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 (C-SID) is proposed. This document defines new behaviors for service segments with REPLACE-C-SID and NEXT-C-SID 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 1 September 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in
Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
   1.1. Requirements Language
   1.2. Terminology
2. SR Proxy Behaviors
   2.1. Static SR Proxy
      2.1.1. Static Proxy for Inner Type Ethernet
      2.1.2. Static Proxy for Inner Type IPv4
      2.1.3. Static Proxy for Inner Type IPv6
   2.2. Dynamic SR Proxy
      2.2.1. Dynamic Proxy for Inner Type Ethernet
      2.2.2. Dynamic Proxy for Inner Type IPv4
      2.2.3. Dynamic Proxy for Inner Type IPv6
   2.3. Shared Memory SR Proxy
3. Security Considerations
4. IANA Considerations
5. References
   5.1. Normative References
   5.2. Informative References
Appendix A. Complete Pseudocodes
   A.1. REPLACE-C-SID Flavor for Static Proxy for Inner Type Ethernet
   A.2. NEXT-C-SID Flavor for Static Proxy for Inner Type Ethernet
   A.3. REPLACE-C-SID Flavor for Static Proxy for Inner Type IPv4
   A.4. NEXT-C-SID Flavor for Static Proxy for Inner Type IPv4
   A.5. REPLACE-C-SID Flavor for Static Proxy for Inner Type IPv6
   A.6. NEXT-C-SID Flavor for Static Proxy for Inner Type IPv6
   A.7. REPLACE-C-SID Flavor for Dynamic Proxy for Inner Type Ethernet
   A.8. NEXT-C-SID Flavor for Dynamic Proxy for Inner Type Ethernet
   A.9. REPLACE-C-SID Flavor for Dynamic Proxy for Inner Type IPv4
   A.10. NEXT-C-SID Flavor for Dynamic Proxy for Inner Type IPv4
   A.11. REPLACE-C-SID Flavor for Dynamic Proxy for Inner Type IPv6
   A.12. NEXT-C-SID Flavor for Dynamic Proxy for Inner Type IPv6
   A.13. REPLACE-C-SID Flavor for SRv6 Masquerading Proxy
   A.14. REPLACE-C-SID Flavor for SRv6 De-masquerading Proxy
   A.15. NEXT-C-SID Flavor for SRv6 Masquerading Proxy
   A.16. NEXT-C-SID Flavor for SRv6 De-masquerading Proxy
   A.17. REPLACE-C-SID Flavor for Destination NAT Flavor
   A.18. NEXT-C-SID Flavor for Destination NAT Flavor
Acknowledgements
Authors' Addresses

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.srcompdt-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 (C-SID), 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 C-SID flavors, such as REPLACE-C-SID flavor and NEXT-C-SID flavor.

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

2.1. Static SR Proxy

2.1.1. Static Proxy for Inner Type Ethernet

2.1.1.1. REPLACE-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-C-SID flavor 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-srv6-srh-compression].

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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-C-SID flavor 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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-C-SID flavor 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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the NEXT-C-SID flavor 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 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 6.1.2.2 of [I-D.ietf-spring-sr-service-programming].

2.1.3. Static Proxy for Inner Type IPv6

2.1.3.1. REPLACE-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 static proxy SID with the REPLACE-C-SID flavor 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-C-SID Flavor

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

2.2.1. Dynamic Proxy for Inner Type Ethernet

2.2.1.1. REPLACE-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-C-SID flavor 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.
2.2.1.2. NEXT-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-C-SID flavor 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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-C-SID flavor 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.

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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-C-SID flavor 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.

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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the REPLACE-C-SID flavor 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.

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-C-SID Flavor

When processing an IPv6 packet that matches a FIB entry locally instantiated as an SRv6 dynamic proxy SID with the NEXT-C-SID flavor 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 define new flavors for shared proxy behavior as it is just an SR endpoint behavior.

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 C-SID 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 C-SID, 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...
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 C-SID container.

### 2.4.1. SRv6 Masquerading Proxy Pseudocode

#### 2.4.1.1. REPLACE-C-SID Flavor

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

1. Copy the IPv6 Destination Address in a CACHE entry associated with the interface IFACE-IN.
2. Copy Segment List[0] from the SRH to the Destination Address of the IPv6 header.
3. 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.

1. Retrieve the CACHE entry associated with IFACE-IN.
2. If the CACHE entry is not empty {
   3. Destination Address of the IPv6 header is set to CACHE.
5. }

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

#### 2.4.1.2. NEXT-C-SID Flavor

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

The Destination NAT flavor of the SRv6 masquerading proxy with the REPLACE-C-SID 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-C-SID Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the NEXT-C-SID 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 C-SID 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

TBD

5. References

5.1. Normative References


5.2. Informative References


Appendix A. Complete Pseudocodes

A.1. REPLACE-C-SID 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-C-SID flavor for Ethernet traffic:
When an SRH is processed {
   If (Segments Left == 0 and (DA.Arg.Index == 0 or Segment List[0][DA.Arg.Index-1] == 0)) {
      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.
   }
   If (IPv6 Hop Limit <= 1) {
      Send an ICMP Time Exceeded message to the Source Address,
      Code 0 (Hop limit exceeded in transit),
      interrupt packet processing and discard the packet.
   }
   max_LE = (Hdr Ext Len / 2) - 1
   If (DA.Arg.Index != 0) {
      If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
         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.
      }
      Decrement DA.Arg.Index by 1.
      If (Segment List[Segments Left][DA.Arg.Index] == 0) {
         Decrement Segments Left by 1.
         Decrement IPv6 Hop Limit by 1.
         Update IPv6 DA with Segment List[Segments Left]
      }
      If (Upper-layer header type != 143 (Ethernet)) {
         Resubmit the packet to the IPv6 module for
         transmission to the new destination.
      }
      Perform IPv6 decapsulation.
      Submit the frame to the Ethernet module for transmission
      via interface IFACE-OUT.
   }
   } Else {
      If((Last Entry > max.LE) or (Segments Left > Last Entry+1)){
         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.
      }
      Decrement Segments Left by 1.
      Set DA.Arg.Index to (128/LNFL - 1).
   }
   Decrement IPv6 Hop Limit by 1.
   Write Segment List[Segments Left][DA.Arg.Index] into the bits
   [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
   header.
   If (Upper-layer header type != 143 (Ethernet)) {
   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-C-SID 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-C-SID 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. }
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-C-SID 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-C-SID 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. }


R17.   Set DA.Arg.Index to (128/LNFL - 1).

R18. }


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-C-SID 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-C-SID 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. }
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-C-SID 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-C-SID 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.     }
R04.     Perform IPv6 decapsulation.
P05.     Submit the packet to the IPv6 module for transmission on interface IFACE-OUT via NH-ADDR.
P11.   }
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.     }
R17.     Set DA.Arg.Index to (128/LNFL - 1).
R18. }
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-C-SID 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-C-SID 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. }
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-C-SID 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-C-SID 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.
P11. }
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. }
R17.   Set DA.Arg.Index to (128/LNFL - 1).
R18. }
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-C-SID 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-C-SID 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. }
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-C-SID 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-C-SID 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.
P11.   }
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.    }
R17.    Set DA.Arg.Index to (128/LNFL - 1).
R18.    }
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-C-SID 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-C-SID flavor for IPv4 traffic:
If (DA.Argument != 0) {
    If (IPv6 Hop Limit <= 1) {
        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.
    } 
    Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)] of the Destination Address.
    Set the bits [(LBL+AL)..127] of the Destination Address to zero.
    Decrement IPv6 Hop Limit by 1.
    If (Upper-layer header type != 4 (IPv4)) {
        Resubmit the packet to the IPv6 module for transmission to the new destination.
    }
    Copy the IPv6 encapsulation in a CACHE entry associated with the interface IFACE-IN.
    Perform IPv6 decapsulation.
    Submit the frame to the IPv4 module for transmission via interface IFACE-OUT via NH-ADDR.
}

If (Segments Left == 0) {
    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.
}

If (IPv6 Hop Limit <= 1) {
    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.
}

max_LE = (Hdr Ext Len / 2) - 1
If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
    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.
}

Decrement IPv6 Hop Limit by 1.
Decrement Segments Left by 1.
Update IPv6 DA with Segment List[Segments Left].
If (Upper-layer header type != 4 (IPv4)) {
    Resubmit the packet to the IPv6 module for transmission to the new destination.
}
Copy the IPv6 encapsulation in a CACHE entry associated with the interface IFACE-IN.
Perform IPv6 decapsulation.
Q06. Submit the frame to the IPv4 module for transmission via interface IFACE-OUT via NH-ADDR.
A.11. REPLACE-C-SID 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-C-SID 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.   }
R07.   If (Segment List[Segments Left][DA.Arg.Index] == 0) {
R08.     Decrement Segments Left by 1.
R09.     Decrement IPv6 Hop Limit by 1.
R10.   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.
P07.   }
R11.   } Else {
R12.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
R13.       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.
R14.       }
R15.     Decrement Segments Left by 1.
R17.   }
R19.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
     [LBL..LBL+LNFL-1] of the Destination Address of the IPv6
     header.
If (Upper-layer header type != 41 (IPv6)) {
    Resubmit the packet to the IPv6 module for transmission to
    the new destination.
}

Copy the IPv6 encapsulation in a CACHE entry associated with
the interface INTERFACE-IN.

Perform IPv6 decapsulation.

Submit the frame to the IPv6 module for transmission via
interface INTERFACE-OUT via NH-ADDR.
A.12. NEXT-C-SID 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-C-SID 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. }
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-C-SID 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-C-SID flavor:
When an SRH is processed {
    If (Segments Left == 0 and (DA.Arg.Index == 0 or Segment List[0][DA.Arg.Index-1] == 0)) {
        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.
    }

    If (IPv6 Hop Limit <= 1) {
        Send an ICMP Time Exceeded message to the Source Address, Code 0 (Hop limit exceeded in transit), interrupt packet processing and discard the packet.
    }

    max_LE = (Hdr Ext Len / 2) - 1

    If (DA.Arg.Index != 0) {
        If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
            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.
        }

        Decrement DA.Arg.Index by 1.
        If (Segment List[Segments Left][DA.Arg.Index] == 0) {
            Decrement Segments Left by 1.
            Decrement IPv6 Hop Limit by 1.
            Update IPv6 DA with Segment List[Segments Left]
            Copy the IPv6 Destination Address in a CACHE entry associated with the interface IFACE-IN.
            Copy Segment List[0] from the SRH to the Destination Address of the IPv6 header.
            Submit the packet to the IPv6 module for transmission on interface IFACE-OUT via NH-ADDR.
        }
        Else {
            If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
                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.
            }
            Decrement Segments Left by 1.
            Set DA.Arg.Index to (128/LNFL - 1).
        }
    }

    Decrement IPv6 Hop Limit by 1.

    Copy the IPv6 Destination Address in a CACHE entry associated with the interface IFACE-IN.
    Copy Segment List[0] from the SRH to the Destination Address of the IPv6 header.
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-C-SID 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.   }
S13.   Submit the packet to the IPv6 module for transmission to the next destination.
S14. }

A.15. NEXT-C-SID 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-C-SID flavor:
If (DA.Argument != 0) {
    If (IPv6 Hop Limit <= 1) {
        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.

        Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)] of the Destination Address.

        Set the bits [(LBL+AL)..127] of the Destination Address to zero.

        Decrement IPv6 Hop Limit by 1.
        Copy the IPv6 Destination Address in a CACHE entry associated with the interface IFACE-IN.
        Copy Segment List[0] from the SRH to the Destination Address of the IPv6 header.
        Submit the packet to the IPv6 module for transmission on interface IFACE-OUT via NH-ADDR.
    }
}

If (Segments Left == 0) {
    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.
}

If (IPv6 Hop Limit <= 1) {
    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.

    max_LE = (Hdr Ext Len / 2) - 1
    If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
        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.
    }

    Decrement IPv6 Hop Limit by 1.
    Decrement Segments Left by 1.
    Update IPv6 DA with Segment List[Segments Left].
    Copy the IPv6 Destination Address in a CACHE entry associated with the interface IFACE-IN.
    Copy Segment List[0] from the SRH to the Destination Address of the IPv6 header.
    Submit the packet to the IPv6 module for transmission on interface IFACE-OUT via NH-ADDR.
A.16. NEXT-C-SID 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.   }
S13.   Submit the packet to the IPv6 module for transmission to the next destination.
S14. }

A.17. REPLACE-C-SID Flavor for Destination NAT Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the REPLACE-C-SID is executed:
A.18. NEXT-C-SID Flavor for Destination NAT Flavor

The Destination NAT flavor of the SRv6 masquerading proxy with the NEXT-C-SID 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.   }
S13.   Submit the packet to the IPv6 module for transmission to the next destination.
S14. }

The Destination NAT flavor of the SRv6 masquerading proxy with the NEXT-C-SID 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.   }
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