SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: 24 October 2025

C. Li, Ed.
Huawei
W. Cheng
China Mobile
H. Huang, Ed.
Huawei

22 April 2025

Compressed SID (CSID) for SRv6 SFC
draft-lh-spring-srv6-sfc-csid-04

Abstract

In SRv6, an SRv6 SID is a 128-bit value. When too many 128-bit SRv6 SIDs are included in an SRH, the introduced overhead will affect the transmission efficiency of payload. In order to address this problem, Compressed SID(CSID) is proposed. This document defines new behaviors for service segments with REPLACE-CSID and NEXT-CSID flavors to enable compressed SRv6 service programming.

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 24 October 2025.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing [RFC8402] is a source routing paradigm to support steering packets through a programmed path at the ingress node. Currently, two data planes are defined for Segment Routing: MPLS and IPv6. When IPv6 data plane is used in Segment Routing, it is called SRv6 [RFC8754] . [RFC8754] defines a new extension header in IPv6, called Segment Routing Header (SRH), to support SRv6. To support SRv6 network programming, [RFC8986] defines a framework to build a network program with topological and service segments carried in a Segment Routing header (SRH) [RFC8754].

A Service Function Chain (SFC) [RFC7665] defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows.

A service function chain can be implemented by SRv6 by using a sequence of SRv6 SIDs including service segments defined in [I-D.ietf-spring-sr-service-programming].

However, when too many 128-bit SRv6 SIDs are included in an SRH, the overhead of the SRH will affect the transmission efficiency of the payload. [I-D.ietf-spring-compression-requirement] points out the problem of long SRv6 SID lists reduce payload efficiency. To mitigate such overhead, [I-D.ietf-spring-srv6-srh-compression] defines new flavors for basic SR endpoint behaviors defined in [RFC8986]. Using the new flavored behavior SID, a 128-bit SRv6 SID can be compressed to be an 32-bit or 16-bit Compressed SID (CSID), which reduces a lot of size of the SRv6 header.

To enable SRv6 SID lists compression for service function chaining (SFC), this document defines new behaviors of service segments with flavors defined in [I-D.ietf-spring-srv6-srh-compression].

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

This document leverages the terms defined in [RFC8402], [RFC8754], [RFC8986], [I-D.ietf-spring-srv6-srh-compression] and [I-D.ietf-spring-sr-service-programming]. The reader is assumed to be familiar with this terminology. This document does not introduce any new terms.

2. SR Proxy Behaviors

[I-D.ietf-spring-sr-service-programming] defines several SRv6 endpoint behaviors for service proxy segments. A service proxy segment ID is represented as an 128-bit value just like other SIDs defined in [RFC8986]. This section defines some new behaviors of those service proxy segments by combining the existing service proxy segment behaviors with CSID flavors, such as REPLACE-CSID flavor and NEXT-CSID flavor.

The main difference between behaviors are the forwarding instructions. Therefore, when CSID compression mechanism applies to SR Proxy behaviors, the pseudo code of the new behaviors can be generated by updating the forwarding instructions of CSID to SR proxy forwarding instructions. The following sections define the details of the pseudo code of new behaviors.

2.1. Static SR Proxy

In [I-D.ietf-spring-sr-service-programming], an End.AS - Static proxy endpoint behavior is defined for the static SR proxy. According to the traffic type encapsulated in the SRv6 payload, the pseudo code of Ethernet, IPv4 and IPv6 is described as well in [I-D.ietf-spring-sr-service-programming]. This section defines the new behaviors of associate the NEXT-CSID and REPLACE-CSID Flavor to END.AS.

2.1.1. Static Proxy for Inner Type Ethernet

2.1.1.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor (a.k.a End.AS with REPLACE-CSID) for Ethernet traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01.   If (Upper-layer header type != 143 (Ethernet)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Perform IPv6 decapsulation.
P05.   Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
```

A rendering of the complete pseudocode is provided in Appendix A.1.

The upper-layer header processing is unchanged as per Section 6.1.2.1 of [I-D.ietf-spring-sr-service-programming].

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, as per Section 6.1.2.1 of [I-D.ietf-spring-sr-service-programming].

2.1.1.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor (a.k.a End.AS with NEXT-CSID) for Ethernet traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01.   If (Upper-layer header type != 143 (Ethernet)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
Q03.   }
Q04.   Perform IPv6 decapsulation.
Q05.   Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
```

A rendering of the complete pseudocode is provided in Appendix A.2.

The upper-layer header processing is unchanged as per Section 6.1.2.1 of [I-D.ietf-spring-sr-service-programming].

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, as per Section 6.1.2.1 of [I-D.ietf-spring-sr-service-programming].

2.1.2. Static Proxy for Inner Type IPv4

2.1.2.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor (a.k.a End.AS with REPLACE-CSID) for IPv4 traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01.   If (Upper-layer header type != 4 (IPv4)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Perform IPv6 decapsulation.
P05.   Submit the packet to the IPv4 module for transmission on
         interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.3.

The upper-layer header processing is unchanged as per Section 6.1.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.1.2.2 of [I-D.ietf-spring-sr-service-programming].

2.1.2.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor (a.k.a End.AS with NEXT-CSID) for IPv4 traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01.   If (Upper-layer header type != 4 (IPv4)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
Q03.   }
Q04.   Perform IPv6 decapsulation.
Q05.   Submit the packet to the IPv4 module for transmission on
         interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.4.

The upper-layer header processing is unchanged as per Section 6.1.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.1.2.2 of [I-D.ietf-spring-sr-service-programming].

2.1.3. Static Proxy for Inner Type IPv6

2.1.3.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor (a.k.a End.AS with REPLACE-CSID) for IPv6 traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01.   If (Upper-layer header type != 41 (IPv6)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Perform IPv6 decapsulation.
P05.   Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.5.

The upper-layer header processing is unchanged as per Section 6.1.2.3 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.1.2.3 of [I-D.ietf-spring-sr-service-programming].

2.1.3.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor (a.k.a End.AS with NEXT-CSID) for IPv6 traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01.   If (Upper-layer header type != 41 (IPv6)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
Q03.   }
Q04.   Perform IPv6 decapsulation.
Q05.   Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.6.

The upper-layer header processing is unchanged as per Section 6.1.2.3 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.1.2.3 of [I-D.ietf-spring-sr-service-programming].

2.2. Dynamic SR Proxy

In [I-D.ietf-spring-sr-service-programming], an End.AD - Dynamic proxy endpoint behavior is defined for the dynamic SR proxy. According to the traffic type encapsulated in the SRv6 payload, the pseudo code of Ethernet, IPv4 and IPv6 is described as well in [I-D.ietf-spring-sr-service-programming]. This section defines the new behaviors of associate the NEXT-CSID and REPLACE-CSID Flavor to END.AD.

2.2.1. Dynamic Proxy for Inner Type Ethernet

2.2.1.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor (a.k.a End.AD with REPLACE-CSID) for Ethernet traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01.   If (Upper-layer header type != 143 (Ethernet)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Copy the IPv6 encapsulation in a CACHE entry associated with
            the interface IFACE-IN.
P05.   Perform IPv6 decapsulation.
P06.   Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
```

A rendering of the complete pseudocode is provided in Appendix A.7.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.2.1.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor (a.k.a End.AD with NEXT-CSID) for Ethernet traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01.   If (Upper-layer header type != 143 (Ethernet)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
Q03.   }
Q04.   Copy the IPv6 encapsulation in a CACHE entry associated with
            the interface IFACE-IN.
Q05.   Perform IPv6 decapsulation.
Q06.   Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
```

A rendering of the complete pseudocode is provided in Appendix A.8.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an Ethernet frame received on the interface IFACE-IN and with a destination MAC address that is neither a broadcast address nor matches the address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.2.2. Dynamic Proxy for Inner Type IPv4

2.2.2.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor (a.k.a End.AD with REPLACE-CSID) for IPv4 traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01.   If (Upper-layer header type != 4 (IPv4)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Copy the IPv6 encapsulation in a CACHE entry associated with
            the interface IFACE-IN.
P05.   Perform IPv6 decapsulation.
P06.   Submit the frame to the IPv4 module for transmission via
         interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.9.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.2.2.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor (a.k.a End.AD with NEXT-CSID) for IPv4 traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01. If (Upper-layer header type != 4 (IPv4)) {
Q02.   Resubmit the packet to the IPv6 module for transmission to
         the new destination.
Q03. }
Q04. Copy the IPv6 encapsulation in a CACHE entry associated with
        the interface IFACE-IN.
Q05. Perform IPv6 decapsulation.
Q06. Submit the frame to the IPv4 module for transmission via
       interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.10.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv4 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.2.3. Dynamic Proxy for Inner Type IPv6

2.2.3.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor (a.k.a End.AD with REPLACE-CSID) for IPv6 traffic, the procedure described in Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01. If (Upper-layer header type != 41 (IPv6)) {
P02.   Resubmit the packet to the IPv6 module for transmission to
         the new destination.
P03. }
P04. Copy the IPv6 encapsulation in a CACHE entry associated with
        the interface IFACE-IN.
P05. Perform IPv6 decapsulation.
P06. Submit the frame to the IPv6 module for transmission via
      interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.11.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.2.3.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor (a.k.a End.AD with NEXT-CSID) for IPv6 traffic, the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01. If (Upper-layer header type != 41 (IPv6)) {
Q02.   Resubmit the packet to the IPv6 module for transmission to
         the new destination.
Q03. }
Q04. Copy the IPv6 encapsulation in a CACHE entry associated with
        the interface IFACE-IN.
Q05. Perform IPv6 decapsulation.
Q06. Submit the frame to the IPv6 module for transmission via
       interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.12.

The upper-layer header processing is unchanged as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, as per Section 6.2.2 of [I-D.ietf-spring-sr-service-programming].

2.3. Shared Memory SR Proxy

This document does not need to define new flavors for shared proxy behavior as the processing will be executed by using the shared memory Section 6.3 of [I-D.ietf-spring-sr-service-programming].

2.4. Masquerading SR Proxy

As per [I-D.ietf-spring-sr-service-programming], when forwarding packets to SR-unaware SFs, masquerading SR proxy sets the destination address of the IPv6 header as segment list[0] which is the original final destination address. When receiving the traffic returning from the service, de-masquerading sets the destination address as segment list[Segment Left].

To be consistent with the behavior of masquerading proxy, it’s required that any segment list containing one or more masquerading proxy CSID MUST NOT apply any compression encoding to the last segment (segment list[0]).

Note: The service receiving an IPv6 packet from the proxy uses the destination address (copied from last segment) as final destination and could apply certain actions based on that. In order to process and forward packets correctly, it is required that the last segment not be compressed.

To be consistent with the behavior of masquerading proxy, when processing an IPv6 packet matching a FIB entry locally instantiated as an SRv6 masquerading CSID, it’s required that the updated

destination address MUST be cached in the proxy by adding a dynamic caching mechanism similar to the one described in Section 6.2 of [I-D.ietf-spring-sr-service-programming] in case that segment list[Segment Left] is a compressed SID.

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the destination address MUST be recovered from CACHE in case that segment list[Segment Left] could be a CSID container.

2.4.1. SRv6 Masquerading Proxy Pseudocode

In [I-D.ietf-spring-sr-service-programming], an End.AM - Masquerading proxy endpoint behavior is defined for the Masquerading SR proxy. According to the traffic type encapsulated in the SRv6 payload, the pseudo code of Ethernet, IPv4 and IPv6 is described as well in [I-D.ietf-spring-sr-service-programming]. This section defines the new behaviors of associate the NEXT-CSID and REPLACE-CSID Flavor to END.AM.

2.4.1.1. REPLACE-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 masquerading proxy SID with the REPLACE-CSID flavor (a.k.a End.AM with REPLACE-CSID), the procedure described in Figure 23 from Section 4.2.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line R10 and R21 that are both replaced as follows.

```
P01. Copy the IPv6 Destination Address in a CACHE entry associated
      with the interface IFACE-IN.
P02. Copy Segment List[0] from the SRH to the Destination Address
      of the IPv6 header.
P03. Submit the packet to the IPv6 module for transmission on
      interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.13.

De-masquerading: When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the procedure described in Figure 24 from Section 6.4.1 of [I-D.ietf-spring-sr-service-programming] is executed except for line S10 that is replaced as follows.

```
D01. Retrieve the CACHE entry associated with IFACE-IN.
D02. If the CACHE entry is not empty {
D03.    Destination Address of the IPv6 header is set to CACHE.
D04. }
```

A rendering of the complete pseudocode is provided in Appendix A.14.

2.4.1.2. NEXT-CSID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 masquerading proxy SID with the NEXT-CSID flavor (a.k.a End.AM with NEXT-CSID), the procedure described in Section 4.1.1 of [I-D.ietf-spring-srv6-srh-compression] is executed except for line N08 of that and line S15 of Section 4.1 of [RFC8986] that are both replaced as follows.

```
Q01. Copy the IPv6 Destination Address in a CACHE entry associated
      with the interface IFACE-IN.
Q02. Copy Segment List[0] from the SRH to the Destination Address
      of the IPv6 header.
Q03. Submit the packet to the IPv6 module for transmission on
      interface IFACE-OUT via NH-ADDR.
```

A rendering of the complete pseudocode is provided in Appendix A.15}.

De-masquerading: When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN, the procedure described in Section 6.4.1 of [I-D.ietf-spring-sr-service-programming] is executed except for line S10 that is replaced as follows.

```
E01. Retrieve the CACHE entry associated with IFACE-IN.
E02. If the CACHE entry is not empty {
E03.    Destination Address of the IPv6 header is set to CACHE.
E04. }
```

A rendering of the complete pseudocode is provided in Appendix A.16.

2.4.2. Destination NAT Flavor

2.4.2.1. REPLACE-CSID Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the REPLACE-CSID (a.k.a End.AM- Masquerading proxy with NAT with REPLACE-CSID, End.AMN with REPLACE-CSID for short) is executed except for line S09.1 and S10 in Section 6.4.2 of [I-D.ietf-spring-sr-service-programming] replaced as follows.

```
P01. Copy the Destination Address of the IPv6 header to the
      Segment List[0] entry of the SRH.
P02. Retrieve the CACHE entry associated with IFACE-IN.
P03. If the CACHE entry is not empty {
P04.    Destination Address of the IPv6 header is set to CACHE.
P05. }
```

A rendering of the complete pseudocode is provided in Appendix A.17.

2.4.2.2. NEXT-CSID Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the NEXT-CSID (a.k.a End.AM- Masquerading proxy with NAT with NEXT-CSID, End.AMN with NEXT-CSID for short) is executed except for line S09.1 and S10 in Section 6.4.2 of [I-D.ietf-spring-sr-service-programming] replaced as follows.

```
Q01. Copy the Destination Address of the IPv6 header to the
      Segment List[0] entry of the SRH.
Q02. Retrieve the CACHE entry associated with IFACE-IN.
Q03. If the CACHE entry is not empty {
Q04.    Destination Address of the IPv6 header is set to CACHE.
Q05. }
```

A rendering of the complete pseudocode is provided in Appendix A.18.

2.4.3. Cache Flavor

The caching flavor of the SRv6 masquerading proxy with CSID is enabled as per Section 6.4.3 of [I-D.ietf-spring-sr-service-programming] without any modification.

3. Security Considerations

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

This document does not introduce any new security considerations.

4. IANA Considerations

4.1. SRv6 Endpoint Behaviors

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
180	End.AS with REPLACE-CSID	[This.ID]
181	End.AS with NEXT-CSID	[This.ID]
182	End.AD with REPLACE-CSID	[This.ID]
183	End.AD with NEXT-CSID	[This.ID]
184	End.AM with REPLACE-CSID	[This.ID]
185	End.AM with NEXT-CSID	[This.ID]
186	End.AMN with REPLACE-CSID	[This.ID]
187	End.AMN with NEXT-CSID	[This.ID]

5. References

5.1. 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/rfc/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/rfc/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/rfc/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/rfc/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/rfc/rfc8986>.

[I-D.ietf-spring-srv6-srh-compression] Cheng, W., Filsfils, C., Li, Z., Decraene, B., and F. Clad, "Compressed SRv6 Segment List Encoding (CSID)", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-srh-compression-23, 6 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-srv6-srh-compression-23>.

[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>.

5.2. Informative References

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/rfc/rfc7665>.

[I-D.ietf-spring-compression-requirement]
Cheng, W., Xie, C., Bonica, R., Dukes, D., Li, C., Peng, S., and W. Henderickx, "Compressed SRv6 SID List Requirements", Work in Progress, Internet-Draft, draft-ietf-spring-compression-requirement-03, 3 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-compression-requirement-03>.

Appendix A. Complete Pseudocodes

A.1. REPLACE-CSID Flavor for Static Proxy for Inner Type Ethernet

When processing the SRH of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor for Ethernet traffic:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
           Segment List[0][DA.Arg.Index-1] == 0)) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (Hop limit exceeded in transit),
           interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
R01.   If (DA.Arg.Index != 0) {
R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
R03.       Send an ICMP Parameter Problem to the Source Address,
```

```
                Code 0 (Erroneous header field encountered),
                Pointer set to the Segments Left field,
                interrupt packet processing and discard the packet.
   R04.       }
   R05.       Decrement DA.Arg.Index by 1.
   R06.       If (Segment List[Segments Left][DA.Arg.Index] == 0) {
   R07.         Decrement Segments Left by 1.
   R08.         Decrement IPv6 Hop Limit by 1.
   R09.         Update IPv6 DA with Segment List[Segments Left]
   P01.         If (Upper-layer header type != 143 (Ethernet)) {
   P02.           Resubmit the packet to the IPv6 module for
                     transmission to the new destination.
   P03.         }
   P04.         Perform IPv6 decapsulation.
   P05.         Submit the frame to the Ethernet module for transmission
                  via interface IFACE-OUT.
   R11.       }
   R12.     } Else {
   R13.       If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
   R14.         Send an ICMP Parameter Problem to the Source Address,
                  Code 0 (Erroneous header field encountered),
                  Pointer set to the Segments Left field,
                  interrupt packet processing and discard the packet.
   R15.       }
   R16.       Decrement Segments Left by 1.
   R17.       Set DA.Arg.Index to (128/LNFL - 1).
   R18.     }
   R19.     Decrement IPv6 Hop Limit by 1.
   R20.     Write Segment List[Segments Left][DA.Arg.Index] into the bits
              [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
              header.
   P01.     If (Upper-layer header type != 143 (Ethernet)) {
   P02.       Resubmit the packet to the IPv6 module for transmission to
                the new destination.
   P03.     }
   P04.     Perform IPv6 decapsulation.
   P05.     Submit the frame to the Ethernet module for transmission via
              interface IFACE-OUT.
   S16. }
```

A.2. NEXT-CSID Flavor for Static Proxy for Inner Type Ethernet

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor for Ethernet traffic:

```
   N01. If (DA.Argument != 0) {
   N02.   If (IPv6 Hop Limit <= 1) {
   N03.     Send an ICMP Time Exceeded message to the Source Address
              with Code 0 (Hop limit exceeded in transit),
              interrupt packet processing, and discard the packet.
   N04.   }
   N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
            of the Destination Address.
   N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
            zero.
   N07.   Decrement IPv6 Hop Limit by 1.
   Q01.   If (Upper-layer header type != 143 (Ethernet)) {
   Q02.     Resubmit the packet to the IPv6 module for transmission to
              the new destination.
   Q03.   }
   Q04.   Perform IPv6 decapsulation.
   Q05.   Submit the frame to the Ethernet module for transmission via
              interface IFACE-OUT.
   N09. }
   S02. If (Segments Left == 0) {
   S03.   Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. If (Upper-layer header type != 143 (Ethernet)) {
   Q02.   Resubmit the packet to the IPv6 module for transmission to
            the new destination.
   Q03. }
   Q04. Perform IPv6 decapsulation.
   Q05. Submit the frame to the Ethernet module for transmission via
          interface IFACE-OUT.
```

A.3. REPLACE-CSID Flavor for Static Proxy for Inner Type IPv4

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor for IPv4 traffic:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
           Segment List[0][DA.Arg.Index-1] == 0)) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (Hop limit exceeded in transit),
           interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
R01.   If (DA.Arg.Index != 0) {
R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
R03.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R04.     }
R05.     Decrement DA.Arg.Index by 1.
R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
R07.       Decrement Segments Left by 1.
R08.       Decrement IPv6 Hop Limit by 1.
R09.       Update IPv6 DA with Segment List[Segments Left]
P01.       If (Upper-layer header type != 4 (IPv4)) {
P02.         Resubmit the packet to the IPv6 module for transmission
               to the new destination.
P03.       }
P04.       Perform IPv6 decapsulation.
P05.       Submit the packet to the IPv4 module for transmission on
             interface IFACE-OUT via NH-ADDR.
R11.     }
R12.   } Else {
R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
R14.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R15.     }
R16.     Decrement Segments Left by 1.
R17.     Set DA.Arg.Index to (128/LNFL - 1).
```

```
   R18.   }
   R19.   Decrement IPv6 Hop Limit by 1.
   R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
            [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
            header.
   P01.   If (Upper-layer header type != 4 (IPv4)) {
   P02.     Resubmit the packet to the IPv6 module for transmission to
              the new destination.
   P03.   }
   P04.   Perform IPv6 decapsulation.
   P05.   Submit the packet to the IPv4 module for transmission on
            interface IFACE-OUT via NH-ADDR.
   S16. }
```

A.4. NEXT-CSID Flavor for Static Proxy for Inner Type IPv4

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor for IPv4 traffic:

```
   N01. If (DA.Argument != 0) {
   N02.   If (IPv6 Hop Limit <= 1) {
   N03.     Send an ICMP Time Exceeded message to the Source Address
              with Code 0 (Hop limit exceeded in transit),
              interrupt packet processing, and discard the packet.
   N04.   }
   N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
            of the Destination Address.
   N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
            zero.
   N07.   Decrement IPv6 Hop Limit by 1.
   Q01.   If (Upper-layer header type != 4 (IPv4)) {
   Q02.     Resubmit the packet to the IPv6 module for transmission to
              the new destination.
   Q03.   }
   Q04.   Perform IPv6 decapsulation.
   Q05.   Submit the packet to the IPv4 module for transmission on
            interface IFACE-OUT via NH-ADDR.
   N09. }
   S02. If (Segments Left == 0) {
   S03.   Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. If (Upper-layer header type != 4 (IPv4)) {
   Q02.   Resubmit the packet to the IPv6 module for transmission to
            the new destination.
   Q03. }
   Q04. Perform IPv6 decapsulation.
   Q05. Submit the packet to the IPv4 module for transmission on
          interface IFACE-OUT via NH-ADDR.
```

A.5. REPLACE-CSID Flavor for Static Proxy for Inner Type IPv6

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-CSID flavor for IPv6 traffic:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
           Segment List[0][DA.Arg.Index-1] == 0)) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (Hop limit exceeded in transit),
           interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
R01.   If (DA.Arg.Index != 0) {
R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
R03.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R04.     }
R05.     Decrement DA.Arg.Index by 1.
R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
R07.       Decrement Segments Left by 1.
R08.       Decrement IPv6 Hop Limit by 1.
R09.       Update IPv6 DA with Segment List[Segments Left]
P01.     If (Upper-layer header type != 41 (IPv6)) {
P02.       Resubmit the packet to the IPv6 module for transmission to
             the new destination.
P03.     }
P04.     Perform IPv6 decapsulation.
P05.     Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
R11.     }
R12.   } Else {
R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
R14.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R15.     }
R16.     Decrement Segments Left by 1.
R17.     Set DA.Arg.Index to (128/LNFL - 1).
```

```
R18.   }
R19.   Decrement IPv6 Hop Limit by 1.
R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
         [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
         header.
P01.   If (Upper-layer header type != 41 (IPv6)) {
P02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
P03.   }
P04.   Perform IPv6 decapsulation.
P05.   Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
S16. }
```

A.6. NEXT-CSID Flavor for Static Proxy for Inner Type IPv6

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-CSID flavor for IPv6 traffic:

```
   N01. If (DA.Argument != 0) {
   N02.   If (IPv6 Hop Limit <= 1) {
   N03.     Send an ICMP Time Exceeded message to the Source Address
              with Code 0 (Hop limit exceeded in transit),
              interrupt packet processing, and discard the packet.
   N04.   }
   N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
            of the Destination Address.
   N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
            zero.
   N07.   Decrement IPv6 Hop Limit by 1.
   Q01.   If (Upper-layer header type != 41 (IPv6)) {
   Q02.     Resubmit the packet to the IPv6 module for transmission to
              the new destination.
   Q03.   }
   Q04.   Perform IPv6 decapsulation.
   Q05.   Submit the packet to the IPv6 module for transmission on
            interface IFACE-OUT via NH-ADDR.
   N09. }
   S02. If (Segments Left == 0) {
   S03.   Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. If (Upper-layer header type != 41 (IPv6)) {
   Q02.   Resubmit the packet to the IPv6 module for transmission to
            the new destination.
   Q03. }
   Q04. Perform IPv6 decapsulation.
   Q05. Submit the packet to the IPv6 module for transmission on
          interface IFACE-OUT via NH-ADDR.
```

A.7. REPLACE-CSID Flavor for Dynamic Proxy for Inner Type Ethernet

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor for Ethernet traffic:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
           Segment List[0][DA.Arg.Index-1] == 0)) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (Hop limit exceeded in transit),
           interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
R01.   If (DA.Arg.Index != 0) {
R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
R03.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R04.     }
R05.     Decrement DA.Arg.Index by 1.
R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
R07.       Decrement Segments Left by 1.
R08.       Decrement IPv6 Hop Limit by 1.
R09.       Update IPv6 DA with Segment List[Segments Left]
P01.       If (Upper-layer header type != 143 (Ethernet)) {
P02.         Resubmit the packet to the IPv6 module for transmission to
               the new destination.
P03.       }
P04.       Copy the IPv6 encapsulation in a CACHE entry associated with
                the interface IFACE-IN.
P05.       Perform IPv6 decapsulation.
P06.       Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
R11.     }
R12.   } Else {
R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
R14.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R15.     }
```

```
 R16.     Decrement Segments Left by 1.
 R17.     Set DA.Arg.Index to (128/LNFL - 1).
 R18.   }
 R19.   Decrement IPv6 Hop Limit by 1.
 R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
          [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
          header.
 P01.   If (Upper-layer header type != 143 (Ethernet)) {
 P02.     Resubmit the packet to the IPv6 module for transmission to
            the new destination.
 P03.   }
 P04.   Copy the IPv6 encapsulation in a CACHE entry associated with
             the interface IFACE-IN.
 P05.   Perform IPv6 decapsulation.
 P06.   Submit the frame to the Ethernet module for transmission via
          interface IFACE-OUT.
 S16. }
```

A.8. NEXT-CSID Flavor for Dynamic Proxy for Inner Type Ethernet

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor for Ethernet traffic:

```
N01. If (DA.Argument != 0) {
N02.   If (IPv6 Hop Limit <= 1) {
N03.     Send an ICMP Time Exceeded message to the Source Address
           with Code 0 (Hop limit exceeded in transit),
           interrupt packet processing, and discard the packet.
N04.   }
N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
         of the Destination Address.
N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
         zero.
N07.   Decrement IPv6 Hop Limit by 1.
Q01.   If (Upper-layer header type != 143 (Ethernet)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
           the new destination.
Q03.   }
Q04.   Copy the IPv6 encapsulation in a CACHE entry associated with
            the interface IFACE-IN.
Q05.   Perform IPv6 decapsulation.
Q06.   Submit the frame to the Ethernet module for transmission via
         interface IFACE-OUT.
N09. }
S02. If (Segments Left == 0) {
S03.   Stop processing the SRH, and proceed to process the next
         header in the packet, whose type is identified by
```

```
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. If (Upper-layer header type != 143 (Ethernet)) {
   Q02.   Resubmit the packet to the IPv6 module for transmission to
           the new destination.
   Q03. }
   Q04. Copy the IPv6 encapsulation in a CACHE entry associated with
          the interface IFACE-IN.
   Q05. Perform IPv6 decapsulation.
   Q06. Submit the frame to the Ethernet module for transmission via
          interface IFACE-OUT.
```

A.9. REPLACE-CSID Flavor for Dynamic Proxy for Inner Type IPv4

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor for IPv4 traffic:

```
 S01. When an SRH is processed {
 S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
           Segment List[0][DA.Arg.Index-1] == 0)) {
 S03.     Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
 S04.   }
 S05.   If (IPv6 Hop Limit <= 1) {
 S06.     Send an ICMP Time Exceeded message to the Source Address,
            Code 0 (Hop limit exceeded in transit),
            interrupt packet processing and discard the packet.
 S07.   }
 S08.   max_LE = (Hdr Ext Len / 2) - 1
 R01.   If (DA.Arg.Index != 0) {
 R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
```

```
 R03.       Send an ICMP Parameter Problem to the Source Address,
              Code 0 (Erroneous header field encountered),
              Pointer set to the Segments Left field,
              interrupt packet processing and discard the packet.
 R04.     }
 R05.     Decrement DA.Arg.Index by 1.
 R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
 R07.       Decrement Segments Left by 1.
 R08.       Decrement IPv6 Hop Limit by 1.
 R09.       Update IPv6 DA with Segment List[Segments Left]
 P01.       If (Upper-layer header type != 4 (IPv4)) {
 P02.        Resubmit the packet to the IPv6 module for transmission to
                the new destination.
 P03.       }
 P04.       Copy the IPv6 encapsulation in a CACHE entry associated with
                 the interface IFACE-IN.
 P05.       Perform IPv6 decapsulation.
 P06.       Submit the frame to the IPv4 module for transmission via
              interface IFACE-OUT via NH-ADDR.
 R11.     }
 R12.   } Else {
 R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
 R14.       Send an ICMP Parameter Problem to the Source Address,
              Code 0 (Erroneous header field encountered),
              Pointer set to the Segments Left field,
              interrupt packet processing and discard the packet.
 R15.     }
 R16.     Decrement Segments Left by 1.
 R17.     Set DA.Arg.Index to (128/LNFL - 1).
 R18.   }
 R19.   Decrement IPv6 Hop Limit by 1.
 R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
          [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
          header.
 P01.   If (Upper-layer header type != 4 (IPv4)) {
 P02.     Resubmit the packet to the IPv6 module for transmission to
            the new destination.
 P03.   }
 P04.   Copy the IPv6 encapsulation in a CACHE entry associated with
             the interface IFACE-IN.
 P05.   Perform IPv6 decapsulation.
 P06.   Submit the frame to the IPv4 module for transmission via
          interface IFACE-OUT via NH-ADDR.
 S16. }
```

A.10. NEXT-CSID Flavor for Dynamic Proxy for Inner Type IPv4

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor for IPv4 traffic:

```
N01. If (DA.Argument != 0) {
N02.   If (IPv6 Hop Limit <= 1) {
N03.     Send an ICMP Time Exceeded message to the Source Address
           with Code 0 (Hop limit exceeded in transit),
           interrupt packet processing, and discard the packet.
N04.   }
N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
         of the Destination Address.
N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
         zero.
N07.   Decrement IPv6 Hop Limit by 1.
Q01.   If (Upper-layer header type != 4 (IPv4)) {
Q02.     Resubmit the packet to the IPv6 module for transmission to
          the new destination.
Q03.   }
Q04.   Copy the IPv6 encapsulation in a CACHE entry associated with
          the interface IFACE-IN.
Q05.   Perform IPv6 decapsulation.
Q06.   Submit the frame to the IPv4 module for transmission via
          interface IFACE-OUT via NH-ADDR.
N09. }
S02. If (Segments Left == 0) {
S03.   Stop processing the SRH, and proceed to process the next
         header in the packet, whose type is identified by
         the Next Header field in the routing header.
S04. }
S05. If (IPv6 Hop Limit <= 1) {
S06.   Send an ICMP Time Exceeded message to the Source Address
         with Code 0 (Hop limit exceeded in transit),
         interrupt packet processing, and discard the packet.
S07. }
S08. max_LE = (Hdr Ext Len / 2) - 1
S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.   Send an ICMP Parameter Problem to the Source Address
         with Code 0 (Erroneous header field encountered)
         and Pointer set to the Segments Left field,
         interrupt packet processing, and discard the packet.
S11. }
S12. Decrement IPv6 Hop Limit by 1.
S13. Decrement Segments Left by 1.
S14. Update IPv6 DA with Segment List[Segments Left].
Q01. If (Upper-layer header type != 4 (IPv4)) {
```

```
   Q02.   Resubmit the packet to the IPv6 module for transmission to
           the new destination.
   Q03. }
   Q04. Copy the IPv6 encapsulation in a CACHE entry associated with
           the interface IFACE-IN.
   Q05. Perform IPv6 decapsulation.
   Q06. Submit the frame to the IPv4 module for transmission via
         interface IFACE-OUT via NH-ADDR.
```

A.11. REPLACE-CSID Flavor for Dynamic Proxy for Inner Type IPv6

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-CSID flavor for IPv6 traffic:

```
   S01. When an SRH is processed {
   S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
              Segment List[0][DA.Arg.Index-1] == 0)) {
   S03.     Stop processing the SRH, and proceed to process the next
              header in the packet, whose type is identified by
              the Next Header field in the routing header.
   S04.   }
   S05.   If (IPv6 Hop Limit <= 1) {
   S06.     Send an ICMP Time Exceeded message to the Source Address,
              Code 0 (Hop limit exceeded in transit),
              interrupt packet processing and discard the packet.
   S07.   }
   S08.   max_LE = (Hdr Ext Len / 2) - 1
   R01.   If (DA.Arg.Index != 0) {
   R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
   R03.       Send an ICMP Parameter Problem to the Source Address,
                Code 0 (Erroneous header field encountered),
                Pointer set to the Segments Left field,
                interrupt packet processing and discard the packet.
   R04.     }
   R05.     Decrement DA.Arg.Index by 1.
   R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
   R07.       Decrement Segments Left by 1.
   R08.       Decrement IPv6 Hop Limit by 1.
   R09.       Update IPv6 DA with Segment List[Segments Left]
   P01.       If (Upper-layer header type != 41 (IPv6)) {
   P02.       Resubmit the packet to the IPv6 module for transmission to
              the new destination.
   P03.       }
   P04.     Copy the IPv6 encapsulation in a CACHE entry associated with
              the interface IFACE-IN.
   P05.     Perform IPv6 decapsulation.
   P06.     Submit the frame to the IPv6 module for transmission via
```

```
            interface IFACE-OUT via NH-ADDR.
   R11.     }
   R12.   } Else {
   R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
   R14.       Send an ICMP Parameter Problem to the Source Address,
                Code 0 (Erroneous header field encountered),
                Pointer set to the Segments Left field,
                interrupt packet processing and discard the packet.
   R15.     }
   R16.     Decrement Segments Left by 1.
   R17.     Set DA.Arg.Index to (128/LNFL - 1).
   R18.   }
   R19.   Decrement IPv6 Hop Limit by 1.
   R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
            [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
            header.
   P01.   If (Upper-layer header type != 41 (IPv6)) {
   P02.     Resubmit the packet to the IPv6 module for transmission to
             the new destination.
   P03.   }
   P04.   Copy the IPv6 encapsulation in a CACHE entry associated with
             the interface IFACE-IN.
   P05.   Perform IPv6 decapsulation.
   P06.   Submit the frame to the IPv6 module for transmission via
             interface IFACE-OUT via NH-ADDR.
   S16. }
```

A.12. NEXT-CSID Flavor for Dynamic Proxy for Inner Type IPv6

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-CSID flavor for IPv6 traffic:

```
   N01. If (DA.Argument != 0) {
   N02.   If (IPv6 Hop Limit <= 1) {
   N03.     Send an ICMP Time Exceeded message to the Source Address
              with Code 0 (Hop limit exceeded in transit),
              interrupt packet processing, and discard the packet.
   N04.   }
   N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
            of the Destination Address.
   N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
            zero.
   N07.   Decrement IPv6 Hop Limit by 1.
   Q01.   If (Upper-layer header type != 41 (IPv6)) {
   Q02.     Resubmit the packet to the IPv6 module for transmission to
             the new destination.
   Q03.   }
```

```
   Q04.   Copy the IPv6 encapsulation in a CACHE entry associated with
             the interface IFACE-IN.
   Q05.   Perform IPv6 decapsulation.
   Q06.   Submit the frame to the IPv6 module for transmission via
            interface IFACE-OUT via NH-ADDR.
   N09. }
   S02. If (Segments Left == 0) {
   S03.   Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. If (Upper-layer header type != 41 (IPv6)) {
   Q02.   Resubmit the packet to the IPv6 module for transmission to
            the new destination.
   Q03. }
   Q04. Copy the IPv6 encapsulation in a CACHE entry associated with
            the interface IFACE-IN.
   Q05. Perform IPv6 decapsulation.
   Q06. Submit the frame to the IPv6 module for transmission via
          interface IFACE-OUT via NH-ADDR.
```

A.13. REPLACE-CSID Flavor for SRv6 Masquerading Proxy

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 masquerading proxy SID with the REPLACE-CSID flavor:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0 and (DA.Arg.Index == 0 or
             Segment List[0][DA.Arg.Index-1] == 0)) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
```

```
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address,
           Code 0 (Hop limit exceeded in transit),
           interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
R01.   If (DA.Arg.Index != 0) {
R02.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
R03.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R04.     }
R05.     Decrement DA.Arg.Index by 1.
R06.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
R07.       Decrement Segments Left by 1.
R08.       Decrement IPv6 Hop Limit by 1.
R09.       Update IPv6 DA with Segment List[Segments Left]
P01.       Copy the IPv6 Destination Address in a CACHE entry associated
           with the interface IFACE-IN.
P02.       Copy Segment List[0] from the SRH to the Destination Address
             of the IPv6 header.
P03.       Submit the packet to the IPv6 module for transmission on
             interface IFACE-OUT via NH-ADDR.
R11.     }
R12.   } Else {
R13.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
R14.       Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
R15.     }
R16.     Decrement Segments Left by 1.
R17.     Set DA.Arg.Index to (128/LNFL - 1).
R18.   }
R19.   Decrement IPv6 Hop Limit by 1.
R20.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
         [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
         header.
P01.   Copy the IPv6 Destination Address in a CACHE entry associated
        with the interface IFACE-IN.
P02.   Copy Segment List[0] from the SRH to the Destination Address
        of the IPv6 header.
P03.   Submit the packet to the IPv6 module for transmission on
        interface IFACE-OUT via NH-ADDR.
S16. }
```

A.14. REPLACE-CSID Flavor for SRv6 De-masquerading Proxy

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN:

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
D01.     Retrieve the CACHE entry associated with IFACE-IN.
D02.     If the CACHE entry is not empty {
D03.        Destination Address of the IPv6 header is set to CACHE.
D04.     }
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

A.15. NEXT-CSID Flavor for SRv6 Masquerading Proxy

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 masquerading proxy SID with the NEXT-CSID flavor:

```
   N01. If (DA.Argument != 0) {
   N02.   If (IPv6 Hop Limit <= 1) {
   N03.     Send an ICMP Time Exceeded message to the Source Address
              with Code 0 (Hop limit exceeded in transit),
              interrupt packet processing, and discard the packet.
   N04.   }
   N05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
            of the Destination Address.
   N06.   Set the bits [(LBL+AL)..127] of the Destination Address to
            zero.
   N07.   Decrement IPv6 Hop Limit by 1.
   Q01.    Copy the IPv6 Destination Address in a CACHE entry associated
            with the interface IFACE-IN.
   Q02.    Copy Segment List[0] from the SRH to the Destination Address
            of the IPv6 header.
   Q03.    Submit the packet to the IPv6 module for transmission on
            interface IFACE-OUT via NH-ADDR.
   N09. }
   S02. If (Segments Left == 0) {
   S03.   Stop processing the SRH, and proceed to process the next
            header in the packet, whose type is identified by
            the Next Header field in the routing header.
   S04. }
   S05. If (IPv6 Hop Limit <= 1) {
   S06.   Send an ICMP Time Exceeded message to the Source Address
            with Code 0 (Hop limit exceeded in transit),
            interrupt packet processing, and discard the packet.
   S07. }
   S08. max_LE = (Hdr Ext Len / 2) - 1
   S09. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
   S10.   Send an ICMP Parameter Problem to the Source Address
            with Code 0 (Erroneous header field encountered)
            and Pointer set to the Segments Left field,
            interrupt packet processing, and discard the packet.
   S11. }
   S12. Decrement IPv6 Hop Limit by 1.
   S13. Decrement Segments Left by 1.
   S14. Update IPv6 DA with Segment List[Segments Left].
   Q01. Copy the IPv6 Destination Address in a CACHE entry associated
         with the interface IFACE-IN.
   Q02. Copy Segment List[0] from the SRH to the Destination Address
         of the IPv6 header.
   Q03. Submit the packet to the IPv6 module for transmission on
         interface IFACE-OUT via NH-ADDR.
```

A.16. NEXT-CSID Flavor for SRv6 De-masquerading Proxy

When processing an IPv6 packet received on the interface IFACE-IN and with a destination address that does not match any address of IFACE-IN:

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
E01.     Retrieve the CACHE entry associated with IFACE-IN.
E02.     If the CACHE entry is not empty {
E03.        Destination Address of the IPv6 header is set to CACHE.
E04.     }
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

A.17. REPLACE-CSID Flavor for Destination NAT Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the REPLACE-CSID is executed:

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
P01.      Copy the Destination Address of the IPv6 header to the
           Segment List[0] entry of the SRH.
P02.      Retrieve the CACHE entry associated with IFACE-IN.
P03.      If the CACHE entry is not empty {
P04.         Destination Address of the IPv6 header is set to CACHE.
P05.      }
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

A.18. NEXT-CSID Flavor for Destination NAT Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the NEXT-CSID is executed:

```
S01. When an SRH is processed {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Send an ICMP Time Exceeded message to the Source Address,
         Code 0 (hop limit exceeded in transit),
         Interrupt packet processing and discard the packet.
S04.   }
S05.   If (Segments Left != 0) {
S06.     max_last_entry = (Hdr Ext Len / 2) - 1
S07.     If ((Last Entry > max_last_entry) or
             (Segments Left > Last Entry)) {
S08.       Send an ICMP Parameter Problem message to the Source Address,
           Code 0 (Erroneous header field encountered),
           Pointer set to the Segments Left field,
           Interrupt packet processing and discard the packet.
S09.     }
Q01.      Copy the Destination Address of the IPv6 header to the
           Segment List[0] entry of the SRH.
Q02.      Retrieve the CACHE entry associated with IFACE-IN.
Q03.      If the CACHE entry is not empty {
Q04.         Destination Address of the IPv6 header is set to CACHE.
Q05.      }
S11.   }
S12.   Decrement Hop Limit by 1.
S13.   Submit the packet to the IPv6 module for transmission to the
       next destination.
S14. }
```

Acknowledgements

TBD.

Authors' Addresses

Cheng Li (editor)
Huawei
China
Email: c.l@huawei.com

Weiqiang Cheng
China Mobile
China
Email: chengweiqiang@chinamobile.com

Hongyi Huang (editor)
Huawei
China

Email: hongyi.huang@huawei.com