1	End.ASBK	- SFF Backup static proxy	[This.ID]
TBA1-2	End.ADBK	- SFF Backup dynamic proxy	[This.ID]
TBA1-3	End.AMBK	- SFF Backup masquerading proxy	[This.ID]
TBA1-4	End.ASFB	- SF backup static proxy	[This.ID]
TBA1-5	End.ADFB	- SF backup dynamic proxy	[This.ID]
TBA1-6	End.AMFB	- SF backup masquerading proxy	[This.ID]
TBA1-7	End.ASBF	- Bypass forwarding static proxy	[This.ID]
TBA1-8	End.ADBF	- Bypass forwarding dynamic proxy	[This.ID]
TBA1-9	End.AMBF	- Bypass forwarding masquerading proxy	[This.ID]

Table 1

5.2. SRv6 Endpoint Flavor

This document requests IANA to allocate the following codepoints for PSD flavor behaviors within the "SRv6 Endpoint Behaviors" registry in the "Segment Routing" registry group.

Value	Hex	Endpoint behavior	Reference
TBA	TBA	End.AS with BAK	[This.ID]
TBA	TBA	End.AD with BAK	[This.ID]
TBA	TBA	End.AM with BAK	[This.ID]
TBA	TBA	End.AS with SFBK	[This.ID]
TBA	TBA	End.AD with SFBK	[This.ID]
TBA	TBA	End.AM with SFBK	[This.ID]
TBA	TBA	End.AS with BFWD	[This.ID]
TBA	TBA	End.AD with BFWD	[This.ID]
TBA	TBA	End.AM with BFWD	[This.ID]

Table 2

6. Security Considerations

The security requirements and mechanisms described in [RFC8402], [RFC8754] and [RFC8986] also apply to this document.

This document does not introduce any new security vulnerabilities.

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>.

[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>.

[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>.

[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", Work in Progress, Internet-Draft, draft-ietf-spring-sr-service-programming-11, 23 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-11>.

7.2. Informative References

TBD

8. Acknowledgments

The authors would like to thank the following for their valuable contributions of this document:

TBD

Authors' Addresses

Feng Yang
China Mobile
Beijing
China
Email: yangfeng@chinamobile.com

Changwang Lin
New H3C Technologies
Beijing
China

Email: linchangwang.04414@h3c.com

Yuanxiang Qiu
New H3C Technologies
China
Email: qiuyuanxiang@h3c.com

Xiaoqiu Zhang
China Mobile
Beijing
China
Email: zhangxiaoqiu@chinamobile.com