

SPRING
Internet-Draft
Intended status: Standards Track
Expires: 31 August 2026

W. Cheng, Ed.
China Mobile
C. Li, Ed.
Huawei Technologies
27 February 2026

Combined NEXT-CSID and REPLACE-CSID flavor in SRv6
draft-cl-spring-srv6-next-and-replace-00

Abstract

In order to reduce the size of SRv6 SID, NEXT-CSID and REPLACE-CSID flavors for SRv6 endpoint behaviors are proposed. Similar to PSP and USD, NEXT-CSID and REPLACE-CSID can be combined just like PSP & USD. This document defines the combined NEXT&REPLACE-CSID flavor, which can provide more efficient compression for 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 31 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Requirements Language	3
3. NEXT&REPLACE-CSID Flavor {#sec-next&replace-flavor}	3
3.1. End with NEXT&REPLACE-CSID {#sec-next&replace-end}	5
4. SR Source Node	6
4.1. SID Validation for Compression	6
4.2. Segment List Compression	6
4.3. Upper-Layer Checksums	6
5. Control Plane	6
6. Operational Considerations	6
7. Security Considerations	7
8. IANA Considerations	7
8.1. SRv6 Endpoint Behaviors	7
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	9
Appendix A. Complete Pseudocodes	10
A.1. End with NEXT&REPLACE-CSID	10
A.2. End.X with NEXT&REPLACE-CSID	12
A.3. Other End behaviors with NEXT&REPLACE-CSID	13
Authors' Addresses	14

1. Introduction

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].

[RFC9800] specifies NEXT-CSID and REPLACE-CSID 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.ietf-spring-compression-requirement].

NEXT-CSID flavor can enable an efficient segment list encoding when using a short locator block. However, when the locator block is long, the compression efficiency is reduced, because fewer CSIDs can be encoded in a CSID container. This will affect all the containers because NEXT-CSID flavor requires to encode the locator block in each CSID container. REPLACE-CSID flavor can enable an efficient segment list encoding no matter the length of locator block. However, when the CSID sequence is short, the compression efficiency is not ideal, because the first container of the CSID sequence will only contain a single CSID without compression, and the compression begins from the second container and the block is not encoded in a container.

Similar to PSP, USP, USD defined in [RFC8986], NEXT-CSID and REPLACE-CSID flavor can be combined in use. By combining NEXT-CSID and REPLACE-CSID flavor, the compression efficiency can be enhanced comparing to both NEXT-CSID and REPLACE-CSID flavor. Originally, the Combined NEXT&REPLACE-CSID flavor was defined in [RFC9800], and moved out of the document in order to accelerate the standardization progress of [RFC9800]. This document defines the combined NEXT&REPLACE-CSID flavor, which leverages the SRv6 data plane defined in [RFC8754] and [RFC8986], and are compatible with the SRv6 control plane extensions.

2. Terminology

This document leverages the terminologies defined in [RFC8402], [RFC8754], [RFC8986] and [RFC9800]. Readers are assumed to be familiar with terminologies.

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. NEXT&REPLACE-CSID Flavor {#sec-next&replace-flavor}

This section defines a new flavor NEXT&REPLACE-CSID, which is the combination of NEXT-CSID and REPLACE-CSID [RFC9800]. The NEXT&REPLACE-CSID can be applied for END, END.X, END.T, End.B6.Encaps, End.B6.Encaps.Red, End.BM, End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DT2U, and End.DT2M behaviors defined in [RFC8986].

Similar to NEXT-CSID flavor and the REPLACE-CSID flavor, the combined NEXT&REPLACE flavor leverages the SID Argument to determine the next SID to be processed. A SID instantiated with the NEXT&REPLACE-CSID flavor can take an argument to

- * carry the remaining CSIDs in the current CSID container, same as NEXT-CSID flavor [RFC9800].
- * indicate the index of the next CSID in the appropriate CSID container, same as REPLACE-CSID flavor [RFC9800].

The length of the argument equals to 128-LBL-LNFL.

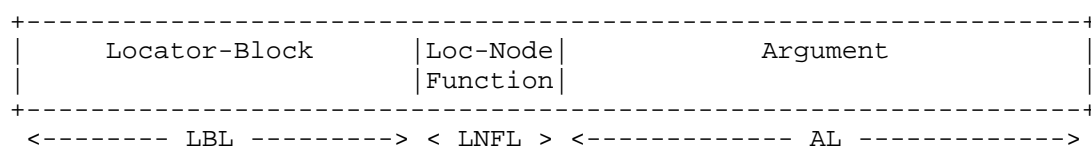


Figure 1: Example of a NEXT&REPLACE-CSID flavored SID structure using a 48-bit Locator-Block, 16-bit combined locator and function, and 64-bit argument

The encoding rules of the NEXT&REPLACE-CSID flavor is a combination of NEXT-CSID and REPLACE-CSID.

- * First container: follows the rules of encoding the first container of NEXT-CSID flavor [RFC9800].
- * Other containers: follows the rules of encoding the other containers of REPLACE-CSID flavor [RFC9800].

The following figure shows an example of encoding a segment list using NEXT&REPLACE-CSID flavor SIDs.

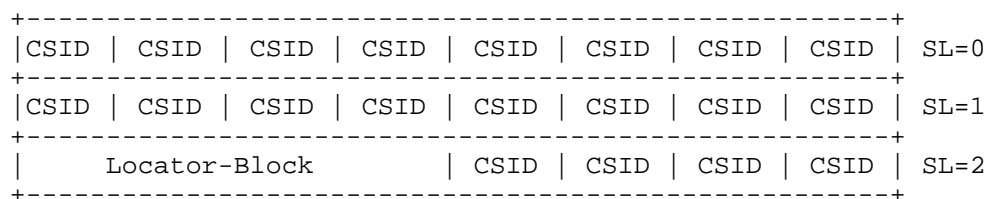


Figure 2: NEXT&REPLACE-CSID SIDs Encoding

The NEXT&REPLACE-CSID flavor supports both 16- and 32-bit CSID lengths. A CSID length of 16-bit is RECOMMENDED. The length of a CSID is determined by its behavior and LNFL by reusing the rules of REPLACE-CSID flavor defined in section 4.2 of [RFC9800].

3.1. End with NEXT&REPLACE-CSID {#sec-next&replace-end}

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

The S01 of NEXT-CSID flavor pseudo code [RFC9800] is replaced by the below pseudo code, and the updated pseudo code is inserted between lines S01 and S02 of the REPLACE-CSID flavor pseudo code [RFC9800], 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.Arg.Next != 0) {
```

Notes:

- * DA.Arg.Next identifies the bits [(LBL+LNFL)..(LBL+2*LNFL-1)] in the DA of the IPv6 header, which is the next CSID if it appears.

According to the pseudo code, when the next CSID in the DA is not zero, then the node performs NEXT-CSID processing, otherwise, it performs the REPLACE-CSID processing. Therefore, the best compression can be achieved by combining the advantages of NEXT-CSID flavor and REPLACE-CSID flavor. An high-level pseudo code is provided for better understanding.

```
S01. If (DA.Arg.Next != 0) {  
S02.     NEXT-CSID flavor processing  
S03. } else{  
S04.     REPLACE-CSID flavor processing  
S05. }
```

A rendering of the complete pseudocode of END with NEXT&REPLACE-CSID is provided in Appendix A.1. This modification applies to other behaviors such as END.X, END.T, END.B6.Encaps, and End.B6.Encaps.Red when they combine with NEXT&REPLACE-CSID flavor.

All the complete pseudocode of END behaviors are defined in Appendix A.1.

4. SR Source Node

The NEXT&REPLACE-CSID does not bring new requirements comparing to NEXT-CSID and REPLACE-CSID flavor defined in section 6 of [RFC9800].

4.1. SID Validation for Compression

The rules defined in section 6.1 of [RFC9800] apply to NEXT&REPLACE-CSID.

4.2. Segment List Compression

An SR source node MAY compress a segment list when it includes NEXT&REPLACE-CSID flavor SIDs. If an SR source node chooses to compress the segment list, a method is defined below.

- * First container: reuses the rules of NEXT-CSID flavor for the first container.
- * Other containers: reuses the rules of REPLACE-CSID flavor for the other containers(except the first container).

Please see the figure {NEXT&REPLACE-CSID SIDs Encoding} for an example of encoding a segment list using NEXT&REPLACE-CSID flavor SIDs.

4.3. Upper-Layer Checksums

The rules defined in section 6.5 of [RFC9800] apply to NEXT&REPLACE-CSID.

5. Control Plane

The rules defined in section 8 of [RFC9800] apply to NEXT&REPLACE-CSID.

6. Operational Considerations

The rules defined in section 9 of [RFC9800] apply to NEXT&REPLACE-CSID.

In order to enhance the compression efficiency, this document recommends to use 16-bit NEXT&REPLACE-CSID, which can provide the best compression.

7. 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.

8. IANA Considerations

8.1. 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
201	End with NEXT&REPLACE-CSID	This I-D.
202	End with NEXT&REPLACE-CSID & PSP	This I-D.
203	End with NEXT&REPLACE-CSID & USP	This I-D.
204	End with NEXT&REPLACE-CSID, PSP & USP	This I-D.
205	End.X with NEXT&REPLACE-CSID	This I-D.
206	End.X with NEXT&REPLACE-CSID & PSP	This I-D.
207	End.X with NEXT&REPLACE-CSID & USP	This I-D.
208	End.X with NEXT&REPLACE-CSID, PSP & USP	This I-D.
209	End.T with NEXT&REPLACE-CSID	This I-D.
210	End.T with NEXT&REPLACE-CSID & PSP	This I-D.
211	End.T with NEXT&REPLACE-CSID & USP	This I-D.
212	End.T with NEXT&REPLACE-CSID, PSP & USP	This I-D.
214	End.B6.Encaps with NEXT&REPLACE-CSID	This I-D.
215	End.BM with NEXT&REPLACE-CSID	This I-D.
216	End.DX6 with NEXT&REPLACE-CSID	This I-D.

217	End.DX4 with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
218	End.DT6 with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
219	End.DT4 with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
220	End.DT46 with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
221	End.DX2 with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
222	End.DX2V with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
223	End.DX2U with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
224	End.DT2M with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
227	End.B6.Encaps.Red with NEXT&REPLACE-CSID	This I-D.
+-----+		+-----+
228	End with NEXT&REPLACE-CSID & USD	This I-D.
+-----+		+-----+
229	End with NEXT&REPLACE-CSID, PSP & USD	This I-D.
+-----+		+-----+
230	End with NEXT&REPLACE-CSID, USP & USD	This I-D.
+-----+		+-----+
231	End with NEXT&REPLACE-CSID, PSP, USP & USD	This I-D.
+-----+		+-----+
232	End.X with NEXT&REPLACE-CSID & USD	This I-D.
+-----+		+-----+
233	End.X with NEXT&REPLACE-CSID, PSP & USD	This I-D.
+-----+		+-----+
234	End.X with NEXT&REPLACE-CSID, USP & USD	This I-D.
+-----+		+-----+
235	End.X with NEXT&REPLACE-CSID, PSP, USP & USD	This I-D.
+-----+		+-----+
236	End.T with NEXT&REPLACE-CSID & USD	This I-D.
+-----+		+-----+
237	End.T with NEXT&REPLACE-CSID, PSP & USD	This I-D.
+-----+		+-----+
238	End.T with NEXT&REPLACE-CSID, USP & USD	This I-D.
+-----+		+-----+
239	End.T with NEXT&REPLACE-CSID, PSP, USP & USD	This I-D.
+-----+		+-----+

Table 1: Registration List

9. Acknowledgements

Many thanks to Yang Xia, Ka Zhang, Chenxi Li, Fang Qiu, Weidi Shi, Chao Wang, Yang Huang, Shuying Liu, Yisong Liu, Xinyue Zhang, Weier Li for their professional supports.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [RFC9800] Cheng, W., Ed., Filsfils, C., Li, Z., Decraene, B., and F. Clad, Ed., "Compressed SRv6 Segment List Encoding", RFC 9800, DOI 10.17487/RFC9800, June 2025, <<https://www.rfc-editor.org/info/rfc9800>>.

10.2. Informative References

[I-D.ietf-spring-compression-requirement]

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

Appendix A. Complete Pseudocodes

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

A.1. End with NEXT&REPLACE-CSID

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

```
S01. When an SRH is processed {
S1.1   If (DA.Arg.Next != 0) {                               //NEXT-CSID Processing
S1.2.   If (IPv6 Hop Limit <= 1) {
S1.3.     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.
S1.4.   }
S1.5.   Copy the value of DA.Argument into the bits [LBL..(LBL+AL-1)]
           of the Destination Address.
S1.6.   Set the bits [(LBL+AL)..127] of the Destination Address to
           zero.
S1.7.   Decrement IPv6 Hop Limit by 1.
S1.8.   Submit the packet to the egress IPv6 FIB lookup for
           transmission to the next destination.
S1.9.   } //Below is REPLACE-CSID Processing
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.      }
```

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&REPLACE-CSID flavor:

```
S01. If (DA.Arg.Next != 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&REPLACE-CSID 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. }
```

A.2. End.X with NEXT&REPLACE-CSID

When processing the SRH of a packet matching a FIB entry locally instantiated as an End.X SID with the NEXT&REPLACE-CSID flavor:

```
S01. When an SRH is processed {
S02.   If (DA.Arg.Next != 0) {                                //NEXT-CSID Processing
S03.     NEXT-CSID flavor END.X processing as per {{RFC9800}}
S04.   }                                                        //Below is REPLACE-CSID Processing
S05.     REPLACE-CSID flavor END.X processing as per {{RFC9800}}.
S06. }
```

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.X SID with the NEXT&REPLACE-CSID flavor:

```
S01. If (DA.Arg.Next != 0) {
S02.   NEXT-CSID flavor END.X processing as per {{RFC9800}}
S03. }
```

When processing the Upper-Layer header of a packet matching a FIB entry locally instantiated as an End.X SID with the NEXT&REPLACE-CSID flavor, it reuses the pseudo code of NEXT-CSID flavor [RFC9800].

```
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. }
```

A.3. Other End behaviors with NEXT&REPLACE-CSID

Similar to END and END.X, the other behaviors including END.T, End.B6.Encaps, End.B6.Encaps.Red, End.BM, End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DT2U, and End.DT2M are following the same logic of combining NEXT-CSID and REPLACE-CSID flavor pseudo code.

When processing the SRH of a packet matching a FIB entry locally instantiated as one of the END.T, End.B6.Encaps, End.B6.Encaps.Red, End.BM, End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DT2U, and End.DT2M SID with the NEXT&REPLACE-CSID flavor:

```
S01. When an SRH is processed {
S02.   If (DA.Arg.Next != 0) {                               //NEXT-CSID Processing
S03.     NEXT-CSID flavor processing as per {{RFC9800}}
S04.   }                                                     //Below is REPLACE-CSID Processing
S05.     REPLACE-CSID flavor processing as per {{RFC9800}}.
S06. }
```

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 one of the END.T, End.B6.Encaps, End.B6.Encaps.Red, End.BM, End.DX6, End.DX4, End.DT6, End.DT4, End.DT46, End.DX2, End.DX2V, End.DT2U, and End.DT2M SID with the NEXT&REPLACE-CSID flavor:

```
S01. If (DA.Arg.Next != 0) {
S02.   NEXT-CSID flavor END.X processing as per {{RFC9800}}
S03. }
```

When processing the Upper-Layer header of a packet matching a FIB entry locally instantiated as one of the END.T, End.B6.Encaps, End.B6.Encaps.Red, End.BM, End.DX6, End.DX4, End.DT6, End.DT4,

End.DT46, End.DX2, End.DX2V, End.DT2U, and End.DT2M SID with the NEXT&REPLACE-CSID flavor, it reuses the pseudo code of NEXT-CSID flavor [RFC9800].

```
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. }
```

Authors' Addresses

Weiqiang Cheng (editor)
China Mobile
China
Email: chengweiqiang@chinamobile.com

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