

SPRING Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 23, 2020

Z. Li  
C. Li  
S. Peng  
Z. Wang  
B. Liu  
Huawei Technologies  
July 22, 2019

**Compressed SRv6 Network Programming  
draft-li-spring-compressed-srv6-np-00**

**Abstract**

Segment Routing can be applied to the IPv6 data plane by leveraging a new type of Routing Extension Header, called Segment Routing Header(SRH). However, the overhead introduced by SRH may be a challenge for the current hardware capability, which would have much effect on the forwarding performance and the payload efficiency.

This document defines a compressed SRv6 network programming mechanism in order to reduce the overhead of SRv6 by introducing the Compressed Segment Identifier(C-SID) and the Compressed SRH(C-SRH). The C-SRH can be a new Routing Header or an enhancement of SRH, which is compatible with SRH well.

**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 January 23, 2020.

**Copyright Notice**

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) 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

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">3</a>
<a href="#">2.1.</a>	<a href="#">Requirements Language</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Compressed SID(C-SID)</a>	<a href="#">3</a>
<a href="#">4.</a>	<a href="#">Compressed Segment Routing Header(C-SRH)</a>	<a href="#">4</a>
<a href="#">5.</a>	<a href="#">C-SRH Processing</a>	<a href="#">7</a>
<a href="#">6.</a>	<a href="#">Illustration</a>	<a href="#">9</a>
<a href="#">6.1.</a>	<a href="#">Reference Diagram</a>	<a href="#">9</a>
<a href="#">6.2.</a>	<a href="#">Compressed SRv6 Forwarding Example</a>	<a href="#">9</a>
<a href="#">7.</a>	<a href="#">Benefits</a>	<a href="#">11</a>
<a href="#">8.</a>	<a href="#">IANA Considerations</a>	<a href="#">13</a>
<a href="#">9.</a>	<a href="#">Security Considerations</a>	<a href="#">13</a>
<a href="#">10.</a>	<a href="#">References</a>	<a href="#">13</a>
<a href="#">10.1.</a>	<a href="#">Normative References</a>	<a href="#">13</a>
<a href="#">10.2.</a>	<a href="#">Informative References</a>	<a href="#">13</a>
	<a href="#">Authors' Addresses</a>	<a href="#">14</a>

## [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 IPv6 data plane, it is called SRv6 [[I-D.ietf-6man-segment-routing-header](#)]. An SRv6 Segment ID(SID) is a 128-bit value, and it can be represented as LOC:FUNCT where LOC is the L most significant bits and FUNCT is the 128-L least significant bits [[I-D.ietf-spring-srv6-network-programming](#)]. L is called the locator length and is flexible. Each operator is free to use the locator length it chooses. The LOC part of the SID is routable and leads to the node which instantiates that SID.

For supporting SR, a new routing header called Segment Routing Header (SRH), which contains a list of SIDs and other information such as Segments Left, 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 in an explicit path.

However, the overhead of SIDs(16 bytes per SID) may be a challenge for the current hardware processing capability. The large size of SRH will have much effect on the forwarding performance. Also, when the packet is small, the payload efficiency is not ideal due to the large overhead of SRH. When the packet is large, the large overhead of SRH may also cause the packet to be dropped due to PMTU [[RFC8200](#)].

This document defines a compressed SRv6 network programming mechanism to order to reduce the overhead of SRv6 by introducing the Compressed Segment Identifier(C-SID) and the Compressed SRH(C-SRH). The C-SRH can be a new Routing Header or an enhancement of SRH, which is compatible with SRH well.

## **2. Terminology**

This document makes use of the terms defined in [[I-D.ietf-6man-segment-routing-header](#)], [[RFC8402](#)] and [[RFC8200](#)]. The reader is assumed to be familiar with the terminology defined in them. This document introduce the following terms:

C-SRH: Compressed Segment Routing Header

C-SID: Compressed Segment Identifier

C-Tag: Compressed Tag

### **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. Compressed SID(C-SID)**

This document defines a Compressed SID(C-SID) to carry the last N bytes of the original SRv6 SID, which is the different part of the SID comparing with other SIDs in the SID List.

N is the length of the common prefix among SIDs in the SID list. The common prefix part contains the common part of Locators in the SID list, while the C-SID contains the different part of Locator and Function ID part of an SRv6 SID.



The IPv6 DA contains a 128-bits(16 Bytes) SRv6 SID, and it can be separated as two parts: the common prefix part among all SIDs, and the different part of a specific SID called C-SID that includes the different part of the locator and the Function ID.

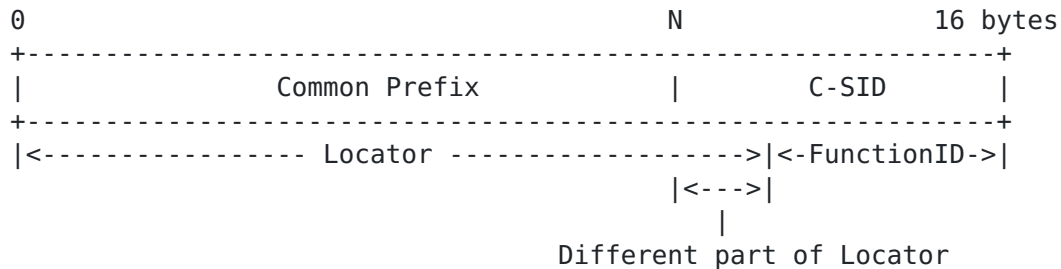


Figure 1. C-SID in IPv6 DA

In this way, the common prefix is carried by the IPv6 DA only, and the SIDs in SID list will not carry the prefix part, but only the different part, the C-SID.

Therefore, this document does not define any new types of the SRv6 Segment.

#### 4. Compressed Segment Routing Header(C-SRH)

In order to carry the C-SID, this document defines the Compressed Segment Routing Header (C-SRH).

The C-SRH can be a new Routing Header which need to introduce a new Routing type or just an enhancement of SRH (Note: the latter is preferred in the document).

The C-SRH provides a more efficient encoding mechanism for SRv6, and it can be compatible with the SRv6 very well.

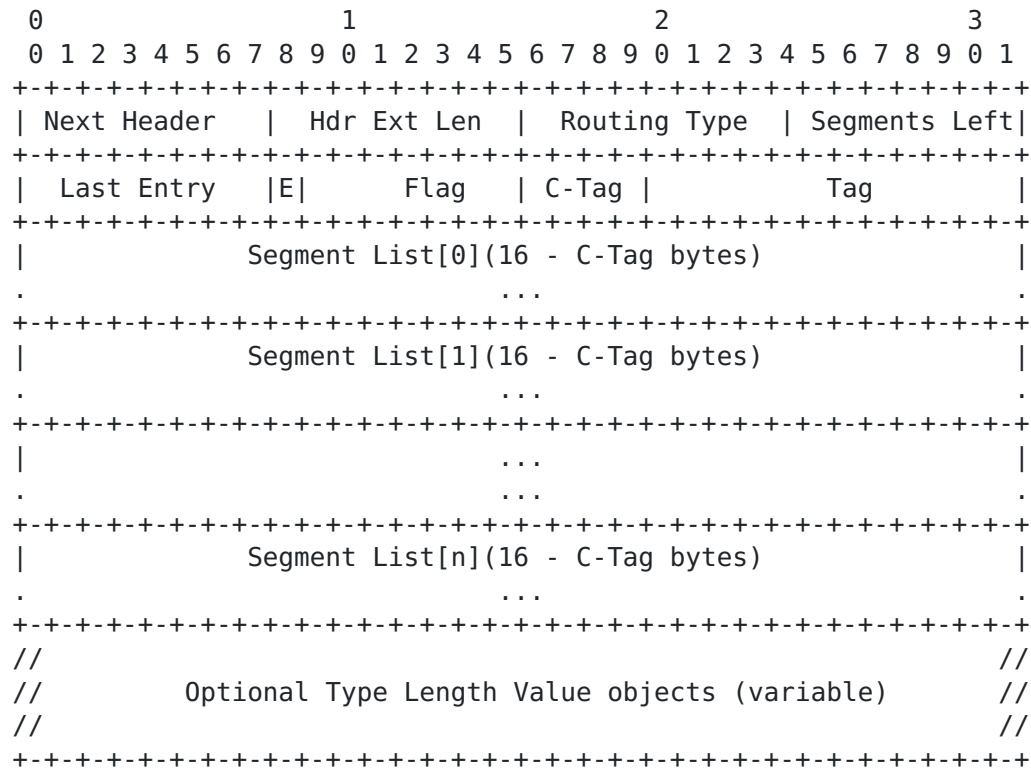


Figure 2. Compressed Segment Routing Header

where:

- o Next Header: Defined in [\[RFC8200\]](#)
- o Hdr Ext Len: Defined in [\[RFC8200\]](#)
- o Routing Type: 4 when C-SRH is an enhancement of SRH, or other type when C-SRH is a new Routing Header.
- o Segments Left: Defined in [\[RFC8200\]](#)
- o Last Entry: contains the index (zero based), in the Segment List, of the last element of the Segment List.
- o Flags:



```

0 1 2 3 4 5 6 7
+--+--+--+--+--+
|E|U U U U U U U|
+--+--+--+--+--+

```

U: Unused and for future use. MUST be 0 on transmission and ignored on receipt.

E: Exclude flag, set when the last SID is excluded in compression.

1: the last SID is excluded in compression, and it is 16 bytes(128 bits) value

0: the last is included in compression, and it is a 16 - C-Tag bytes value

- o C-Tag: 4-bit unsigned integer to indicate the length of the common prefix. Therefore, the length of the C-SID in C-SRH is 16 - C-Tag bytes except the last segment if and only if the E-flag is set. When the C-Tag is 0, means the length of C-SIDs in C-SRH is 16 bytes, which is compatible with the SRH [[I-D.ietf-6man-segment-routing-header](#)].
- o Tag: 12 bits value to tag a packet as part of a class or group of packets, e.g., packets sharing the same set of properties as per [[I-D.ietf-6man-segment-routing-header](#)].
- o Segment List[0]: 16 bytes (128 bits ) IPv6 address when E-flag is set, and last 16 - C-Tag bytes of SID when E-flag is unset.
- o Segment List[n]: a compressed SRv6 SID, which is the last 16 - C-Tag bytes value of the original nth SRv6 SID. The Segment List is encoded starting from the last segment of the SR Policy, i.e., the first element of the segment list (Segment List [0]) contains the last segment of the SR Policy, the second element contains the penultimate segment of the SR Policy and so on.
- o Type Length Value (TLV) are described in [[I-D.ietf-6man-segment-routing-header](#)].

In some use cases, the last SID may be a normal SID, which has a different prefix against all other SIDs, so it can be excluded in C-SID generation for better compression.

The E-flag indicates whether the last SID is excluded in compression. When E-flag is set, Segment List[0] will carry the original SID,



otherwise, it carries the compressed SID, i.e. the last 16 - C-Tag bytes of the original Segment List[0].

## 5. C-SRH Processing

The compressed SID List can be generated by the ingress node itself by comparing the SIDs to get the C-Tag value according to the length of the prefix, and derive the compressed SID list. The compressed SID List can also be generated by the Controller and send to the ingress as well which need to introduce the extra protocol extension. (Note: The former is preferred in the document)

When the ingress node applies SRv6 policy to packets, a C-SRH can be encapsulated in a new IPv6 header(Encapsulation Mode). The first SID is carried in the DA, and the different parts of all SIDs, as known as C-SIDs, are carried in the SID List of C-SRH. The last SID is inserted according to the E-flag.

When an SRv6 endpoint node receives the packets, the node will follow the same processing procedure of SRH, that is, to inspect whether the DA is a local SID or not, if yes, then process the SID according to its function. Otherwise, it will perform regular IPv6 forwarding.

When DA is a local SID, then the node will process the C-SRH and the C-SIDs.

In C-SID processing, the C-SID will be updated to the IPv6 DA to replace the last 16 - C-Tag bytes. Regarding the last SID, if the E-flag is set, the entire 128 bit of Segment List[0] is updated to IPv6 DA. Otherwise, the C-SID will be updated to replace the last 16 - C-Tag bytes of IPv6 DA. After updating the IPv6 DA, the packet will be forwarded accordingly.

The pseudo code of C-SRH processing is shown below.

```
01. When a C-SRH is processed {
02.   If Segments Left is equal to zero {
03.     Proceed to process the next header in the packet,
       whose type is identified by the Next Header field in
       the Routing header.
04.   }
05.   Else {
06.     If local configuration requires TLV processing {
07.       Perform TLV processing
08.     }
09.     If E-flag is set:
10.       max_last_entry = (Hdr Ext Len - 8 - C-Tag)/(16 - C-Tag)
11.     Else:
12.       max_last_entry = (Hdr Ext Len - 8)/(16 - C-Tag)
13.     If ((Last Entry > max_last_entry) or
14.        (Segments Left is greater than (Last Entry+1)) {
15.       Send an ICMP Parameter Problem, Code 0, message to
       the Source Address, pointing to the Segments Left
       field, and discard the packet.
16.     }
17.     Else {
18.       Decrement Segments Left by 1.
19.       if Segments Left > 0 or Segments Left = 0 and E-flag = 0:
20.         // Update the C-SID to the DA
21.         Copy Segment List[Segments Left] from the SRH to
       replace the last 16 - C-Tag bytes of
       destination address of the IPv6 header.
22.       else:
23.         // Segment Left = 0 and E-flag = 1
       // Segment List[0] is a 16 bytes value.
24.         Copy Segment List[Segments Left] from the SRH to
       destination address of the IPv6 header.
25.       If the IPv6 Hop Limit is less than or equal to 1 {
26.         Send an ICMP Time Exceeded -- Hop Limit Exceeded in
       Transit message to the Source Address and discard
       the packet.
27.       }
28.       Else {
29.         Decrement the Hop Limit by 1
30.         Resubmit the packet to the IPv6 module for
       transmission to the new destination.
31.       }
32.     }
33.   }
34. }
```



## 6. Illustration

This section describes a simple example to illustrate the usage of C-SID. Similar to [\[I-D.filsfils-spring-srv6-net-pgm-illustration\]](#), in order to ease the reading of the example, we introduce a simple reference diagram and simplified SID allocations.

### 6.1. Reference Diagram

Nodes 1 - 8 are SRv6 enabled nodes within the network domain.

Nodes CE1, CE2, and CE3 are outside the domain.

CE1 and CE2 are tenants of VPN 100.

Nodes 1 and 8 act as PE respectively to nodes CE1 and CE3.

All the links within the domain have the same IGP metric.

The IGP metric shortest-path from 1 to 8 is 1-2-7-8, while the latency-metric shortest-path from 1 to 8 is 1-2-3-4-5-6-7-8.

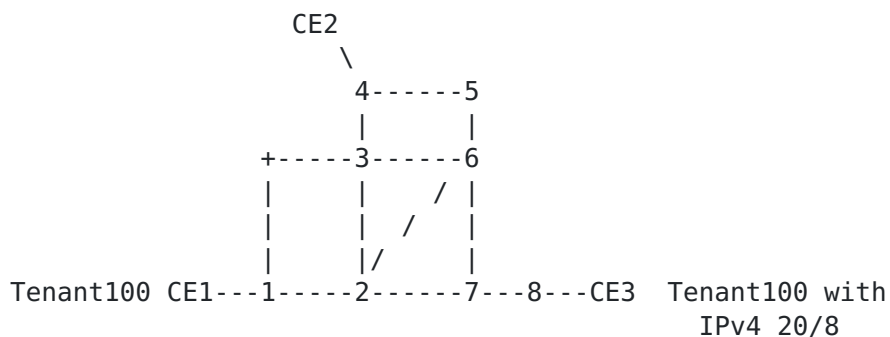


Figure 3: Reference topology

## 6.2. Compressed SRv6 Forwarding Example

This section describes a simple example to show how efficient C-SRH can reduce the overhead of SRv6.

In order to ease the reading of the example, it is better to introduce a simplified SID allocations. We assume:

- o B::/112 is dedicated to the internal SRv6 SID space, which is the common prefix. Therefore the C-SID is a 16-bits value.

- o A locator expressed in 120 bits and a function expressed in 8 bits.
- o Node k has B::k/120 for its local SID space. Its SIDs will be explicitly allocated from that block.
- o Node k advertises B::k/120 in its IGP.
- o Function ::1 (function 1, for short) represents the End function with PSP support.
- o B::k::1 represents the End function with PSP support allocated by node K, such as B::6::1 represents the End function with PSP support allocated by node 6.
- o B::8::D100 is an END.DT4 SID initiated by node 8, which is associated with the VRF100.

In SRH based SRv6, the PE 1 encapsulates the packets (CE1, CE3) from CE1 to CE3 in an outer IPv6 header with DA = B::3::1 and SRH (B::8::D100, B::7::1, B::6::1, B::5::1, B::4::1, B::3::1, B::2::1; SL=6; NH=4).

<B::2::1, B::3::1, B::4::1, B::5::1, B::6::1, B::7::1, B::8::D100> follows the latency-metric shortest-path. The total length of SRH is  $8+16*7=120$  bytes.

In Compressed SRv6, PE 1 encapsulates (CE1, CE3) in an outer IPv6 header with DA = B::2::1 and C-SRH (B::8::D100, 7::1, 6::1, 5::1, 4::1, 3::1, 2::1, SL=6; NH=4) with E-flag set. The C-Tag is 14, since the length of same prefix is 112 bits. Therefore, the total length of C-SRH is  $8 + (16-14)*6 + 16 = 36$  bytes, then 84 bytes are reduced meaning 70% size of the SRv6 overhead or 87.5% of SIDs (except the last SID) overhead are reduced.

The packet leaves node 1 to node 2 according to the FIB associated with the IPv6 DA B::2::1. The packet leaves node 1 can be present as

```
(A::1, B::2::1)
(B::8::D100, 7::1, 6::1, 5::1, 4::1, 3::1, 2::1, SL=6; NH=4)
(X, Y)
```

When 2 receives the packet, 2 matches B::2::1 in its "My SID Table" and executes the END function behavior to update the IPv6 DA. Since the updated SL is greater than 0, and the C-Tag is 14, then it copies the C-SID that is a 2 bytes value to replace the last 2 bytes of IPv6 DA, and then forward the packet according the new IPv6 DA B::3::1. The packet leaves node 2 can be present as



```
(A::1, B::3::1)
(B::8::D100, 7::1, 6::1, 5::1, 4::1, 3::1, 2::1, SL=5; NH=4)
(X, Y)
```

Similar to node 2, the node 3, 4, 5, and 6 performs the END function behavior to update the IPv6 DA with the corresponding C-SID and then forward the packet by looking up FIB accordingly. Therefore, the packet leaves node 6 can be present as

```
(A::1, B::7::1)
(B::8::D100, 7::1, 6::1, 5::1, 4::1, 3::1, 2::1, SL=1; NH=4)
(X, Y)
```

When 7 receives the packet, 7 matches B::7::1 in its "My SID Table" and executes the END function behavior to update the IPv6 DA. Since the updated SL is 0 and E-flag is set, then the 128-bits Segment List[0] is copied to the IPv6 DA. Also, the C-SRH is popped since the B::7::1 is an END SID with PSP flavor. Node 7 then performs a lookup on the updated IPv6 DA B::8::D100 to forward the packet along the shortest path to node 8. The packet leaves node 8 can be present as

```
(A::1, B::8::D100)
(X, Y)
```

When 8 receives the packet, 8 matches B::8::D100 in its "My SID Table" and executes the END.DT4 function behavior to sends the IP packet (CE1, CE3) to its VPN destination.

This example illustrates the procedure of C-SRH based SRv6 forwarding, it shows that the longer prefix can reduce more overhead of SRv6. More benefits are described in the [section 7](#).

## 7. Benefits

1. Seamless integration with SRv6 Network Programming
  - o No new type(Functions, such as END) of SRv6 SIDs is defined. A C-SID is a sub-set of an SRv6 SID.
  - o Neither redefines the IPv6 address space nor requires any specific IPv6 space.
2. Supporting Full Set Functionalities of SRv6
  - o Full set functionalities of SRv6(BE, Loose TE and strict TE, etc.) are supported without any extra routes advertisements.



- o Function ID information is maintained.

### 3. Control-Plane friendly

- o No need to make any extensions in Control-Plane to advertise new type of SIDs or binding information.
- o No indexed mapping table is required
- o No routing extension is required.
- o No new routes advertisement is required if without new Locators

### 4. Hardware-friendly

- o Hardware has the mature capability to overwrite the IPv6 DA.
- o Avoids any extra lookup in indexed mapping table

### 5. Efficient MTU overhead

- o C-SRH has the smallest MTU overhead among alternative solutions (VxLAN with SR-MPLS, CRH, uSID), When all the Segment endpoint nodes information is maintained in the packet.

### 6. Scalable SR TE

- o 8 C-SIDs can be carried in 128 bits when C-SID is 16 bits value
- o 16 Segment endpoint nodes (1 in DA and 16 in C-SRH including the one in DA) in 40 bytes of overhead.
  - o T.Encaps with a C-SRH of 40 bytes (8 fixed + 2 \* 16 bytes)
  - o ALL C-SIDs are maintained in C-SRH, which can be used for recording the explicit routing path.

### 7. Saving IPv6 address

- o Very limited IPv6 address are needed for SID space. Longer Common Prefix means smaller IPv6 address burning and smaller overhead of SRv6.
- o Very easy to meet the requirement of C-SRH since any operator or person can offer a 112/, 80/ or even 64/ prefix.

## **8. IANA Considerations**

TBD

## **9. Security Considerations**

TBD

## **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. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-21](#) (work in progress), June 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>>.

### **10.2. Informative References**

- [I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6 Network Programming", [draft-ietf-spring-srv6-network-programming-01](#) (work in progress), July 2019.

[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-00](#) (work  
in progress), February 2019.

#### 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: chengli13@huawei.com

Shuping Peng  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Rd.  
Beijing 100095  
China

Email: pengshuping@huawei.com

Zhongzheng Wang  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Rd.  
Beijing 100095  
China

Email: wangzhongzhen@huawei.com

Bing Liu  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Rd.  
Beijing 100095  
China

Email: [remy.liubing@huawei.com](mailto:remy.liubing@huawei.com)