

SPRING
Internet-Draft
Intended status: Standards Track
Expires: 2 October 2023

W. Cheng, Ed.
China Mobile
C. Filsfils
Cisco Systems, Inc.
Z. Li
Huawei Technologies
B. Decraene
Orange
F. Clad, Ed.
Cisco Systems, Inc.
31 March 2023

**Compressed SRV6 Segment List Encoding in SRH
draft-ietf-spring-srv6-srh-compression-04**

Abstract

This document specifies new flavors for the SR endpoint behaviors defined in [RFC 8986](#), which enable a compressed SRV6 Segment-List encoding in the Segment Routing Header (SRH).

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 2 October 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights

and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	4
2.1.	Requirements Language	4
3.	Basic Concepts	4
4.	SR Endpoint Flavors	5
4.1.	NEXT-C-SID Flavor	6
4.1.1.	End with NEXT-C-SID	6
4.1.2.	End.X with NEXT-C-SID	7
4.1.3.	End.T with NEXT-C-SID	8
4.1.4.	End.B6.Encaps with NEXT-C-SID	8
4.1.5.	End.B6.Encaps.Red with NEXT-C-SID	9
4.1.6.	End.BM with NEXT-C-SID	9
4.1.7.	Combination with PSP, USP and USD flavors	9
4.2.	REPLACE-C-SID Flavor	9
4.2.1.	End with REPLACE-C-SID	10
4.2.2.	End.X with REPLACE-C-SID	12
4.2.3.	End.T with REPLACE-C-SID	12
4.2.4.	End.B6.Encaps with REPLACE-C-SID	12
4.2.5.	End.B6.Encaps.Red with REPLACE-C-SID	13
4.2.6.	End.BM with REPLACE-C-SID	13
4.2.7.	End.DX and End.DT with REPLACE-C-SID	13
4.2.8.	Combination with PSP, USP, and USD flavors	14
5.	C-SID Allocation	14
5.1.	Global C-SID	15
5.2.	Local C-SID	15
6.	C-SID and Locator-Block Length	15
6.1.	C-SID Length	15
6.2.	Locator-Block Length	16
6.3.	GIB/LIB Usage	16
6.4.	Recommended Installation of C-SIDs in FIB	17
7.	SR Source Node	18
7.1.	Rules for NEXT-C-SID flavor uncompressed SID lists	20
7.2.	Rules for REPLACE-C-SID flavor uncompressed SID lists	21
8.	Inter Routing Domains with the End.XPS behavior	21
9.	Control Plane	23
10.	Operational Considerations	24
10.1.	Ping a SID without a Segment List	24
10.2.	Ping a SID via a Segment List	24
10.3.	ICMP Error Processing	24
11.	Illustrations	25
12.	Deployment Model	25

13.	Implementation Status	25
13.1.	Cisco Systems	26
13.2.	Huawei Technologies	26
13.3.	Nokia	27
13.4.	Arrcus	28
13.5.	Marvell	28
13.6.	Broadcom	28
13.7.	ZTE Corporation	29
13.8.	New H3C Technologies	29
13.9.	Ruijie Network	30
13.10.	Open Source	30
13.11.	Interoperability Report	30
14.	Security Considerations	32
15.	IANA Considerations	32
15.1.	SRv6 Endpoint Behaviors	32
16.	Acknowledgements	35
17.	References	35
17.1.	Normative References	35
17.2.	Informative References	36
Appendix A.	Open Issues	39
Appendix B.	Complete pseudocodes	40
B.1.	End with NEXT-C-SID	40
B.2.	End with REPLACE-C-SID	42
	Contributors	44
	Authors' Addresses	45

1. Introduction

The Segment Routing (SR) architecture and SR for IPv6 (SRv6) are defined in [[RFC8402](#)].

SRv6 Network Programming [[RFC8986](#)] defines a framework to build a network program with topological and service segments (also referred to by their segment identifier (SID)) carried in a Segment Routing header (SRH) [[RFC8754](#)].

This document specifies new flavors to the SR endpoint behaviors defined in [Section 4 of \[\[RFC8986\]\(#\)\]](#). These flavors enable a compressed encoding of the SRv6 Segment-List in the SRH and therefore address the requirements described in [[I-D.srcompdt-spring-compression-requirement](#)].

The flavors defined in this document leverage the SRv6 data plane defined in [[RFC8754](#)] and [[RFC8986](#)], and are compatible with the SRv6 control plane extensions for IS-IS [[RFC9352](#)], OSPF [[I-D.ietf-lsr-ospfv3-srv6-extensions](#)], and BGP [[RFC9252](#)].

2. Terminology

This document leverages the terms defined in [\[RFC8402\]](#), [\[RFC8754\]](#), and [\[RFC8986\]](#). The reader is assumed to be familiar with this terminology.

This document introduces the following new terms:

- * Locator-Block: The SRv6 SID block (IPv6 prefix allocated for SRv6 SIDs by the operator) of an SRv6 SID Locator, as defined in [Section 3.1 of \[RFC8986\]](#).
- * Locator-Node: The identifier of the parent node instantiating a SID in an SRv6 SID Locator, as defined in [Section 3.1 of \[RFC8986\]](#).
- * Compressed-SID (C-SID): The Locator-Node and Function bits of a SID that supports compressed encoding of SIDs.
- * C-SID container: A 128-bit container holding a list of C-SIDs.
- * C-SID sequence: A group of one or more consecutive C-SID containers in a segment list.
- * Uncompressed SID sequence: A group of one or more uncompressed SIDs in a segment list.
- * Compressed Segment List encoding: A segment list encoding that reduces the packet header length thanks to one or more C-SID sequences. A compressed Segment List encoding may contain any number of uncompressed SID sequences.

2.1. Requirements Language

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

3. Basic Concepts

In an SRv6 domain, the SIDs are allocated from a particular IPv6 prefix: the Locator-Block. All SRv6 SIDs instantiated from the same Locator-Block share the same most significant bits.

When the combined length of the SRv6 SID Locator, Function, and Argument is smaller than 128 bits, the trailing bits are set to zero.

When a sequence of consecutive SIDs in a Segment List shares a common Locator-Block, a compressed Segment-List encoding can optimize the packet header length by avoiding the repetition of the Locator-Block and trailing bits with each individual SID.

The compressed Segment List encoding is fully compliant with the specifications in [\[RFC8402\]](#), [\[RFC8754\]](#), and [\[RFC8986\]](#). Efficient encoding is achieved by combining a compressed Segment List encoding logic on the SR policy headend with new flavors of the base SRv6 endpoint behaviors that decode this compressed encoding.

A Segment List can be encoded in the packet header using any combination of compressed and uncompressed sequences. The C-SID sequences leverage the flavors defined in this document, while the uncompressed sequences use behaviors and flavors defined in other documents, such as [\[RFC8986\]](#). An SR Policy headend constructs and compresses the SID-list depending on the capabilities of each SR endpoint node that the packet should traverse, as well as its own compression capabilities.

It is expected that compressed encoding flavors be available on devices with limited packet manipulation capabilities, such as legacy ASICs.

The compressed Segment List encoding supports any Locator-Block allocation. While other options are supported and may provide higher efficiency, each routing domain can be allocated a /48 prefix from a global IPv6 block (see [Section 6.2](#)).

4. SR Endpoint Flavors

This section defines several options to achieve compressed Segment List encoding in the form of two new flavors for the End, End.X, End.T, End.B6.Encaps, End.B6.Encaps.Red, and End.BM behaviors of [\[RFC8986\]](#). These flavors could also be combined with behaviors defined in other documents.

The compressed encoding can be achieved by leveraging any of these SR endpoint flavors. The NEXT-C-SID flavor and the REPLACE-C-SID flavor expose the same high-level behavior in their use of the SID argument to determine the next segment to be processed, but they have different low-level characteristics that can make one more or less efficient than the other for a particular SRv6 deployment.

It is RECOMMENDED, for ease of operation, that a single compressed encoding flavor be used in a given SRv6 domain. However, in a multi-domain deployment, different flavors can be used in different domains.

Both flavors leverage the following variables:

- * Variable LBL is the Locator-Block length of the SID.
- * Variable LNFL is the sum of the Locator-Node and the Function lengths of the SID. It is also referred to as C-SID length.
- * Variable AL is the Argument length of the SID.

4.1. NEXT-C-SID Flavor

A SID instantiated with the NEXT-C-SID flavor takes an argument that carries the remaining C-SIDs in the current C-SID container.

The length AL of the argument is equal to 128-LBL-LNFL.

An SR segment endpoint node instantiating a SID with the NEXT-C-SID flavor MUST accept any argument value for that SID.

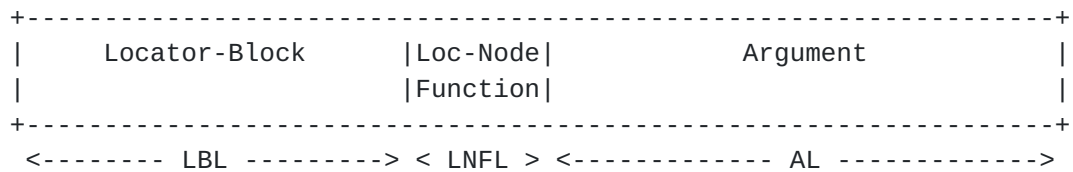


Figure 1: Example of a NEXT-C-SID flavored SID structure using a 48-bit Locator-Block, 16-bit combined locator and function, and 64-bit argument

4.1.1. End with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End SID with the NEXT-C-SID flavor, the procedure described in [Section 4.1 of \[RFC8986\]](#) is executed with the following modifications.

The below pseudocode is inserted between lines S01 and S02 of the SRH processing in [Section 4.1 of \[RFC8986\]](#), and a second time before line S01 of the upper-layer header processing in [Section 4.1.1 of \[RFC8986\]](#), or prior to processing any extension header other than Hop-by-Hop or Destination Option.


```
S01. If (DA.Argument != 0) {
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.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
        of the Destination Address.
S06.   Set the bits [(LBL+AL)..127] of the Destination Address to
        zero.
S07.   Decrement Hop Limit by 1.
S08.   Submit the packet to the egress IPv6 FIB lookup for
        transmission to the next destination.
S09. }
```

Notes:

- * DA.Argument identifies the bits [(LBL+LNFL)..127] in the Destination Address of the IPv6 header.
- * The value in the Segments Left field of the SRH is not modified when DA.Argument in the received packet has a non-zero value.

A rendering of the complete pseudocode is provided in [Appendix B.1](#).

4.1.2. End.X with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.X SID with the NEXT-C-SID flavor, the procedure described in [Section 4.2 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.1.1](#) of this document is modified by replacing line S08 as shown below.

```
S08.   Submit the packet to the IPv6 module for transmission to the
        new destination via a member of J.
```

The resulting pseudocode is inserted between lines S01 and S02 of the SRH processing in [Section 4.2 of \[RFC8986\]](#), and a second time before line S01 of the upper-layer header processing in [Section 4.1.1 of \[RFC8986\]](#), or prior to processing any extension header other than Hop-by-Hop or Destination Option.

4.1.3. End.T with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.T SID with the NEXT-C-SID flavor, the procedure described in [Section 4.3 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.1.1](#) of this document is modified by replacing line S08 as shown below.

- S08.1. Set the packet's associated FIB table to T.
- S08.2. Submit the packet to the egress IPv6 FIB lookup for transmission to the new destination.

The resulting pseudocode is inserted between lines S01 and S02 of the SRH processing in [Section 4.3 of \[RFC8986\]](#), and a second time before line S01 of the upper-layer header processing in [Section 4.1.1 of \[RFC8986\]](#), or prior to processing any extension header other than Hop-by-Hop or Destination Option.

4.1.4. End.B6.Encaps with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.B6.Encaps SID with the NEXT-C-SID flavor, the procedure described in [Section 4.13 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.1.1](#) of this document is modified by replacing line S08 as shown below.

- S08.1. Push a new IPv6 header with its own SRH containing B.
- S08.2. Set the outer IPv6 SA to A.
- S08.3. Set the outer IPv6 DA to the first SID of B.
- S08.4. Set the outer Payload Length, Traffic Class, Flow Label, Hop Limit, and Next Header fields.
- S08.5. Submit the packet to the egress IPv6 FIB lookup for transmission to the next destination.

The resulting pseudocode is inserted between lines S01 and S02 of the SRH processing in [Section 4.13 of \[RFC8986\]](#), and a second time before line S01 of the upper-layer header processing in [Section 4.1.1 of \[RFC8986\]](#), or prior to processing any extension header other than Hop-by-Hop or Destination Option.

4.1.5. End.B6.Encaps.Red with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.B6.Encaps.Red SID with the NEXT-C-SID flavor, the procedure described in [Section 4.14 of \[RFC8986\]](#) is executed with the same modifications as in [Section 4.1.4](#) of this document.

4.1.6. End.BM with NEXT-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.BM SID with the NEXT-C-SID flavor, the procedure described in [Section 4.15 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.1.1](#) of this document is modified by replacing line S08 as shown below.

- S08.1. Push the MPLS label stack for B.
- S08.2. Submit the packet to the MPLS engine for transmission.

The resulting pseudocode is inserted between lines S01 and S02 of the SRH processing in [Section 4.15 of \[RFC8986\]](#), and a second time before line S01 of the upper-layer header processing in [Section 4.1.1 of \[RFC8986\]](#), or prior to processing any extension header other than Hop-by-Hop or Destination Option.

4.1.7. Combination with PSP, USP and USD flavors

PSP: The PSP flavor defined in [Section 4.16.1 of \[RFC8986\]](#) is unchanged when combined with the NEXT-C-SID flavor.

USP: The USP flavor defined in [Section 4.16.2 of \[RFC8986\]](#) is unchanged when combined with the NEXT-C-SID flavor.

USD: The USD flavor is unchanged when combined with the NEXT-C-SID flavor. The pseudocodes defined in [Section 4.1.1](#), [Section 4.1.2](#), and [Section 4.1.3](#) of this document are inserted at the beginning of the modified upper-layer header processing defined in [Section 4.16.3 of \[RFC8986\]](#) for End, End.X, and End.T, respectively.

4.2. REPLACE-C-SID Flavor

A SID instantiated with the REPLACE-C-SID flavor takes an argument that indicates the index of the next C-SID in the appropriate C-SID container.

The length AL of the argument is equal to 128-LBL-LNFL. The index value is encoded in the least significant $\text{ceil}(\log_2(128/\text{LNFL}))$ bits of the argument field.

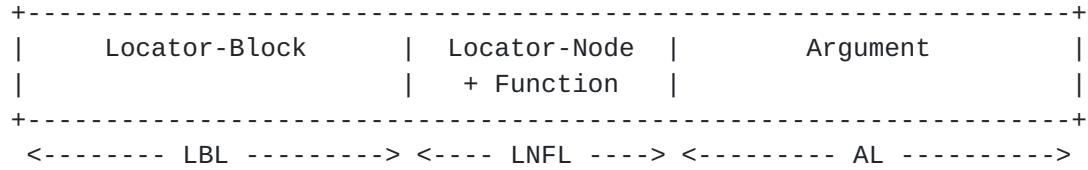


Figure 2: Example of a REPLACE-C-SID flavored SID structure using a 48-bit Locator-Block, 32-bit combined locator and function, and 48-bit argument

4.2.1. End with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End SID with the REPLACE-C-SID flavor, the SRH processing described in [Section 4.1 of \[RFC8986\]](#) is executed with the following modifications.

Line S02 of SRH processing in [Section 4.1 of \[RFC8986\]](#) is replaced as follows.

```

S02.  If (Segments Left == 0 and (DA.Arg.Index == 0 or
      Segment List[0][DA.Arg.Index-1] == 0)) {

```

Lines S09 to S16 are replaced by the following pseudo code.


```
S09.  If (DA.Arg.Index != 0) {
S10.    If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
S11.      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.
S12.    }
S13.    Decrement DA.Arg.Index by 1.
S14.    If (Segment List[Segments Left][DA.Arg.Index] == 0) {
S15.      Decrement Segments Left by 1.
S16.      Decrement IPv6 Hop Limit by 1.
S17.      Update IPv6 DA with Segment List[Segments Left]
S18.      Submit the packet to the egress IPv6 FIB lookup for
          transmission to the new destination.
S19.    }
S20.  } Else {
S21.    If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
S22.      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.
S23.    }
S24.    Decrement Segments Left by 1.
S25.    Set DA.Arg.Index to (128/NF - 1).
S26.  }
S27.  Decrement IPv6 Hop Limit by 1.
S28.  Write Segment List[Segments Left][DA.Arg.Index] into the bits
      [B..B+NF-1] of the Destination Address of the IPv6 header.
S29.  Submit the packet to the egress IPv6 FIB lookup for
      transmission to the new destination.
S30. }
```

Notes:

- * DA.Arg.Index identifies the bits [(128-ceil(log₂(128/LNFL))..127] in the Destination Address of the IPv6 header.
- * Segment List[Segments Left][DA.Arg.Index] identifies the bits [DA.Arg.Index*LNFL..(DA.Arg.Index+1)*LNFL-1] in the SRH Segment List entry at index Segments Left.

The upper-layer header processing described in [Section 4.1.1 of RFC8986](#) is unchanged.

A rendering of the complete pseudocode is provided in [Appendix B.2](#).

4.2.2. End.X with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.X SID with the REPLACE-C-SID flavor, the procedure described in [Section 4.2 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.2.1](#) of this document is modified by replacing line S18 and S29 as shown below.

S01. Submit the packet to the IPv6 module for transmission to the new destination via a member of J.

The SRH processing in [Section 4.2 of \[RFC8986\]](#) is replaced with the resulting pseudocode. The upper-layer header processing is unchanged.

4.2.3. End.T with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.T SID with the REPLACE-C-SID flavor, the procedure described in [Section 4.3 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.2.1](#) of this document is modified by replacing line S18 and S29 as shown below.

S01. Set the packet's associated FIB table to T.

S02. Submit the packet to the egress IPv6 FIB lookup for transmission to the new destination.

The SRH processing in [Section 4.3 of \[RFC8986\]](#) is replaced with the resulting pseudocode. The upper-layer header processing is unchanged.

4.2.4. End.B6.Encaps with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.B6.Encaps SID with the REPLACE-C-SID flavor, the procedure described in [Section 4.13 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.2.1](#) of this document is modified by replacing line S18 and S29 as shown below.

- S01. Push a new IPv6 header with its own SRH containing B.
- S02. Set the outer IPv6 SA to A.
- S03. Set the outer IPv6 DA to the first SID of B.
- S04. Set the outer Payload Length, Traffic Class, Flow Label, Hop Limit, and Next Header fields.
- S05. Submit the packet to the egress IPv6 FIB lookup for transmission to the next destination.

The SRH processing in [Section 4.13 of \[RFC8986\]](#) is replaced with the resulting pseudocode. The upper-layer header processing is unchanged.

[4.2.5.](#) End.B6.Encaps.Red with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.B6.Encaps.Red SID with the REPLACE-C-SID flavor, the procedure described in [Section 4.14 of \[RFC8986\]](#) is executed with the same modifications as in [Section 4.2.4](#) of this document.

[4.2.6.](#) End.BM with REPLACE-C-SID

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.BM SID with the REPLACE-C-SID flavor, the procedure described in [Section 4.15 of \[RFC8986\]](#) is executed with the following modifications.

The pseudocode in [Section 4.2.1](#) of this document is modified by replacing line S18 and S29 as shown below.

- S01. Push the MPLS label stack for B.
- S02. Submit the packet to the MPLS engine for transmission.

The SRH processing in [Section 4.15 of \[RFC8986\]](#) is replaced with the resulting pseudocode. The upper-layer header processing is unchanged.

[4.2.7.](#) End.DX and End.DT with REPLACE-C-SID

New behaviors of End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DX2U, or End.DX2M [\[RFC8986\]](#) with REPLACE-C-SID flavor are also defined in this draft. These new behaviors can be used to indicate the capability of compression of Node and SID, which are required in path computation and compressed SID list encoding.

As per [Section 6.4](#), when allocating a C-SID value from a Local Identifiers Block (LIB), an extra prefix of Locator_block:FunctionID::/LBL+FL is required on the Segment Endpoint

node to support LIB matching in forwarding. The new behaviors with REPLACE-C-SID flavor explicitly require the node to do so in SID initiation.

When processing an IPv6 packet that matches a FIB entry locally instantiated as an End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DT2U or End.DT2M SID with the REPLACE-C-SID flavor, the procedures described in [RFC8986] are executed. For End.DT2M with REPLACE-C-SID flavor, the Arg.FE2 is a 16-bit value located in the least significant $[(128 - \text{ceil}(\log_2(128 / \text{LNFL})) - 16) .. (128 - \text{ceil}(\log_2(128 / \text{LNFL})) - 1)]$ bits of the argument.

4.2.8. Combination with PSP, USP, and USD flavors

PSP: When combined with the REPLACE-C-SID flavor, the additional PSP flavor instructions defined in [Section 4.16.1.2 of \[RFC8986\]](#) are inserted after line S17 and S28 of the pseudocode in [Section 4.2.1](#), and the first line of the inserted instructions after S28 is modified as follows.

```
S28.1.  If (Segments Left == 0 and (DA.Arg.Index == 0 or
        Segment List[0][DA.Arg.Index-1] == 0)) {
```

Note:

* Segment List[Segments Left][DA.Arg.Index-1] identifies the bits $[(\text{DA.Arg.Index}-1) * \text{LNFL} .. \text{DA.Arg.Index} * \text{LNFL} - 1]$ in the SRH Segment List entry at index Segments Left.

USP: When combined with the REPLACE-C-SID flavor, the line S03 of the pseudocode in [Section 4.2.1](#) are substituted by the USP flavor instructions S03.1 to S03.4 defined in [Section 4.16.2 of \[RFC8986\]](#). Note that S03 is shown in the complete pseudocode in [Appendix B.2](#).

USD: The USD flavor defined in [Section 4.16.3 of \[RFC8986\]](#) is unchanged when combined with the REPLACE-C-SID flavor.

5. C-SID Allocation

The C-SID value of 0 is RESERVED. It is used to indicate the end of a C-SID container.

In order to efficiently manage the C-SID numbering space, it may be beneficial to divide it into two non-overlapping sub-spaces: a Global Identifiers Block (GIB) and a Local Identifiers Block (LIB).

* The GIB is the pool of C-SID values available for global allocation.

- * The LIB is the pool of C-SID values available for local allocation.

The concept of LIB is applicable to SRv6 and specifically to its NEXT-C-SID and REPLACE-C-SID flavors. The shorter the C-SID, the more benefit the LIB brings.

The opportunity to use these sup-spaces, their size, and their C-SID allocation policy depends on the C-SID length relative to the size of the network (e.g., number of nodes, links, service routes). Some guidelines for a typical deployment scenario are provided in [Section 6.3](#).

5.1. Global C-SID

A C-SID from the GIB.

A Global C-SID typically identifies a shortest path to a node in the SRv6 domain. An IP route is advertised by the parent node to each of its global C-SIDs, under the associated Locator-Block. The parent node executes a variant of the End behavior.

A node can have multiple global C-SIDs under the same Locator-Block (e.g., one per IGP flexible algorithm). Multiple nodes may share the same global C-SID (anycast).

5.2. Local C-SID

A C-SID from the LIB.

A local C-SID may identify a cross-connect to a direct neighbor over a specific interface or a VPN context.

No IP route is advertised by a parent node for its local C-SIDs.

If N1 and N2 are two different physical nodes of the SRv6 domain and I is a local C-SID value, then N1 and N2 may bind two different behaviors to I.

6. C-SID and Locator-Block Length

6.1. C-SID Length

The NEXT-C-SID flavor supports both 16- and 32-bit C-SID lengths. A C-SID length of 16-bit is RECOMMENDED.

The REPLACE-C-SID flavor supports both 16- and 32-bit C-SID lengths. A C-SID length of 32-bit is RECOMMENDED.

6.2. Locator-Block Length

The RECOMMENDED Locator-Block sizes for the NEXT-C-SID flavor are 16, 32, or 48 bits. The smaller the block, the higher the compression efficiency.

The RECOMMENDED Locator-Block size for the REPLACE-C-SID flavor can be 48, 56, 64, 72, or 80 bits, depending on the needs of the operator.

6.3. GIB/LIB Usage

GIB and LIB usage is a local implementation and/or configuration decision, however, some guidelines for determining usage for specific SID behaviors and recommendations are provided.

The GIB number space is shared among all segment endpoint nodes using SRv6 locators under a Block space. The more SIDs assigned from this space, per node, the faster it is exhausted. Therefore its use is prioritized for SIDs that identify a node, like End behavior SIDs.

The LIB number space is unique per node. Each node is able to fully utilize the entire LIB number space without consideration of assignments at other nodes. Therefore its use is prioritized for SIDs that identify services (of which there may be many) at nodes, like cross-connects, adjacencies, etc.

While a longer C-SID length permits more flexibility in which SID behaviors may be assigned from the GIB, it also reduces compression.

Given the previous Locator-Block and C-SID length recommendations, the following GIB/LIB usage is RECOMMENDED:

* NEXT-C-SID:

- GIB: End
- LIB: End.X, End.T, End.DT4/6/46/2U/2M, End.DX4/6/2/2V (including large-scale pseudowire), End.B6.Encaps, End.B6.Encaps.Red, End.BM

* REPLACE-C-SID:

- GIB: End, End.X, End.T, End.DT4/6/46/2U/2M, End.DX4/6/2/2V, End.B6.Encaps, End.B6.Encaps.Red, End.BM
- LIB: End.DX2/2V for large-scale pseudowire

6.4. Recommended Installation of C-SIDs in FIB

An SR segment endpoint node instantiating a NEXT-C-SID or REPLACE-C-SID flavored SID should install the corresponding FIB entry to match only the locator and function parts of the SID (i.e., with a prefix length of LBL + LNL + FL).

In addition, an SR segment endpoint node instantiating NEXT-C-SID flavored SIDs from both GIB and LIB may install combined "Global + Local" FIB entries to match a sequence of global and local C-SIDs in a single LPM lookup.

For example, let us consider an SR segment endpoint node 10 instantiating the following two NEXT-C-SID flavored SIDs according to the C-SID length, Locator-Block length, and GIB/LIB recommendations in this section.

- * 2001:db8:b1:10:: from GIB with Locator-Block length (LBL) = 48, Locator-Node length (LNL) = 16, Function length (FL) = 0, Argument length (AL) = 64, and bound to the End behavior with the NEXT-C-SID flavor.
- * 2001:db8:b1:f123:: from LIB with Locator-Block length (LBL) = 48, Locator-Node length (LNL) = 0, Function length (FL) = 16, Argument length (AL) = 64, and bound to the End.X behavior for its local IGP adjacency 123 with the NEXT-C-SID flavor.

For SID 2001:db8:b1:10::, Node 10 would install the FIB entry 2001:db8:b1:10::/64 bound the End SID with the NEXT-C-SID flavor.

For SID 2001:db8:b1:f123::, Node 10 would install the FIB entry 2001:db8:b1:f123::/64 bound the End.X SID for adjacency 123 with the NEXT-C-SID flavor.

In addition, Node 10 may also install the combined FIB entry 2001:db8:b1:10:f123::/80 bound the End.X SID for adjacency 123 with the NEXT-C-SID flavor.

As another example, let us consider an SR segment endpoint node 20 instantiating the following two REPLACE-C-SID flavored SIDs according to the C-SID length, Locator-Block length, and GIB/LIB recommendations in this section.

- * 2001:db8:b2:20:1:: from GIB with Locator-Block length (LBL) = 48, Locator-Node length (LNL) = 16, Function length (FL) = 16, Argument length (AL) = 48, and bound to the End behavior with the REPLACE-C-SID flavor.

- * 2001:db8:b2:20:123:: from GIB with Locator-Block length (LBL) = 48, Locator-Node length (LNL) = 16, Function length (FL) = 16, Argument length (AL) = 48, and bound to the End.X behavior for its local IGP adjacency 123 with the REPLACE-C-SID flavor.

For SID 2001:db8:b2:20:1::, Node 20 would install the FIB entry 2001:db8:b2:20:1::/80 bound the End SID with the REPLACE-C-SID flavor.

For SID 2001:db8:b2:20:123::, Node 20 would install the FIB entry 2001:db8:b2:20:123::/80 bound the End.X SID for adjacency 123 with the REPLACE-C-SID flavor.

7. SR Source Node

An SR source node MAY compress a segment list when it includes NEXT-C-SID or REPLACE-C-SID flavor SIDs.

If an SR source node chooses to compress the segment list, one method is described below for illustrative purposes. Any other method, resulting in a path equivalent to the original segment list, is compliant.

For a series of NEXT-C-SID flavor SIDs in the uncompressed segment list, an SR source node compresses an uncompressed segment list to a compressed segment list as follows.

1. Initialize a C-SID container as the first segment in the uncompressed segment list, with all argument bits set to 0, and the remaining capacity to AL
2. For each subsequent segment in the uncompressed segment list
 1. If the current segment Locator-Block matches that of the C-SID container and the current segment LNFL is lower than or equal to the remaining capacity
 1. Copy the current segment Locator-Node and Function to the most significant remaining argument bits of the C-SID container, and decrement the remaining capacity by LNFL
 2. Else
 1. Push the C-SID container onto the compressed segment list
 2. Initialize a C-SID container as the current segment in the uncompressed segment list, with all argument bits set to 0, and the remaining capacity to AL

3. Push the C-SID container onto the compressed segment list

For a series of REPLACE-C-SID flavor SIDs in the uncompressed segment list, the first segment is not compressed. The SR source then initializes an empty C-SID container with all bits set to 0 and inserts consecutive C-SIDs from the uncompressed segment list into the container. When the container is full, it is pushed on the compressed segment list and a new container is initialized.

Regardless of how an SR Policy compressed SID list is produced, it is encoded in the IPv6 header and optional SRH as described in [Section 4.1](#) and 4.1.1 of [[RFC8754](#)]. The text is reproduced below for reference.

A source node steers a packet into an SR Policy. If the SR Policy results in a Segment List containing a single segment, and there is no need to add information to the SRH flag or add TLV; the DA is set to the single Segment List entry, and the SRH MAY be omitted.

When needed, the SRH is created as follows:

The Next Header and Hdr Ext Len fields are set as specified in [\[RFC8200\]](#).

The Routing Type field is set to 4.

The DA of the packet is set with the value of the first segment.

The first element of the SRH Segment List is the ultimate segment. The second element is the penultimate segment, and so on.

The Segments Left field is set to $n-1$, where n is the number of elements in the SR Policy.

The Last Entry field is set to $n-1$, where n is the number of elements in the SR Policy.

TLVs (including HMAC) may be set according to their specification.

The packet is forwarded toward the packet's Destination Address (the first segment).

When a source does not require the entire SID list to be preserved in the SRH, a reduced SRH may be used.

A reduced SRH does not contain the first segment of the related SR Policy (the first segment is the one already in the DA of the IPv6 header), and the Last Entry field is set to $n-2$, where n is the number of elements in the SR Policy.

7.1. Rules for NEXT-C-SID flavor uncompressed SID lists

1. If a Destination Option header would follow an SRH with a segment list of more than one segment compressed as a single NEXT-C-SID container, the SR source MUST NOT omit the SRH.

2. When the Segment List entry in the SRH is a C-SID container representing more than one segment, the PSP operation is performed at the segment preceding this C-SID container in the segment list. If the PSP behavior should instead be performed at the penultimate segment along the path, the SR source MUST NOT compress the ultimate segment of the segment list into a C-SID container.
3. If a Destination Option header would follow an SRH with a last Segment List entry being a NEXT-C-SID container representing more than one segment, the SR source node MUST ensure that the PSP operation is not performed before the penultimate segment endpoint node along the path.

7.2. Rules for REPLACE-C-SID flavor uncompressed SID lists

1. All REPLACE-C-SID flavor SIDs in a series MUST share the same Locator-Block length (LBL) and the same combined Locator-Node and Function length (LNFL). An uncompressed SID list may contain any number of such series.
2. A series of REPLACE-C-SID flavor SIDs MUST contain at least two elements.
3. End of a C-SID sequence can be indicated by:
 - * a SID without REPLACE-C-SID flavor (e.g., End, End.X or NEXT-C-SID flavor End, End.X SID) regardless of the index of the C-SID in the container. The SID MUST have the same Locator-Block and LNFL length as the preceding REPLACE-C-SID flavor SID.
 - * a SID with REPLACE-C-SID flavor when the index of the C-SID is greater than 0.

The length of a C-SID is determined by its behavior and LNFL.

8. Inter Routing Domains with the End.XPS behavior

The End.XPS behavior described in this section is OPTIONAL.

Some SRv6 traffic may need to cross multiple routing domains, such as different Autonomous Systems (ASes) or different routing areas. Different routing domains may use different addressing schema and Locator-Blocks.

This section defines an optional solution and SID behavior allowing for the use of different Locator-Blocks between routing domains.

The solution requires a new SID behavior, called "Endpoint with cross-connect to an array of layer-3 adjacencies and SRV6 Prefix Swap" (End.XPS for short) allowing for this transition of Locator-Block between two routing domains.

End.XPS is a variant of End.X, performing both "End.X Layer-3 Cross-Connect" and the translation of the Locator-Block between the two routing domains.

The processing takes as an additional parameter the prefix B2/m corresponding the Locator-Block in the second domain. This parameter is a property of the (received) SID and is given as a result of the lookup on the IPv6 destination address which identifies the SRV6 SID and its properties.

The End.XPS behavior is compatible with the NEXT-C-SID and REPLACE-C-SID flavors described in this document.

When a router R receives a packet whose IPv6 DA matches a local End.XPS SID with the NEXT-C-SID flavor, that is associated with a set J of one or more Layer-3 adjacencies and the Locator-Block B2/m of the neighbor routing domain, R processes the packet as follows.

1. If (DA.Argument != 0) {
2. Write B2 into the most significant bits of the Destination Address of the IPv6 header.
3. Write DA.Argument into the bits [m..(m+AL-1)] of the Destination Address of the IPv6 header.
4. Set the bits [(m+AL)..127] of the Destination Address of the IPv6 header to zero.
5. } Else {
6. Decrement Segments Left by 1.
7. Copy Segment List[Segments Left] from the SRH to the Destination Address of the IPv6 header.
8. }
9. Submit the packet to the IPv6 module for transmission to the new destination via a member of J.

When a router R receives a packet whose IPv6 DA matches a local End.XPS SID with the REPLACE-C-SID flavor, that is associated with a set J of one or more Layer-3 adjacencies and the Locator-Block B2/m of the neighbor routing domain, R processes the packet as follows.

1. If (DA.Arg.Index != 0) {
2. Decrement DA.Arg.Index by 1.
3. } Else {
4. Decrement Segments Left by 1.
5. Set DA.Arg.Index to (128/LNFL - 1).
6. }
7. Write B2 into the most significant bits of the Destination Address of the IPv6 header.
8. Write Segment List[Segments Left][DA.Arg.Index] into the bits [m..m+LNFL-1] of the Destination Address of the IPv6 header.
9. Write DA.Arg.Index into the bits [m+LNFL..m+LNFL+AL-1] of the Destination Address of the IPv6 header.
10. Set the bits [(m+LNFL+AL)..127] of the Destination Address of the IPv6 header to zero.
11. Submit the packet to the IPv6 module for transmission to the new destination via a member of J.

Note: the way the Locator-Block B2 of the next routing domain is known is out of scope of this document. As examples, it could be learnt via configuration, or using a signaling protocol either with the peer domain or with a central controller (e.g. Path Computation Element (PCE)).

When End.XPS SID behavior is used, the restriction on the C-SID length for the REPLACE-C-SID flavor is relaxed and becomes: all SID the are part of a C-SID sequence *within a domain* MUST have the same C-SID length LNFL.

9. Control Plane

This document does not require any new extensions to routing protocols.

[Section 8 of \[RFC8986\]](#) provides an overview of the control plane protocols used for signaling of the SRv6 SIDs introduced by that document. The SRv6 SIDs introduced by this document are advertised using the same SRv6 extensions for various routing protocols, such as

- * IS-IS [[RFC9352](#)]
- * OSPFv3 [[I-D.ietf-lsr-ospfv3-srv6-extensions](#)]
- * BGP [[RFC9252](#)], [[I-D.ietf-idr-bgpls-srv6-ext](#)], [[I-D.ietf-idr-segment-routing-te-policy](#)], [[I-D.ietf-idr-bgp-ls-sr-policy](#)]
- * PCEP [[I-D.ietf-pce-segment-routing-ipv6](#)]

Signaling the SRv6 SID Structure is REQUIRED for all the SIDs introduced in this document. It is used by an SR source node to compress a segment list as described in [Section 7](#).

For a segment list computed by a controller and signaled to an SR source node (e.g., via BGP [[I-D.ietf-idr-segment-routing-te-policy](#)] or PCEP [[I-D.ietf-pce-segment-routing-ipv6](#)]), the controller provides the ordered segment list comprising the individual SIDs (i.e., without any compression) to the SR source node. The SR source node may then compress the segment list as described in [Section 7](#).

[10. Operational Considerations](#)

[10.1. Ping a SID without a Segment List](#)

An SR source node may ping a routable SRv6 SID by sending an ICMPv6 echo request packet destined to the SRv6 SID, as illustrated in [Appendix A.1.2 of \[RFC9259\]](#).

When the SRv6 SID in the destination address of the ICMPv6 echo request is one of the SID flavors defined in this document, the SR source node MUST set the arguments of the SID to 0.

[10.2. Ping a SID via a Segment List](#)

An SR source node may ping a routable or non-routable SRv6 SID via a segment list as illustrated in [Appendix A.1.2 of \[RFC9259\]](#).

Regardless of the behavior of the SIDs in the SID list, the SR source node computes the ICMP echo request checksum using the ultimate segment in the segment list, i.e., the IPv6 destination address as it is expected to appear at the ultimate destination of the packet.

[10.3. ICMP Error Processing](#)

When an IPv6 node encounters an error while processing a packet, it may report that error by sending an IPv6 error message to the packet source with an enclosed copy of the invoking packet. For the source of an invoking packet to process the ICMP error message, the ultimate destination address of the IPv6 header may be required.

[Section 5.4 of \[RFC8754\]](#) defines the logic that an SR source node should follow to determine the ultimate destination of an invoking packet containing an SRH.

For an SR source node that supports the compressed segment list encoding defined in this document, the logic to determine the ultimate destination is generalized as follows.

- * If the destination address of the invoking IPv6 packet matches a known SRv6 SID, modify the invoking IPv6 packet by applying the SID behavior associated with the matched SRv6 SID;
- * Repeat until the application of the SID behavior would result in the processing of the upper-layer header.

The destination address of the resulting IPv6 packet may be used as the ultimate destination of the invoking IPv6 packet.

Since the SR source node that needs to determine the ultimate destination is the same node that originally built the segment list in the invoking packet, it is able to perform this operation for all the SIDs in the packet.

11. Illustrations

Illustrations for the functionalities defined in this document are provided in [[I-D.clad-spring-srv6-srh-compression-illus](#)].

12. Deployment Model

[Section 5 of \[RFC8754\]](#) defines the intra-SR-domain deployment model and associated security procedures.

The same deployment model applies to the SIDs defined in this document.

13. Implementation Status

This section is to be removed before publishing as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [[RFC7942](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [[RFC7942](#)], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation

and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

This section is provided in compliance with the SPRING working group policies ([\[SPRING-WG-POLICIES\]](#)).

13.1. Cisco Systems

Cisco Systems reported the following implementations of the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)) and the SR source node efficient SID-list encoding ([Section 7](#)) for NEXT-C-SID flavored SIDs. These are used as part of its SRV6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities.

- * Cisco NCS 540 Series routers running IOS XR 7.3.x or above
[\[IMPL-CISCO-NCS540\]](#)
- * Cisco NCS 560 Series routers running IOS XR 7.6.x or above
[\[IMPL-CISCO-NCS560\]](#)
- * Cisco NCS 5500 Series routers running IOS XR 7.3.x or above
[\[IMPL-CISCO-NCS5500\]](#)
- * Cisco NCS 5700 Series routers running IOS XR 7.5.x or above
[\[IMPL-CISCO-NCS5700\]](#)
- * Cisco 8000 Series routers running IOS XR 7.5.x or above
[\[IMPL-CISCO-8000\]](#)
- * Cisco ASR 9000 Series routers running IOS XR 7.5.x or above
[\[IMPL-CISCO-ASR9000\]](#)

At the time of this report, all the implementations listed above are in production and follow the specification in the latest version of this document, including all the "MUST" and "SHOULD" clauses for the NEXT-C-SID flavor.

This report was last updated on January 11, 2023.

13.2. Huawei Technologies

Huawei Technologies reported the following implementations of the SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs. These are used as part of its SRV6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities.

- * Huawei ATN8XX, ATN910C, ATN980B routers running VRPV800R021C00 or above.
- * Huawei CX600-M2 routers running VRPV800R021C00 or above.
- * Huawei NE40E, ME60-X1X2, ME60-X3X8X16 routers running VRPV800R021C00 or above.
- * Huawei NE5000E, NE9000 routers running VRPV800R021C00 or above.
- * Huawei NCE-IP Controller running V1R21C00 or above.

At the time of this report, all the implementations listed above are in production and follow the specification in the latest version of this document, including all the "MUST" and "SHOULD" clauses for the REPLACE-C-SID flavor.

This report was last updated on January 11, 2023.

13.3. Nokia

Nokia reported the following implementations ([\[IMPL-NOKIA-20.10\]](#)) of the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)) for NEXT-C-SID flavored SIDs. These are used as part of its shortest path forwarding (in algorithm 0 and Flex-Algo), remote and TI-LFA repair tunnel, and Traffic Engineering functionalities.

- * Nokia 7950 XRS 20/20e routers running SROS Release 22.10 or above
- * Nokia 7750 SR-12e routers running SROS Release 22.10 or above
- * Nokia 7750 SR-7/12 routers running SROS Release 22.10 or above
- * Nokia 7750 SR-7s/14s routers running SROS Release 22.10 or above
- * Nokia 7750 SR-1/1s/2s routers running SROS Release 22.10 or above

At the time of this report, all the implementations listed above are in production and follow the specification in the latest version of this document, including all the "MUST" and "SHOULD" clauses for the NEXT-C-SID flavor.

This report was last updated on February 3, 2023.

13.4. Arrcus

Arrcus reported the following implementations of the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)) for NEXT-C-SID flavoured SIDs. These are used as part of its SRv6 shortest path forwarding (in algorithm 0 and Flex-Algo), TI-LFA, micro-loop avoidance and Traffic Engineering functionalities.

- * Arrcus running on Ufi Space routers S9510-28DC, S9710-76D, S9600-30DX and S9700-23D with ArcOS v5.2.1 or above
- * Arrcus running n Ufi Space routers S9600-72XC and S9700-53DX with ArcOS v5.1.1D or above
- * Arrcus running on Quanta router IXA and IXAE with ArcOS v5.1.1D or above

At the time of this report, all the implementations listed above are in production and follow the specification in the latest version of this document, including all the "MUST" and "SHOULD" clauses for the NEXT-C-SID flavor.

This report was last updated on March 11, 2023.

13.5. Marvell

Marvell reported support in the Marvell Prestera Packet Processor for the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)) for NEXT-C-SID flavored SIDs and SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs.

This report was last updated on February 15, 2023.

13.6. Broadcom

Broadcom reported the following implementations of the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)) for NEXT-C-SID flavored SIDs and SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs. These are used as part of its SRv6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities. All implementation of the following list is in general availability for customers using BCM SDK 6.5.26 or above.

- * 88850 (Jericho2c+) series
- * 88690 (Jericho2) series
- * 88800 (Jericho2c) series

- * 88480 (Qunran2a) series
- * 88280 (Qunran2u) series
- * 88295 (Qunran2n) series
- * 88830 (Jericho2x) series

At the time of this report, all the implementations listed above are in production and follow the specification in the latest version of this document, including all the "MUST" and "SHOULD" clauses for the NEXT-C-SID and REPLACE-C-SID flavors.

For 78900 (Tomahawk) series-related support, please contact the Broadcom team.

This report was last updated on February 21, 2023.

13.7. ZTE Corporation

ZTE Corporation reported the following implementations of the SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs. These are used as part of its SRv6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities.

- * ZTE M6000-18S(BRAS), M6000-8S Plus(BRAS) routers running V5.00.10.09 or above.
- * ZTE M6000-18S(SR), M6000-8S Plus(SR) routers running V5.00.10.80 or above.
- * ZTE T8000-18 routers running V5.00.10.07 or above.

This report was last updated on March 29, 2023.

13.8. New H3C Technologies

New H3C Technologies reported the following implementations of the SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs. These are used as part of its SRv6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities.

- * H3C CR16000-F, SR8800-X routers running Version 7.1.075 or above.
- * H3C CR18000, CR19000 routers running Version 7.1.071 or above.

This report was last updated on March 29, 2023.

13.9. Ruijie Network

Ruijie Network reported the following implementations of the SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)) for REPLACE-C-SID flavored SIDs. These are used as part of its SRV6 TI-LFA, micro-loop avoidance, and traffic engineering functionalities.

- * RUIJIE RG-N8018-R, RG-N8010-R routers running N8000-R_RGOS 12.8(3)B0801 or above.

This report was last updated on March 29, 2023.

13.10. Open Source

The authors found the following open source implementations of the SR endpoint node NEXT-C-SID flavor ([Section 4.1](#)).

- * The Linux kernel, version 6.1 [[IMPL-OSS-LINUX](#)]
- * The Software for Open Networking in the Cloud (SONiC), version 202212 [[IMPL-OSS-SONIC](#)], and Switch Abstraction Interface (SAI), version 1.9.0 [[IMPL-OSS-SAI](#)]
- * The Vector Packet Processor (VPP), version 20.05 [[IMPL-OSS-VPP](#)]
- * A generic P4 implementation [[IMPL-OSS-P4](#)]

The authors found the following open source implementations of the SR endpoint node REPLACE-C-SID flavor ([Section 4.2](#)).

- * ONOS and P4 Programmable Switch based [[IMPL-OSS-ONOS](#)]
- * Open SRV6 Project [[IMPL-OSS-OPEN-SRV6](#)]

This section was last updated on January 11, 2023.

13.11. Interoperability Report

In November 2020, China Mobile successfully validated multiple interoperable implementations of the NEXT-C-SID and REPLACE-C-SID flavors defined in this document.

This testing covered two different implementations of the SRV6 endpoint flavors defined in this document:

- * Hardware implementation in Cisco ASR 9000 running IOS XR
- * Software implementation in Cisco IOS XRv9000 virtual appliance

- * Hardware implementation in Huawei NE40E and NE5000E running VRP

The interoperability testing consisted of a packet flow sent by an SR source node N0 via an SR traffic engineering policy with a segment list <S1, S2, S3, S4, S5, S6, S7>, where S1..S7 are SIDs instantiated on SR segment endpoint nodes N1..N7, respectively.

```
N0 --- N1 --- N2 --- N3 --- N4 --- N5 --- N6 --- N7
      (S1)  (S2)  (S3)  (S4)  (S5)  (S6)  (S7)
```

- * N0 is a generic packet generator.
- * N1, N2, and N3 are Huawei routers.
- * N4, N5, and N6 are Cisco routers.
- * N7 is a generic traffic generator acting as a packet receiver.

The SR source node N0 steers the packets onto the SR policy by setting the IPv6 destination address and creating an SRH (as described in [Section 4.1 of \[RFC8754\]](#)) using a compressed segment list encoding. The length of the compressed segment list encoding varies for each scenario.

All SR segment endpoint nodes execute a variant of the End behavior: regular End behavior (as defined in [Section 4.1 of \[RFC8986\]](#)), End behavior with NEXT-C-SID flavor, and End behavior with REPLACE-C-SID flavor. The variant being used at each segment endpoint varies for each scenario.

The interoperability was validated for the following scenarios:

Scenario 1:

- * S1 and S2 are associated with the End behavior with the REPLACE-C-SID flavor
- * S3 is associated with the regular End behavior (no flavor)
- * S4, S5, and S6 are associated with the End behavior with the NEXT-C-SID flavor
- * The SR source node imposes a compressed segment list encoding of 3 SIDs.

Scenario 2:

- * S1, S2..., S6 are associated with the End behavior with the NEXT-C-SID flavor
- * The SR source node imposes a compressed segment list encoding of 2 SIDs.
- *Scenario 3:*
- * S1, S2..., S6 are associated with the End behavior with the REPLACE-C-SID flavor
- * The SR source node imposes a compressed segment list encoding of 3 SIDs.

14. 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.

15. IANA Considerations

15.1. SRv6 Endpoint Behaviors

This I-D. requests IANA to make the following registrations from the "SRv6 Endpoint Behaviors" sub-registry under the top-level "Segment Routing" registry (<https://www.iana.org/assignments/segment-routing/>):

Value	Description	Reference
43	End with NEXT-CSID	This I-D.
44	End with NEXT-CSID & PSP	This I-D.
45	End with NEXT-CSID & USP	This I-D.
46	End with NEXT-CSID, PSP & USP	This I-D.
47	End with NEXT-CSID & USD	This I-D.
48	End with NEXT-CSID, PSP & USD	This I-D.
49	End with NEXT-CSID, USP & USD	This I-D.
50	End with NEXT-CSID, PSP, USP & USD	This I-D.

52	End.X with NEXT-CSID	This I-D.
53	End.X with NEXT-CSID & PSP	This I-D.
54	End.X with NEXT-CSID & USP	This I-D.
55	End.X with NEXT-CSID, PSP & USP	This I-D.
56	End.X with NEXT-CSID & USD	This I-D.
57	End.X with NEXT-CSID, PSP & USD	This I-D.
58	End.X with NEXT-CSID, USP & USD	This I-D.
59	End.X with NEXT-CSID, PSP, USP & USD	This I-D.
TBA	End.T with NEXT-CSID	This I-D.
TBA	End.T with NEXT-CSID & PSP	This I-D.
TBA	End.T with NEXT-CSID & USP	This I-D.
TBA	End.T with NEXT-CSID, PSP & USP	This I-D.
TBA	End.T with NEXT-CSID & USD	This I-D.
TBA	End.T with NEXT-CSID, PSP & USD	This I-D.
TBA	End.T with NEXT-CSID, USP & USD	This I-D.
TBA	End.T with NEXT-CSID, PSP, USP & USD	This I-D.
TBA	End.B6.Encaps with NEXT-CSID	This I-D.
TBA	End.B6.Encaps.Red with NEXT-CSID	This I-D.
TBA	End.BM with NEXT-CSID	This I-D.
101	End with REPLACE-CSID	This I-D.
102	End with REPLACE-CSID & PSP	This I-D.
103	End with REPLACE-CSID & USP	This I-D.
104	End with REPLACE-CSID, PSP & USP	This I-D.
105	End.X with REPLACE-CSID	This I-D.

106	End.X with REPLACE-CSID & PSP	This I-D.
107	End.X with REPLACE-CSID & USP	This I-D.
108	End.X with REPLACE-CSID, PSP & USP	This I-D.
109	End.T with REPLACE-CSID	This I-D.
110	End.T with REPLACE-CSID & PSP	This I-D.
111	End.T with REPLACE-CSID & USP	This I-D.
112	End.T with REPLACE-CSID, PSP & USP	This I-D.
114	End.B6.Encaps with REPLACE-CSID	This I-D.
115	End.BM with REPLACE-CSID	This I-D.
116	End.DX6 with REPLACE-CSID	This I-D.
117	End.DX4 with REPLACE-CSID	This I-D.
118	End.DT6 with REPLACE-CSID	This I-D.
119	End.DT4 with REPLACE-CSID	This I-D.
120	End.DT46 with REPLACE-CSID	This I-D.
121	End.DX2 with REPLACE-CSID	This I-D.
122	End.DX2V with REPLACE-CSID	This I-D.
123	End.DX2U with REPLACE-CSID	This I-D.
124	End.DT2M with REPLACE-CSID	This I-D.
127	End.B6.Encaps.Red with REPLACE-CSID	This I-D.
128	End with REPLACE-CSID & USD	This I-D.
129	End with REPLACE-CSID, PSP & USD	This I-D.
130	End with REPLACE-CSID, USP & USD	This I-D.
131	End with REPLACE-CSID, PSP, USP & USD	This I-D.
132	End.X with REPLACE-CSID & USD	This I-D.

133	End.X with REPLACE-CSID, PSP & USD	This I-D.
134	End.X with REPLACE-CSID, USP & USD	This I-D.
135	End.X with REPLACE-CSID, PSP, USP & USD	This I-D.
136	End.T with REPLACE-CSID & USD	This I-D.
137	End.T with REPLACE-CSID, PSP & USD	This I-D.
138	End.T with REPLACE-CSID, USP & USD	This I-D.
139	End.T with REPLACE-CSID, PSP, USP & USD	This I-D.

Table 1: Registration List

16. Acknowledgements

The authors would like to thank Kamran Raza, Xing Jiang, YuanChao Su, Han Li, Yisong Liu, and Martin Vigoureux for their insightful feedback and suggestions.

17. References

17.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filtsils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", [RFC 8402](#), DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filtsils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [RFC 8754](#), DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRV6) Network Programming", [RFC 8986](#), DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

17.2. Informative References

- [EMAIL1] "SPRING chairs email on the adoption of [draft-filsfilscheng-spring-srv6-srh-compression-02](#)", October 2021, <<https://mailarchive.ietf.org/arch/msg/spring/VjvIXo7fZFhsIHJ5wFQXIBvvtNM/>>.
- [EMAIL2] "SPRING chairs email on working group process", February 2022, <<https://mailarchive.ietf.org/arch/msg/spring/vCc9Ckvwu5HA-RCleV712dsA50A/>>.
- [I-D.clad-spring-srv6-srh-compression-illus]
Clad, F. and D. Dukes, "Illustrations for Compressed SRV6 Segment List Encoding in SRH", Work in Progress, Internet-Draft, [draft-clad-spring-srv6-srh-compression-illus-02](#), 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-clad-spring-srv6-srh-compression-illus-02>>.
- [I-D.ietf-idr-bgp-ls-sr-policy]
Previdi, S., Talaulikar, K., Dong, J., Gredler, H., and J. Tantsura, "Advertisement of Segment Routing Policies using BGP Link-State", Work in Progress, Internet-Draft, [draft-ietf-idr-bgp-ls-sr-policy-00](#), 9 March 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-ls-sr-policy-00>>.
- [I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRV6", Work in Progress, Internet-Draft, [draft-ietf-idr-bgpls-srv6-ext-14](#), 17 February 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgpls-srv6-ext-14>>.
- [I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, [draft-ietf-idr-segment-routing-te-policy-20](#), 27 July 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-idr-segment-routing-te-policy-20>>.

[I-D.ietf-lsr-ospfv3-srv6-extensions]

Li, Z., Hu, Z., Talaulikar, K., and P. Psenak, "OSPFv3 Extensions for SRV6", Work in Progress, Internet-Draft, [draft-ietf-lsr-ospfv3-srv6-extensions-09](https://datatracker.ietf.org/doc/html/draft-ietf-lsr-ospfv3-srv6-extensions-09), 14 January 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-lsr-ospfv3-srv6-extensions-09>>.

[I-D.ietf-pce-segment-routing-ipv6]

Li, C., Negi, M. S., Sivabalan, S., Koldychev, M., Kaladharan, P., and Y. Zhu, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing leveraging the IPv6 dataplane", Work in Progress, Internet-Draft, [draft-ietf-pce-segment-routing-ipv6-16](https://datatracker.ietf.org/doc/html/draft-ietf-pce-segment-routing-ipv6-16), 6 March 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-pce-segment-routing-ipv6-16>>.

[I-D.srcompdt-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-srcompdt-spring-compression-requirement-07](https://datatracker.ietf.org/doc/html/draft-srcompdt-spring-compression-requirement-07), 11 July 2021, <<https://datatracker.ietf.org/doc/html/draft-srcompdt-spring-compression-requirement-07>>.

[IMPL-CISCO-8000]

Cisco Systems, "Segment Routing Configuration Guide for Cisco 8000 Series Routers", 4 November 2022, <<https://www.cisco.com/c/en/us/td/docs/iosxr/cisco8000/segment-routing/75x/b-segment-routing-cg-cisco8000-75x/configuring-segment-routing-over-ipv6-srv6-micro-sids.html>>.

[IMPL-CISCO-ASR9000]

Cisco Systems, "Segment Routing Configuration Guide for Cisco ASR 9000 Series Routers", 6 November 2022, <<https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r7-5/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-75x/configure-srv6-micro-sid.html>>.

[IMPL-CISCO-NCS540]

Cisco Systems, "Segment Routing Configuration Guide for Cisco NCS 540 Series Routers", 2 November 2022, <<https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5xx/segment-routing/73x/b-segment-routing-cg-73x-ncs540/configure-srv6.html>>.

[IMPL-CISCO-NCS5500]

Cisco Systems, "Segment Routing Configuration Guide for Cisco NCS 5500 Series Routers", 6 November 2022, <<https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/73x/b-segment-routing-cg-ncs5500-73x/configure-srv6-micro-sid.html>>.

[IMPL-CISCO-NCS560]

Cisco Systems, "Segment Routing Configuration Guide for Cisco NCS 560 Series Routers", 14 October 2022, <<https://www.cisco.com/c/en/us/td/docs/iosxr/ncs560/segment-routing/76x/b-segment-routing-cg-76x-ncs560/m-configure-srv6-usid-ncs5xx.html>>.

[IMPL-CISCO-NCS5700]

Cisco Systems, "Segment Routing Configuration Guide for Cisco NCS 5700 Series Routers", 6 November 2022, <<https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/segment-routing/75x/b-segment-routing-cg-ncs5500-75x/configure-srv6-micro-sid.html>>.

[IMPL-NOKIA-20.10]

Nokia, "Segment Routing and PCE User Guide", December 2022, <<https://documentation.nokia.com/sr/22-10/books/Segment%20Routing%20and%20PCE%20User%20Guide/segment-rout-with-ipv6-data-plane-srv6.html>>.

[IMPL-OSS-LINUX]

Abeni, P., "Add NEXT-C-SID support for SRv6 End behavior", 20 September 2022, <<https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net-next.git/commit/?id=cec9d59e89362809f17f2d854faf52966216da13>>.

[IMPL-OSS-ONOS]

Open Networking Foundation, "Stratum CMCC G-SRV6 Project", 24 March 2021, <<https://wiki.opennetworking.org/display/COM/Stratum+CMCC+G-SRV6+Project>>.

[IMPL-OSS-OPEN-SRV6]

"Open SRv6 Project", n.d., <<http://opensrv6.org/en/srv6-2/>>.

[IMPL-OSS-P4]

Salsano, S. and A. Tulumello, "SRv6 uSID (micro SID) implementation on P4", 3 January 2021, <<https://github.com/netgroup/p4-srv6-usid>>.

[IMPL-OSS-SAI]

Agrawal, A., "Added new behaviors to support uSID instruction", 8 June 2021, <<https://github.com/opencomputeproject/SAI/pull/1231/commits/02e58d95ad966ca9efc24eb9e0c0fa10b21de2a4>>.

[IMPL-OSS-SONIC]

Shah, S. and R. Sudarshan, "SONiC uSID", 21 August 2022, <https://github.com/sonic-net/SONiC/blob/master/doc/srv6/SRV6_uSID.md>.

[IMPL-OSS-VPP]

FD.io, "Srv6 cli reference", n.d., <https://s3-docs.fd.io/vpp/23.02/cli-reference/clis/clicmd_src_vnet_srv6.html>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

[RFC9252] Dawra, G., Ed., Talaulikar, K., Ed., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "BGP Overlay Services Based on Segment Routing over IPv6 (SRV6)", [RFC 9252](#), DOI 10.17487/RFC9252, July 2022, <<https://www.rfc-editor.org/info/rfc9252>>.

[RFC9259] Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRV6)", [RFC 9259](#), DOI 10.17487/RFC9259, June 2022, <<https://www.rfc-editor.org/info/rfc9259>>.

[RFC9352] Psenak, P., Ed., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane", [RFC 9352](#), DOI 10.17487/RFC9352, February 2023, <<https://www.rfc-editor.org/info/rfc9352>>.

[SPRING-WG-POLICIES]

SPRING Working Group Chairs, "SPRING Working Group Policies", 14 October 2022, <https://wiki.ietf.org/en/group/spring/WG_Policies>.

[Appendix A. Open Issues](#)

This section is to be removed before publishing as an RFC.

This section was added as requested by the SPRING chair in [[EMAIL1](#)].

Issues raised during and after the adoption call for this draft are tracked in an issue tracker. The remainder of this section identifies the most significant open issues, from the adoption call, for the working group to keep track of.

As a reminder to those reading this section, this document is a work in progress, and subject to change by the working group. As noted at the front of this document, "It is inappropriate to use Internet-Drafts as reference material"

- * Given that the working group has said that it wants to standardize one data plane solution, and given that the document contains multiple SRV6 EndPoint behaviors that some WG members have stated are multiple data plane solutions, the working group will address whether this is valid and coherent with its one data plane solution objective.
- * As reminded in the conclusion of the adoption call, this document is subject to the policy announced by the SPRING chairs in [\[EMAIL2\]](#). In particular, this means that this document can not go to WG last call until 6man completes handling of an Internet Draft that deals with the relationship of C-SIDs to [RFC 4291](#). It is hoped and expected that said resolution will be a WG last call and document approval in 6man of a document providing for the way that C-SIDs use the IPv6 destination address field.

[Appendix B](#). Complete pseudocodes

The content of this section is purely informative rendering of the pseudocodes of [\[RFC8986\]](#) with the modifications in this document. This rendering may not be used as a reference.

[B.1](#). End with NEXT-C-SID

When processing the SRH of a packet matching a FIB entry locally instantiated as an End SID with the NEXT-C-SID flavor:


```
S01. If (DA.Argument != 0) {
S02.   If (IPv6 Hop Limit <= 1) {
S03.     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.
S04.   }
S05.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
           of the Destination Address.
S06.   Set the bits [(LBL+AL)..127] of the Destination Address to
           zero.
S07.   Decrement IPv6 Hop Limit by 1.
S08.   Submit the packet to the egress IPv6 FIB lookup for
           transmission to the next destination.
S9.   }
S10. If (Segments Left == 0) {
S11.   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.
S12. }
S13. If (IPv6 Hop Limit <= 1) {
S14.   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.
S15. }
S16. max_LE = (Hdr Ext Len / 2) - 1
S17. If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S18.   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.
S19. }
S20. Decrement IPv6 Hop Limit by 1.
S21. Decrement Segments Left by 1.
S22. Update IPv6 DA with Segment List[Segments Left].
S23. Submit the packet to the egress IPv6 FIB lookup for
           transmission to the new destination.
```

Before processing the Upper-Layer header or any IPv6 extension header other than Hop-by-Hop or Destination Option of a packet matching a FIB entry locally instantiated as an End SID with the NEXT-C-SID flavor:


```
S01. If (DA.Argument != 0) {
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.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
        of the Destination Address.
S06.   Set the bits [(LBL+AL)..127] of the Destination Address to
        zero.
S07.   Decrement Hop Limit by 1.
S08.   Submit the packet to the egress IPv6 FIB lookup for
        transmission to the next destination.
S09. }
```

When processing the Upper-Layer header of a packet matching a FIB entry locally instantiated as an End SID with the NEXT-C-SID flavor:

```
S01. If (Upper-Layer header type is allowed by local configuration) {
S02.   Proceed to process the Upper-Layer header
S03. } Else {
S04.   Send an ICMP Parameter Problem to the Source Address
        with Code 4 (SR Upper-layer Header Error)
        and Pointer set to the offset of the Upper-Layer header,
        interrupt packet processing, and discard the packet.
S05. }
```

B.2. End with REPLACE-C-SID

When processing the SRH of a packet matching a FIB entry locally instantiated as an End SID with the REPLACE-C-SID 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
S09.   If (DA.Arg.Index != 0) {
S10.     If ((Last Entry > max_LE) or (Segments Left > Last Entry)) {
S11.       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.
S12.     }
S13.     Decrement DA.Arg.Index by 1.
S14.     If (Segment List[Segments Left][DA.Arg.Index] == 0) {
S15.       Decrement Segments Left by 1.
S16.       Decrement IPv6 Hop Limit by 1.
S17.       Update IPv6 DA with Segment List[Segments Left]
S18.       Submit the packet to the egress IPv6 FIB lookup for
        transmission to the new destination.
S19.     }
S20.   } Else {
S21.     If((Last Entry > max_LE) or (Segments Left > Last Entry+1)){
S22.       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.
S23.     }
S24.     Decrement Segments Left by 1.
S25.     Set DA.Arg.Index to (128/NF - 1).
S26.   }
S27.   Decrement IPv6 Hop Limit by 1.
S28.   Write Segment List[Segments Left][DA.Arg.Index] into the bits
      [B..B+NF-1] of the Destination Address of the IPv6 header.
S29.   Submit the packet to the egress IPv6 FIB lookup for
      transmission to the new destination.
S30. }
```

When processing the Upper-Layer header of a packet matching a FIB entry locally instantiated as an End SID with the REPLACE-C-SID flavor:


```
S01. If (Upper-Layer header type is allowed by local configuration) {
S02.   Proceed to process the Upper-Layer header
S03. } Else {
S04.   Send an ICMP Parameter Problem to the Source Address
        with Code 4 (SR Upper-layer Header Error)
        and Pointer set to the offset of the Upper-Layer header,
        interrupt packet processing, and discard the packet.
S05. }
```

Contributors

Liu Aihua
ZTE Corporation
China
Email: liu.aihua@zte.com.cn

Dennis Cai
Alibaba
United States of America
Email: d.cai@alibaba-inc.com

Darren Dukes
Cisco Systems, Inc.
Canada
Email: ddukes@cisco.com

James N Guichard
Futurewei Technologies Ltd.
United States of America
Email: james.n.guichard@futurewei.com

Cheng Li
Huawei Technologies
China
Email: c.l@huawei.com

Robert Raszuk
NTT Network Innovations
United States of America
Email: robert@raszuk.net

Ketan Talaulikar
Cisco Systems, Inc.
India
Email: ketant.ietf@gmail.com

Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Shay Zadok
Broadcom
Israel
Email: shay.zadok@broadcom.com

Authors' Addresses

Weiqliang Cheng (editor)
China Mobile
China
Email: chengweiqliang@chinamobile.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium
Email: cf@cisco.com

Zhenbin Li
Huawei Technologies
China
Email: lizhenbin@huawei.com

Bruno Decraene
Orange
France
Email: bruno.decraene@orange.com

Francois Clad (editor)
Cisco Systems, Inc.
France
Email: fclad@cisco.com

