

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 14, 2021

Z. Li
C. Li
Huawei Technologies
W. Cheng
China Mobile
C. Xie
C. Li
China Telecom
H. Tian
F. Zhao
CAICT
February 10, 2021

Generalized Segment Routing Header
draft-lc-6man-generalized-srh-03

Abstract

Generalized SRv6 network programming defines the enhanced mechanisms of SRv6 to encode SRv6 SIDs, Compressed SIDs and even the MPLS labels or IPv4 tunnel information in a single SRH. This type of SRH is called Generalized SRH (G-SRH), which can reduce the overhead of SRv6 and also provide more flexibility for network programming. This document defines the encapsulation and packet processing of G-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 August 14, 2021.

Copyright Notice

Copyright (c) 2021 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
2.1. Requirements Language	3
3. Generalized Segment Identifier	4
3.1. SRv6 G-SID	4
3.2. Compression G-SID	4
3.3. MPLS G-SID	5
3.4. IPv4 G-SID	6
4. Generalized Segment Routing Header (G-SRH)	7
4.1. G-SRH for SRv6 Compression	8
4.2. G-SRH for Crossing SR-MPLS Domains	10
4.3. G-SRH for Crossing IPv4 Domains	11
5. Packet Processing	14
5.1. Crossing SRv6 Compression Domains	14
5.2. Crossing SR-MPLS Domains	17
5.2.1. END.M	17
5.3. Crossing IPv4 Domains	17
5.3.1. END.4	17
5.3.2. END.B4	18
6. IANA Considerations	19
7. Security Considerations	19
8. Contributors	19
9. Acknowledgements	19
10. References	19
10.1. Normative References	19
10.2. Informative References	20
Authors' Addresses	20

1. Introduction

Segment routing (SR) [[RFC8402](#)] is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node by inserting an ordered list of instructions, called segments.

When segment routing is deployed on the IPv6 data plane, it is called SRv6 [[I-D.ietf-6man-segment-routing-header](#)]. For support of SR, a

new routing header called Segment Routing Header (SRH), which contains a list of SIDs and other information, has been defined in [I-D.ietf-6man-segment-routing-header]. In use cases like Traffic Engineering, an ordered SID List with multiple SIDs is inserted into the SRH to steer packets along an explicit path.

The overhead of SIDs (16 bytes per SID) in SRH proposes challenges for packet processing and payload efficiency. [I-D.li-spring-compressed-srv6-np] proposes a mechanism for reducing the overhead of SRv6 by using the Compressed SID (C-SID) in SRH. However it requires all the middle segment endpoint nodes to support compression.

[I-D.cl-spring-generalized-srv6-np] proposes Generalized Segment Routing over IPv6 (G-SRv6) Networking Programming, which supports to encode multiple types of Segments in an SRH, called Generalized SRH(G-SRH). These segments can be called Generalized Segment, and the ID can be Generalized Segment Identifier(G-SID), which may include an SRv6 SID(128 bits), C-SIDs, MPLS labels, or IPv4 tunnel information. Therefore, G-SRv6 does not require all the nodes along the path to be compression capable, which supports incremental deployment, and brings benefits to the networks. Moreover it provides more flexibilities for network programming. This document defines the encapsulation and packet processing of Generalized SRH.

2. Terminology

This document makes use of the terms defined in [I-D.ietf-6man-segment-routing-header], [RFC8402] and [RFC8200], and the reader is assumed to be familiar with that terminology. This document introduces the following new terms:

C-SRH: Compressed Segment Routing Header

C-SID: Compressed Segment Identifier

G-SRH: Generalized Segment Routing Header

G-SID: Generalized Segment Identifier.

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. Generalized Segment Identifier

A G-SID is a 128 bits value, and it can be used for carry multiple types of Segment ID or segment information, so may contain:

- o an SRv6 SID
- o a compression G-SID
- o an MPLS G-SID
- o an IPv4 G-SID

3.1. SRv6 G-SID

SRv6 SID is a type of G-SID. A G-SID contains a single SRv6 SID, and does not change anything of SRv6 SID.

3.2. Compression G-SID

As per [I-D.li-spring-compressed-srv6-np], a C-SID is a sub-set of a compressable SRv6 SID, which can be used for routing the packets in compressed SRv6 network programming. The format of compressable SID and C-SID is shown below. The argument part is specified by use cases, and it is zero by default. It is shared by the compressable SRv6 SIDs in a C-SRv6 sub-path.

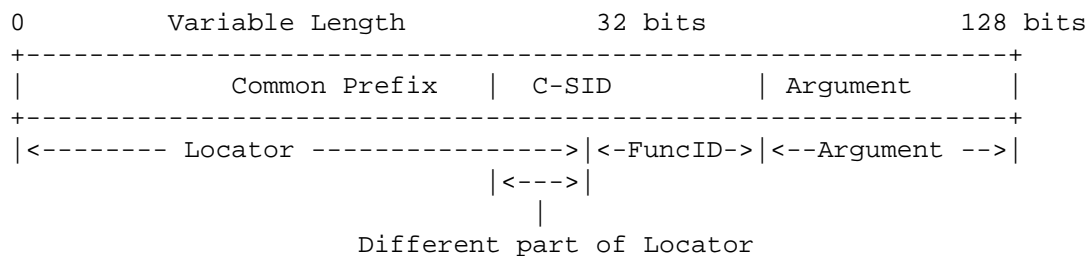


Figure 1. 32 bits C-SID in Compressible SRv6 SID

A compression G-SID shown in Figure 2 may contain several C-SIDs and optional padding. When C-SID is a 32 bits value, a compression G-SID can consist of 4 C-SIDs. If the length of C-SIDs in a G-SID is less than 128 bits, then padding is required.

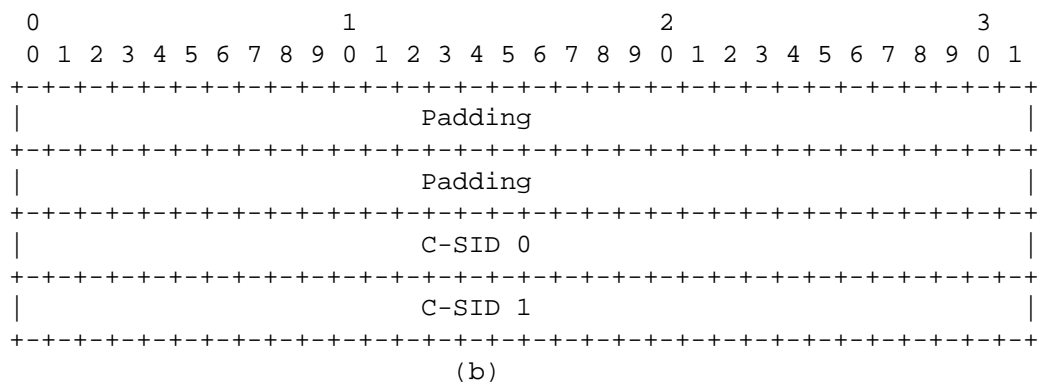
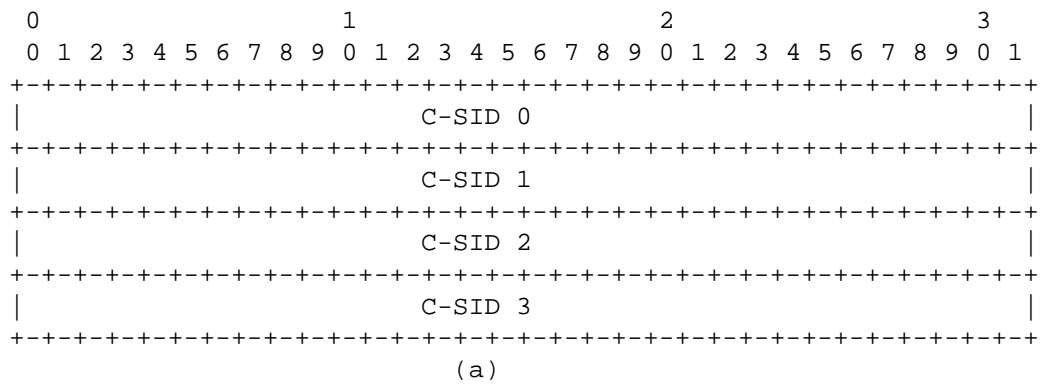


Figure 2. Compression G-SID

3.3. MPLS G-SID

An MPLS G-SID shown in Figure 3 can contain 4 MPLS Label Encapsulations, if number of MPLS Label Encapsulations is less than 4, then padding is required in G-SID to align with 128 bits.

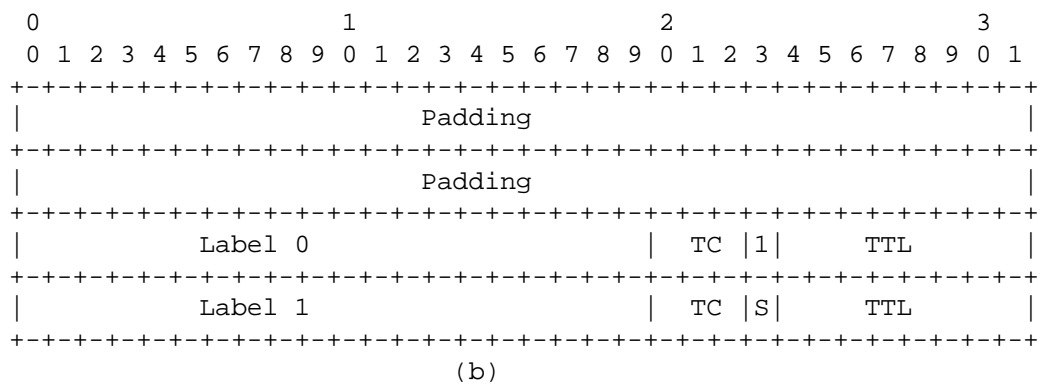
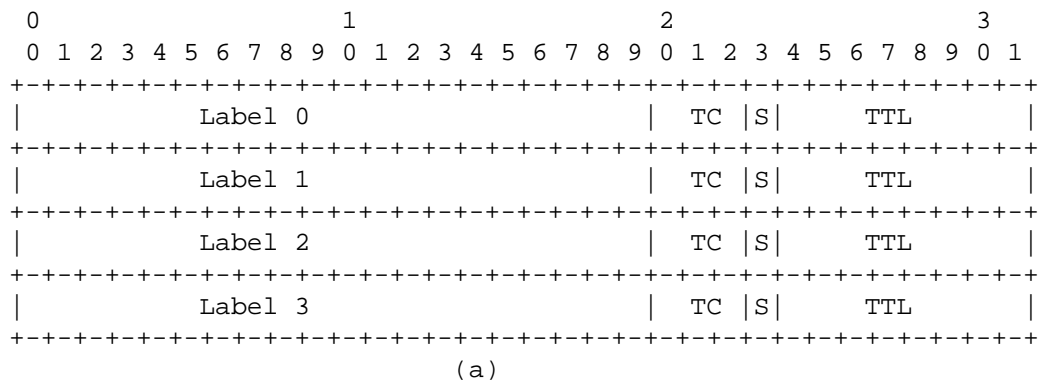


Figure 3. MPLS G-SID

3.4. IPv4 G-SID

An IPv4 G-SID shown in Figure 4 can contain 128 bits information of IPv4 tunnel. The format of the IPv4 G-SID is shown below.

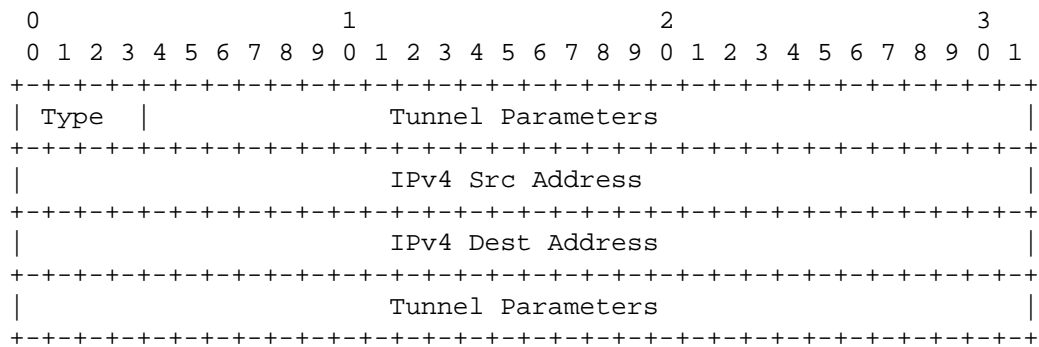


Figure 4. IPv4 G-SID

4. Generalized Segment Routing Header (G-SRH)

Generalized Segment Routing Header (G-SRH) is an enhancement of the SRH, and it supports to encode different types of G-SIDs in a single SRH. The format of G-SRH is shown as below.

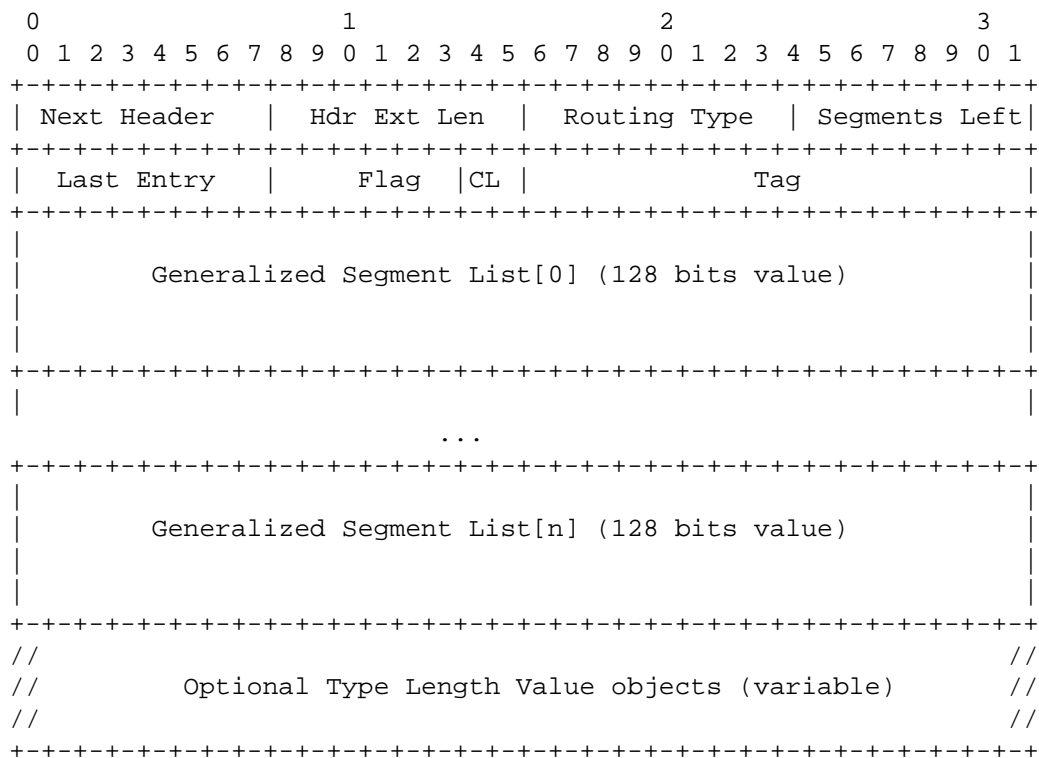


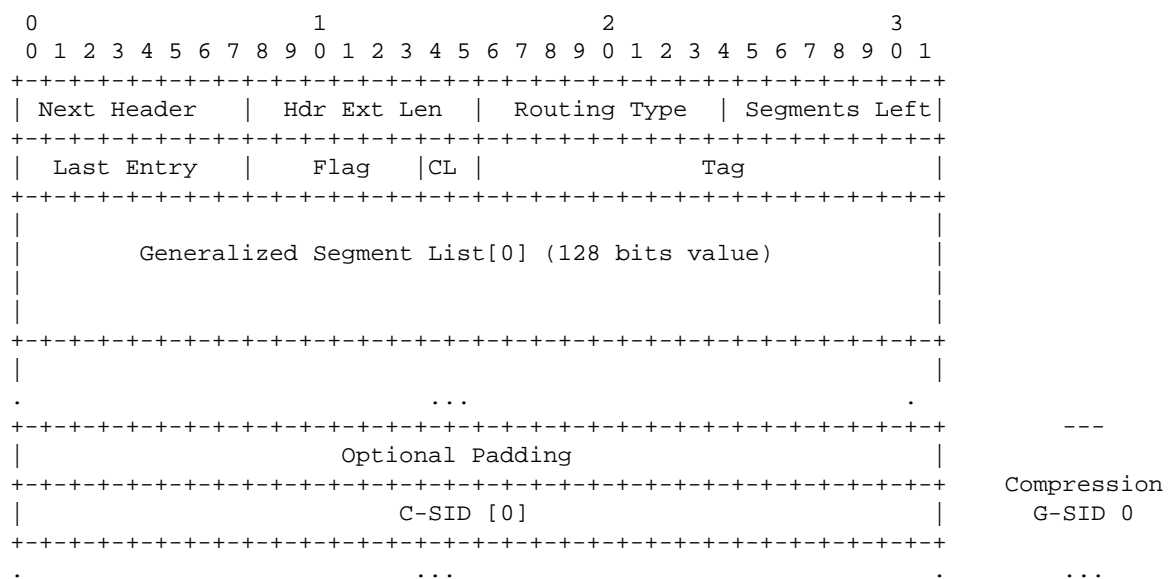
Figure 5. Generalized Segment Routing Header

Where:

- o Next Header, Hdr Ext Len, Routing Type, Segments Left, Last Entry and Tag are the same as those defined in [\[I-D.ietf-6man-segment-routing-header\]](#).
- o CL (C-SID Left): 2-bit unsigned integer to indicate the index of C-SID in the Compression G-SID. There can be four 32-bit C-SIDs within a Compression G-SID.
- o Segment List[n]: 128-bit G-SID representing the nth Generalized Segment in the Segment List. In G-SRH, the Generalized Segment can be an SRv6 SID, Compression G-SID, MPLS G-SID, or IPv4 G-SID, etc.
- o Type Length Value (TLV): It is the same as that defined in [\[I-D.ietf-6man-segment-routing-header\]](#).

4.1. G-SRH for SRv6 Compression

When an SRv6 path travels normal SRv6 domains and compressed SRv6 domains, the SRv6 SID and Compression G-SID(containing C-SIDs) can be encoded in a single G-SRH. For easier understanding, this document assumes that the compressable consists of 96 bits common prefix and 32 bits C-SID, while actually the length of common prefix can be configured as less than 96 to fit the address planning of the networks. The encoding can be shown as follows.



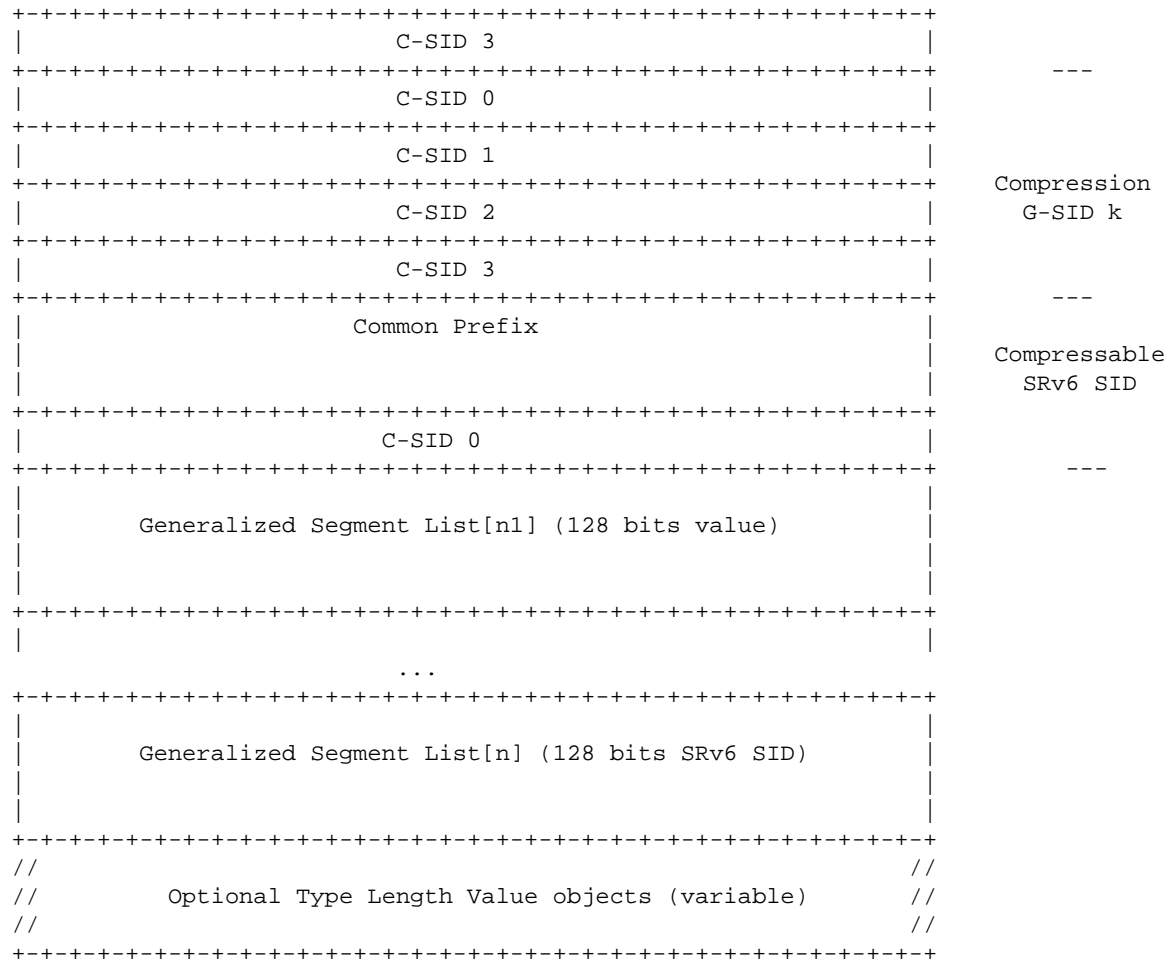


Figure 6. G-SRH for SRv6 Compression

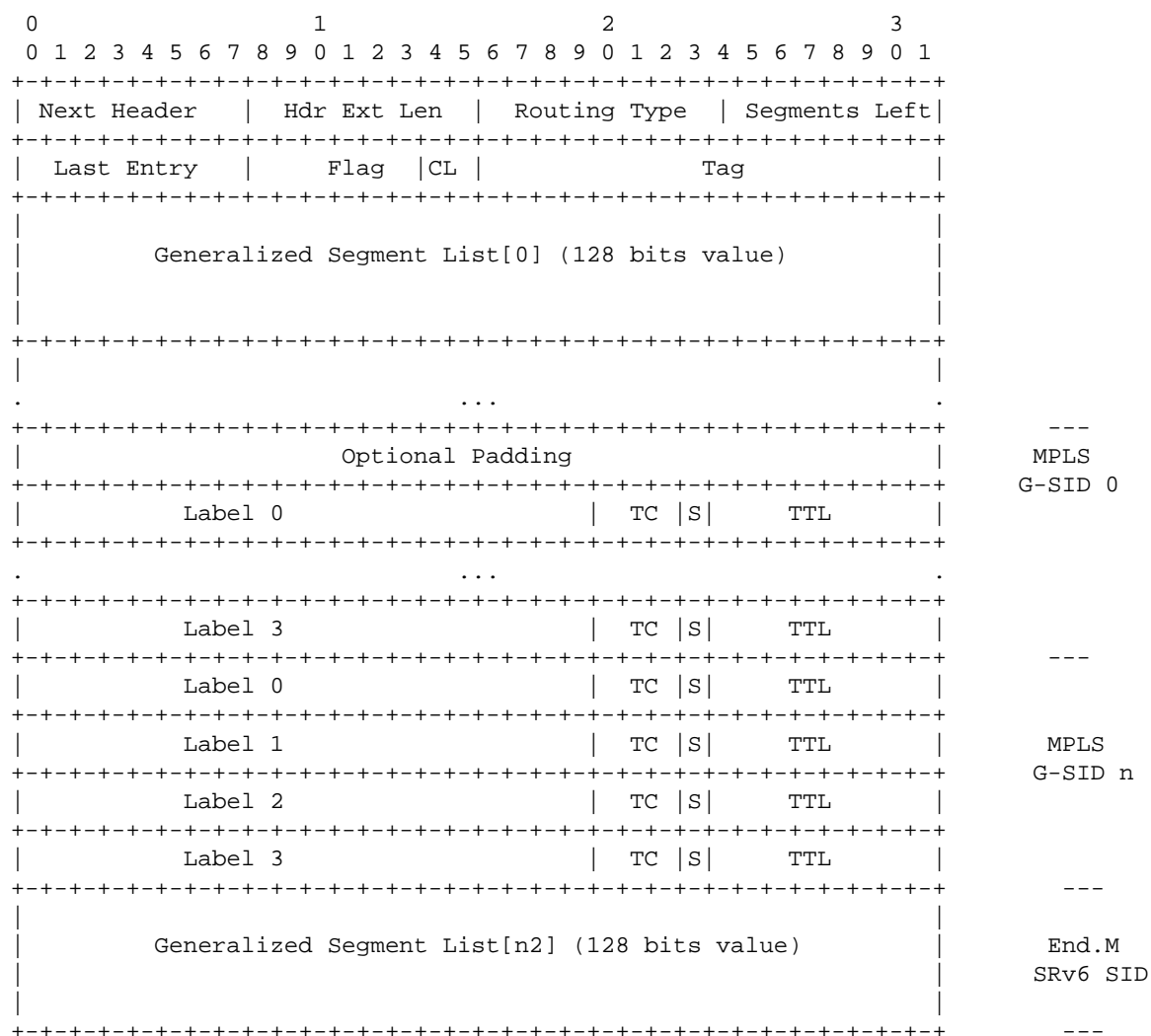
Where:

- o Common Prefix: 96-bit value for the common prefix shared by a group of Compressable SRv6 SIDs. Operators are free to configure the length and the value of the common prefix based on the address planning of their networking. In a Compressable SRv6 SID, when the length of Common prefix is less than 96, and the length of the C-SID is less than 32 bits, the the rest bits can be used as arguments.

- o C-SID: 32-bit compressed SID of a Compressable SRv6 SID.
- o Padding: Must be zero. When the length of C-SIDs within G-SID is less than 128 bits within a compression G-SID, then padding is needed.

4.2. G-SRH for Crossing SR-MPLS Domains

The G-SRH support to encode the MPLS labels and SRv6 SID or Compression G-SID within a G-SRH. The G-SRv6 path for crossing SR-MPLS domains can be encoded in the G-SRH as follows:



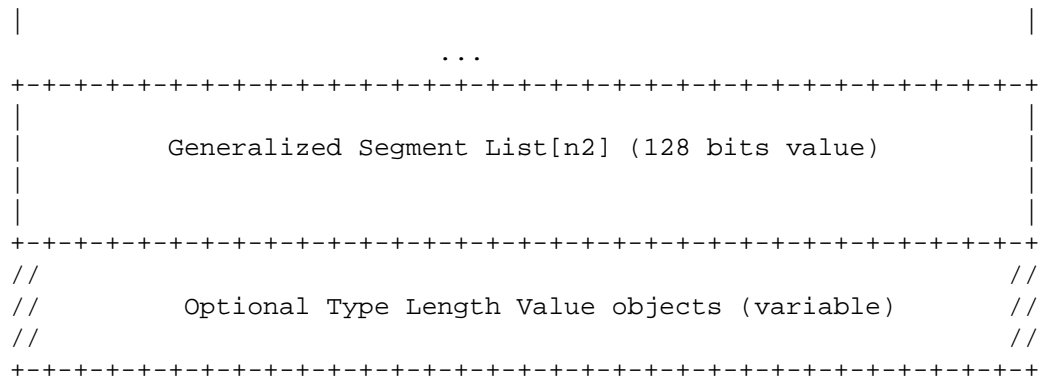


Figure 7. G-SRH for Crossing SR-MPLS Domains

Where:

- o Label: 32-bit MPLS label.
- o TC: 3-bit value for traffic class of MPLS label encapsulation.
- o S: 1-bit flag to indicate the bottom of MPLS label stack. It can be used for indicating the end of the MPLS label stack.
- o TTL: 8-bit value for TTL of MPLS label encapsulation.
- o Padding: When the number of MPLS labels is less than 4 in an MPLS G-SID, then padding is required to align with 128 bits.

4.3. G-SRH for Crossing IPv4 Domains

The G-SRv6 path for crossing IPv4 domains can be encoded in the G-SRH as follows:

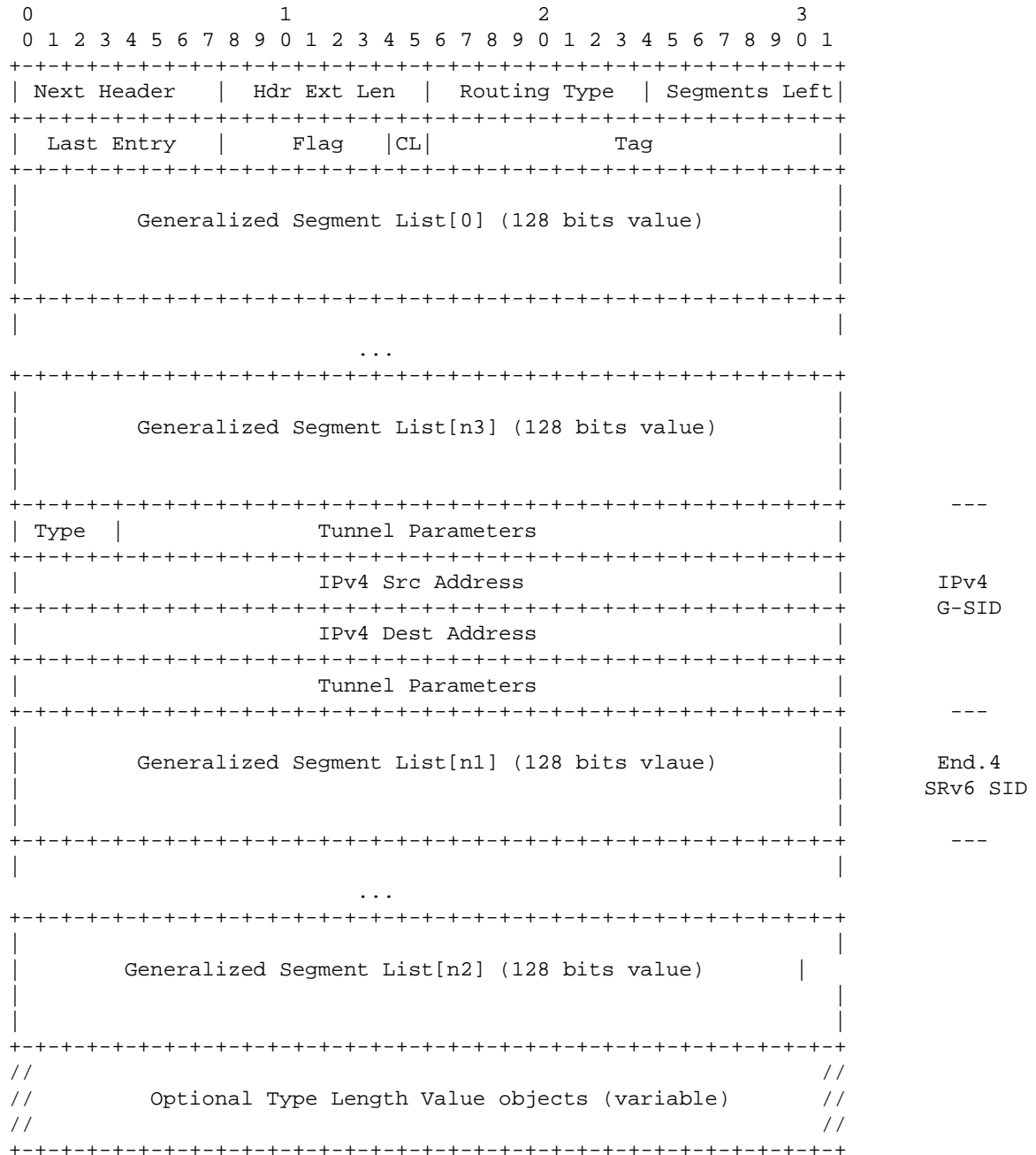


Figure 8. G-SRH for Crossing IPv4 Domains

Where:

- o Type: 4-bit value to indicate the types of IPv4 tunnel.
- o IPv4 Src Address: 32-bit value for the source address of the IPv4 tunnel.
- o IPv4 Dest Address: 32-bit value for the destination address of the IPv4 tunnel.
- o Tunnel Parameters: tunnel parameters of the IPv4 tunnel except the source address and destination address.

This document defines the encoding for IPv4 UDP tunnel and VxLAN tunnel.

The IPv4 UDP tunnel information encapsulated in the G-SRH is as follows:

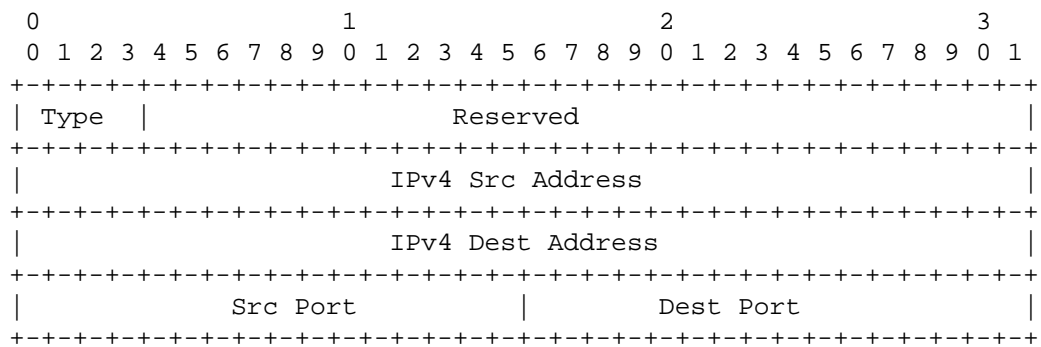


Figure 9. G-SID for IPv4 UDP Tunnel

Where,

- o Reserved: It is the reserved field and MUST be set as 0.
- o Src Port: 16-bit value for the source port of the IPv4 UDP tunnel.
- o Dest Port: 16-bit value for the destination port of the IPv4 UDP tunnel.

The IPv4 VXLAN tunnel information encapsulated in the G-SRH is as follows:

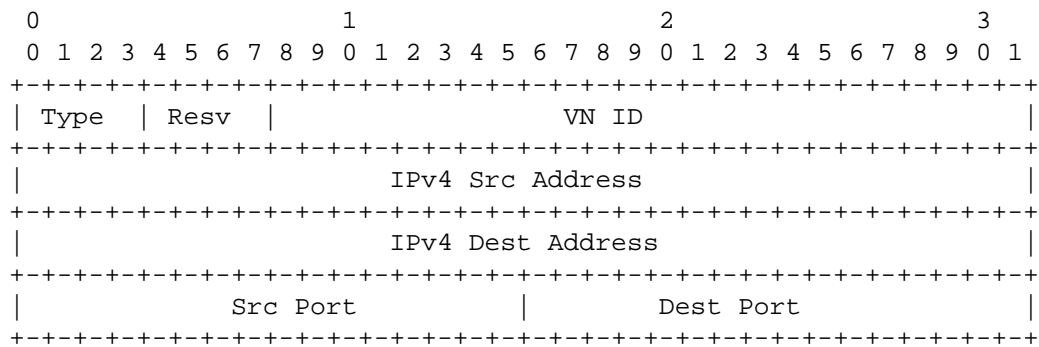


Figure 10. G-SID for IPv4 VXLAN Tunnel

Where,

- o Resv: It is the reserved field and MUST be set as 0.
- o VN ID: 24-bit value for the VN ID of the IPv4 VXLAN tunnel.
- o Src Port: 16-bit value for the source port of the IPv4 VXLAN tunnel.
- o Dest Port: 16-bit value for the destination port of the IPv4 VXLAN tunnel.

The other types of IPv4 tunnel information will be described in the future.

5. Packet Processing

This section describes the packet processing of G-SRH.

5.1. Crossing SRv6 Compression Domains

There are several cases when encoding the SRv6 SID and Compressable SRv6 SID in a single G-SRH.

- o SRv6 path
- o Compressed SRv6 path
- o SRv6 path -> Compressed SRv6 path
- o Compressed SRv6 path -> SRv6 path
- o Combination of the above options.

The ingress node of G-SRv6 encodes the G-SID list in G-SRH based on the type of path and the order of SIDs.

If the beginning of the G-SRv6 path is a Compressed SRv6 sub-path, the CL MUST be initiated accordingly.

If the first Compressable SRv6 SID is inserted in the G-SRH, then the CL MUST be set as 0. If the first Compressable SRv6 SID is reduced, the the CL MUST set to 3, which points to the first C-SID in the first G-SID.

If the beginning of the G-SRv6 path is a SRv6 sub-path, and there is any compressable SRv6 SID in the list, the CL MUST be set to 0.

If the G-SRv6 path is a SRv6 path , then the CL SHOULD be set to 0, and it MUST be ignored in G-SRH processing.

When a compression G-SID is encoded in the G-SRH, new actions should be defined for processing it. When the SL >0, the pseudo code is shown below:

```

01. If ipv6 DA is a normal local SRv6 SID,                                //Ref1
02.     SL- -; DA = SRH [SL];
03.     Process the packet based on the type of SID
04.     Forward the packet to the new DA;
05. if ipv6 DA is a local Compressable SRv6 SID                        //Ref2
06.     If CL = 0
07.         SL- -; CL = 3;
08.         DA[CP..CP+31] = SRH[SL][CL] ;
09.     Else
10.         CL --;
11.         DA[CP..CP+31] = SRH[SL][CL]
12.     Process the packet based on the type of SID
13.     Forward the packet to the new DA;
14. if ipv6 DA is a local Compressable SRv6 SID with EOC flavor //Ref3
15.     SL --
16.     CL = 0
17.     DA = SRH[SL]
18.     Process the packet based on the type of SID
19.     Forward the packet to the new DA
20. else
21.     other G-SIDs processing;

```

- o Ref1: nothing is changed when the IPv6 DA is an SRv6 SID, the SID will be processed as per [\[I-D.ietf-spring-srv6-network-programming\]](#), so G-SRH is compatible with SRH.

- o Ref2: when the IPv6 DA is a local compressable SRv6 SID, the node should process it based on the type of SID and update the IPv6 DA based on the value of CL. If the CL is 0, the next C-SID should be selected from the first C-SID in the next compression G-SID and updated to the IPv6 DA. If the CL is greater than 0, then the next C-SID in this G-SID should be updated to the IPv6 DA.
- o Ref3: If the IPv6 DA is a local Compressable SRv6 SID with EOC flavor, means this is the last C-SID of the C-SIDs list of this compression SRv6 sub-path. The next G-SID should be updated to the IPv6 DA, and the CL is set to 0 in order to help the process of the next Compression SRv6 SID.

When SL = 0, the pseudo code is shown below.

```

01. If ipv6 DA is a local SRv6 SID                                //Ref1
02.     SRv6 processing based on the type of SID
03. If ipv6 DA is a local compressable SID                        //Ref2
04.     If CL = 0
05.         SRv6 processing based on the type of SID
06.     Else
07.         CL --;
08.         DA[CP..CP+31] = SRH[SL][CL];
09.         Process the packet based on the type of SID
09.         Forward the packet to the new DA;
10. If ipv6 DA is a local compressable SID with EOC Flavor      //Ref 3
11.     Process the packet based on the type of SID
12. Else
13.     Other G-SIDs processing;

```

- o Ref1: nothing is changed when the IPv6 DA is an SRv6 SID, the SID will be processed as per [\[I-D.ietf-spring-srv6-network-programming\]](#), so G-SRH is compatible with SRH.
- o Ref2: when the IPv6 DA is a local compressable SRv6 SID, the node should process it based on the type of SID and update the IPv6 DA based on the value of CL. If the CL is 0, meaning this C-SID is the last segment, the node should process the SID and the G-SID processing should be finished. If the CL is greater than 0, the next C-SID in the G-SID should be updated to the IPv6 DA.
- o Ref3: If the IPv6 DA is a local Compressable SRv6 SID with EOC flavor, meaning this is the last C-SID in the C-SIDs list of this compression SRv6 sub-path, the node should process the SID and the G-SID processing should be finished.

5.2. Crossing SR-MPLS Domains

In some scenarios such as SD-WAN, an SRv6 packet may cross MPLS networking. Usually, the packet can be steered into an SR-MPLS policy based on END.BM SID [[I-D.ietf-spring-srv6-network-programming](#)]. For reducing configuration in the middle nodes, this document proposes END.M (End function with SR-MPLS path instantiation) to indicate the SR-MPLS/MPLS path by the MPLS label stack carried after it.

5.2.1. END.M

An End.M SID indicates the beginning of the SR-MPLS SID list/MPLS Label stack, and the S-bit in MPLS label indicates the end of the MPLS label stack.

When a node processes an END.M SID, it copies the SR-MPLS SID list/MPLS labels to the outer MPLS header, and updates the SL pointing to the next G-SID after these MPLS G-SIDs, set the IPv6 DA as the G-SRH[SL], and then forwards the packet based on the active MPLS label.

The pseudo code is shown below:

01. Copy the MPLS labels to MPLS header //Ref1
02. Update the SL point to next G-SID after the MPLS G-SID
03. Set the IPv6 DA as the SRH[SL]
04. Forward the packet based on the active MPLS label

Ref1. The Node will copy the MPLS labels within the MPLS G-SID to the outer MPLS header. The end of the MPLS labels is indicated by the S-bit in the MPLS label.

The MPLS G-SID MUST NOT be the last G-SID in the G-SID list.

5.3. Crossing IPv4 Domains

In some scenarios such as SD-WAN, an SRv6 packet may cross IPv4 networking. Usually, the packet will be transported by an IPv6 over IPv4 tunnel. This document proposes END.4 (End with IPv4 tunnel instantiation) and END.B4 (End binding to an IPv4 tunnel) behaviors for indicating the IPv4 tunnel.

5.3.1. END.4

The End.4 indicates the instantiation of an IPv4 Tunnel, and the parameter of this IPv4 Tunnel is carried by the IPv4 G-SID. An End.4 SID is never the last segment in a G-SID list.

When a node processes a END.4 SID, it will encapsulate an outer IPv4 tunnel header according to the tunnel information in the associated IPv4 G-SID, and set the SL point the next G-SID after the IPv4 G-SIDs, update the inner IPv6 DA by G-SRH[SL], and then forward the packet based on the outer IPv4 DA.

The pseudo code is shown below:

01. Encapsulate the outer IPv4 tunnel header according to the information in IPv4 G-SID(IPv4 DA, SA, etc.) //Ref1
02. Update the SL point to next G-SID after the IPv4 G-SID
03. Set the inner IPv6 DA as the SRH[SL]
04. Forward the packet based on the outer IPv4 DA.

Ref1: When encapsulating the outer IPv4 tunnel header, the source address and destination address and tunnel related parameters are learned from the IPv4 G-SID.

The IPv4 G-SID MUST NOT be the last G-SID in the G-SID list.

5.3.2. END.B4

This is a variation of the End behavior, and it will be bound to an IPv4 tunnel initiated at endpoint node, while the parameters of this tunnel are not carried by SRH but maintained by node.

The parameters of the IPv4 tunnel associated to the END.B4 SID can be configured and advertised in SID instantiation. This is out of scope of this document, and will be described in other documents.

An End.B4 SID is never the last segment in a SID list.

When a node processes an END.B4 SID, it will encapsulate an outer IPv4 tunnel header based on the parameters binding to the End.B4 SID, and set the SL point the next G-SID, update the inner IPv6 DA by G-SRH[SL], and then forward the packet based on the outer IPv4 DA.

The pseudo code is shown below.

01. Encapsulate the outer IPv4 tunnel header according to the parameters binding to the End.B4 SID //Ref1
02. Update the SL point to next G-SID after the IPv4 G-SID
03. Set the inner IPv6 DA as the SRH[SL]
04. Forward the packet based on the outer IPv4 DA.

Ref1: When encapsulating the outer IPv4 tunnel header, the source address and destination address and tunnel related parameters are

learned from the parameters binding to the SID, the information is maintained at the node.

6. IANA Considerations

TBD

7. Security Considerations

TBD

8. Contributors

Zhibo Hu
Huawei Technologies
huzhibo@huawei.com

Yang Xia
Huawei Technologies
yolanda.xia@huawei.com

9. Acknowledgements

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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, [RFC 8200](#), DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-26](#) (work in progress), October 2019.

[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>>.

[I-D.li-spring-compressed-srv6-np]
Li, Z., Li, C., Xie, C., LEE, K., Tian, H., Zhao, F., Guichard, J., Cong, L., and S. Peng, "Compressed SRv6 Network Programming", [draft-li-spring-compressed-srv6-np-02](#) (work in progress), February 2020.

10.2. Informative References

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", [draft-ietf-spring-srv6-network-programming-28](#) (work in progress), December 2020.

[I-D.filsfils-spring-srv6-net-pgm-illustration]
Filsfils, C., Camarillo, P., Li, Z., Matsushima, S., Decraene, B., Steinberg, D., Lebrun, D., Raszuk, R., and J. Leddy, "Illustrations for SRv6 Network Programming", [draft-filsfils-spring-srv6-net-pgm-illustration-03](#) (work in progress), September 2020.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: c.l@huawei.com

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Chongfeng Xie
China Telecom
China Telecom Information Science&Technology Innovation park, Beiqijia Town,Changping District
Beijing
China

Email: xiechf@chinatelecom.cn

Cong Li
China Telecom
China Telecom Information Science&Technology Innovation park, Beiqijia Town,Changping District
Beijing
China

Email: licong@chinatelecom.cn

Hui Tian
CAICT
Beijing
China

Email: tianhui@caict.ac.cn

Feng Zhao
CAICT
Beijing
China

Email: zhaofeng@caict.ac.cn