L H. Jie
Internet-Draft W. Huilai
Intended status: Standards Track ZTE Corporation
Expires: July 21, 2021 January 17, 2021

An Enhanced Source Routing Header Based on RH3
draft-han-6man-enhanced-source-routing-header-02

Abstract

IPv6 Routing header type 3 (termed as RH3) is defined and used in Low-Power and Lossy Networks (LLNs) that are typically constrained in resources (limited communication data rate, processing power, energy capacity, memory). Based on the mechanisms introduced by RH3, this document specifies an new IPv6 Routing Header type that provides encapsulation capability of segments with various lengths and can be applied to more scenarios.

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 July 21, 2021.

Table of Contents

1. Introduction

Routing headers are defined in [RFC8200]. [RFC6554] specifies the Source Routing Header (SRH), i.e., IPv6 Routing header type 3 (termed as RH3), for use strictly between RPL routers in the Low-Power and Lossy Networks (LLNs) [RFC6550], which are typically constrained in resources (limited communication data rate, processing power, energy capacity, memory). It introduces mechanisms to compact the source route entries when all entries share the same prefix with the IPv6 Destination Address of a packet carrying an RH3, a typical scenario in LLNs using source routing. The compaction mechanism reduces consumption of scarce resources such as channel capacity. However, it is challenging when all entries do not have the same prefix, for example, only a part of entries share the same prefix, but still want to encode all routries in a single routing header and reduce the overhead.

This document specifies an enhanced Source Routing Header (termed as E-SRH) to enhance the encapsulation capability of segments with various lengths and different types, and attempt to apply to more scenarios that using source routing mechanism.

2. 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. Format of the Source Routing Header

The Source Routing Header defined in [RFC6554] has the following format:

Figure 1: Source Routing Header format

Where CmprI, CmprE, and Pad fields allow compaction of the Address[1..n] vector when all entries share the same prefix as the IPv6 Destination Address field of the packet carrying the RH3. The CmprI and CmprE fields indicate the number of prefix octets that are shared with the IPv6 Destination Address of the packet carrying the RH3. The shared prefix octets are not carried within the Routing header and each entry in Address[1..n-1] has size (16 - CmprI) octets and Address[n] has size (16 - CmprE) octets.

4. Format of the Enhanced Source Routing Header

To provide the encapsulation capability of segments with various lengths, this section defines the Enhanced Source Routing Header (E-SRH). The basic idea is to place the CmprI or CmprE information with each segment element. The E-SRH has the following format:

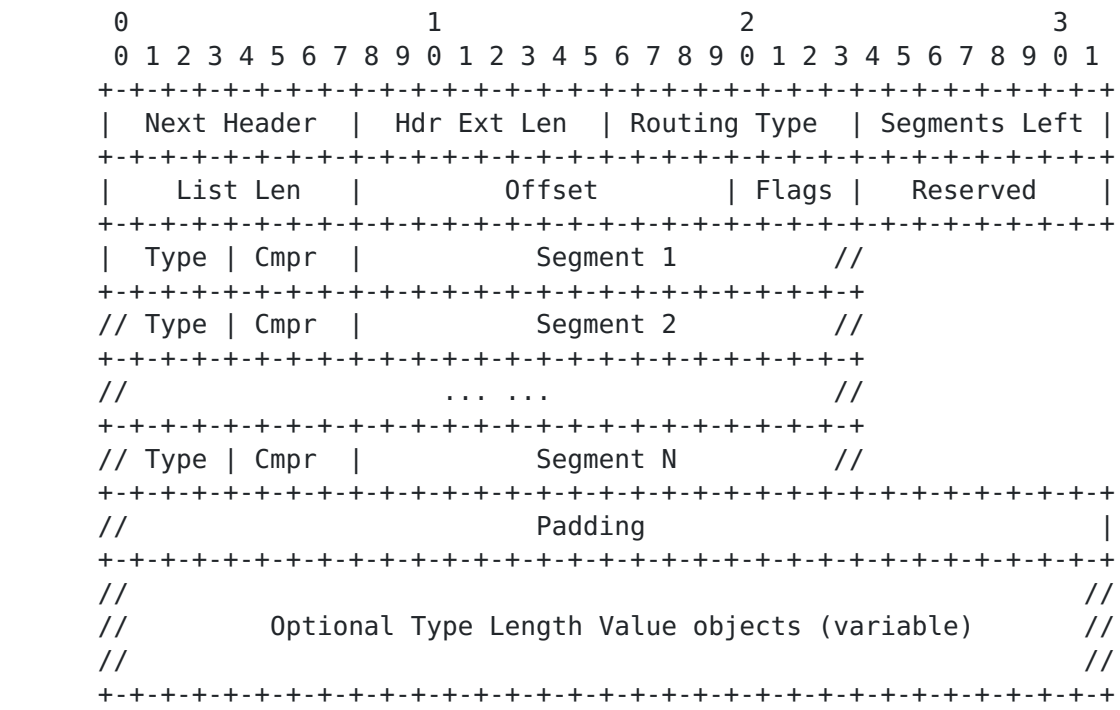

Figure 2: Enhanced Source Routing Header format

where:

Next Header: Defined in [RFC8200], Section 4.4. It identifies the type of header immediately following the Routing header.

Hdr Ext Len: Defined in [RFC8200], Section 4.4. It represents the length of the Routing header in 8-octet units, not including the first 8 octets.

Routing Type: TBA.

Segments Left: Defined in [RFC8200], Section 4.4. It represents the number of route segments remaining, i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination. Note that it represents the count of intermediate nodes, instead of the count of segments always in constant 128-bit units in the routing header. Both this document and [RFC6554] may take segments less than 128 bits, for example, 32 bits, in the routing header.

List Len: 8-bit unsigned integer. It represents the length of the Segment List field in 8-octet units. Note that the size of the entire Segment List must be aligned to 8 bytes. For this purpose, it

may be necessary to pad meaningless zeros after the last segment (i.e., segment N). List Len field must be less than Hdr Ext Len, or equal to Hdr Ext Len when E-SRH does not contain optional TLVs.

Offset: 12-bit unsigned integer. It represents the index of currently active segment in segment list, when the segment list contained in E-SRH is regarded as an array in bytes. When an element in a segment list array is accessed according to Offset field, its access range must not exceed the range represented by List Len * 8.

Flags: 4 bits of flags. No flags are defined currently.

Reserved: This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Each <Type, Cmpr, Segment> tuple provide the information of each segment element in the segment list, and it has variable length.

Type: 4 bits. It represents the type of the segment. The following types are defined:

0: Indicates that the corresponding Segment field is a complete IPv6 address. Note that the corresponding Segment field does not necessarily occupy 16 bytes, and its length is given by Cmpr field. This is because the low-order position of many IPv6 addresses is 0, and it is not necessary to store the entire 16 bytes in the Segment field. The IPv6 address may be a normal address that identify node or interface, or an SRv6 SID ([RFC8402]) that has further functions.

1: Indicates that the corresponding Segment field is 1-octet of IPv6 address fragment. Similar with [RFC6554], the value of Segment field can be stiched with the prefix contained in the IPv6 Destination Address to form a complete IPv6 address. The Cmpr field indicates the number of prefix octets that are shared with the IPv6 Destination Address. For the stiched IPv6 address, the high-order position is the prefix, immediately following the value of the Segment field, and the low-order position is filled with zeros.

2: Indicates that the corresponding Segment field is 2-octet of IPv6 address fragment. The stiching method is the same as Type-1.

3: Indicates that the corresponding Segment field is 3-octet of IPv6 address fragment. The stiching method is the same as Type-1.

4: Indicates that the corresponding Segment field is 4-octet of IPv6 address fragment. The stiching method is the same as Type-1.

5: Indicates that the corresponding Segment field is 5-octet of IPv6 address fragment. The stiching method is the same as Type-1.

6: Indicates that the corresponding Segment field is 6-octet of IPv6 address fragment. The stiching method is the same as Type-1.

7: Indicates that the corresponding Segment field is 7-octet of IPv6 address fragment. The stiching method is the same as Type-1.

8: Indicates that the corresponding Segment field is 8-octet of IPv6 address fragment. The stiching method is the same as Type-1.

9: Indicates that the corresponding Segment field is 3-octet of MPLS Label. The MPLS Label can be mapped to a complete IPv6 address.

10: Indicates that the corresponding Segment field is 4 bytes of SR-MPLS SID index([RFC8402]). The index can be mapped to a complete IPv6 address.

11: Indicates that the corresponding Segment field is 4 bytes of BIER index([RFC8279]). The index can be mapped to a complete IPv6 address.

15: Indicates that the corresponding Segment field is an argument that is used by the previous Segment element in E-SRH. The segment's length is given by Cmpr field. Note that Segment with argument types is not counted in the count represented by the segment left field.

Cmpr: 4 bits. It represents the length of the prefix in octet units for Type-1 to Type-8. For Type-0 and Type-15, it represents the actual length of the Segment field. For Type-9, Type-10 and Type-11, it has no meaning and can be set to 0.

Segment: Represents the nth segment in the Segment List. Similar with [RFC6554], the Segment List is encoded in E-SRH in positive order. The Segment field has variable length.

Padding: Optional padding field immediately following Segment N field. It is used to pad the Segment List to a multiple of 8 octets. If the Segment List is already 8-byte aligned, there is no need to have a Padding field.

Optional TLVs: To be defined in future.

5. Generating E-SRH

For a segment list <S1, S2, S3,..., Sn>, the headend can encode it in E-SRH. S1 is placed to DA field of IPv6 header, and S1 to Sn can be also placed in the Segment 1 to Segment N fields respectively. Because S1 is also placed in the Segment 1 field, Offset field needs to be set to the value that point to <Type, Cmpr, Segment 2> tuple, i.e., Offset = sizeof <Type, Cmpr, Segment 1>. In addition, Segment Left field is set to n-1, which means there are n-1 Segments left to be processed in the Segment List.

For the above segment list <S1, S2, S3,..., Sn>, the headend may not place S1 in E-SRH again, then E-SRH only needs to contain n-1 segments. In this case, S2 to Sn will be placed in the Segment 1 to Segment N-1 fields respectively. Offset needs to be set to the value that point to <Type, Cmpr, Segment 1>, i.e., Offset = 0. In addition, Segment Left field is still set to n-1, which means there are n-1 segments left to be processed in the Segment List.

6. Processing E-SRH

E-SRH is examined and processed when the IPv6 packet reaches the node identified in the Destination Address field of the IPv6 header. In that node, dispatching on the Next Header field of the immediately preceding header causes the Routing header module to be invoked.

The following describes the algorithm performed when processing an E-SRH:

```
  If next argument item is needed in the current segment processing {
    Read the next <Type, Cmpr, Segment> tuple which is pointed by
    current Offset field:
      Type 15(Arg), Segment's length is determined by Cmpr field;

    If Type is not 15 {
      Send an ICMP Parameter Problem, Code 0, message to the Source
      Address, pointing to the Offset field, and discard the packet;
    }

    Offset += sizeof (next <Type, Cmpr, Segment>);
    if Offset > List Len * 8 {
      Send an ICMP Parameter Problem, Code 0, message to the Source
      Address, pointing to the Offset field, and discard the packet;
    }

    Continues the remaining processing of the current element with
    additional aruguments (note: long argument if necessary, instead
    of using multiple arguments);
```

```
    }

    if Segments Left = 0 {
      proceed to process the next header in the packet, whose type is
      identified by the Next Header field in the Routing header;
    }
    else {
      Read the next <Type, Cmpr, Segment> tuple which is pointed by
      current Offset field, in detailed:
        For Type 0, Segment's length is determined by Cmpr field;
        For Type 1-8, Segment's length is 1-8 bytes respectively;
        For Type 9(MPLS), Segment's length is 3 bytes;
        For Type 10-11(index), Segment's length is 4 bytes;

      If Type is 15 or unknown {
        Send an ICMP Parameter Problem, Code 0, message to the Source
        Address, pointing to the Offset field, and discard the packet;
      }

      Offset += sizeof (next <Type, Cmpr, Segment>);

      if Offset > List Len * 8 {
        Send an ICMP Parameter Problem, Code 0, message to the Source
        Address, pointing to the Offset field, and discard the packet;
      }

      Decrement Segments Left by 1;
      Copy the stiched/mapped IPv6 address to the destination address
      of the IPv6 header;

      if the IPv6 Hop Limit is less than or equal to 1 {
        send an ICMP Time Exceeded -- Hop Limit Exceeded in Transit
        message to the Source Address and discard the packet;
      }
      else {
        decrement the IPv6 Hop Limit by 1;
        resubmit the packet to the IPv6 module for transmission to the
        new destination;
      }
    }
```

7. Deployment Considerations

This section analyzes some deployment considerations faced by E-SRH.

7.1. Heterogeneous Segment List

E-SRH can support a combination of various type of segments in a single Segment List to reduce encapsulation size. This can be used in scenarios that an E2E Segment List may across multiple domains and each domain has different segment encapsulation style.

It is important for the border node that connected multiple domains to use Segment sharing the same prefix with each domain as much as possible, otherwise, a Segment that is 128 bit of IPv6 address may be inserted in the Segment List, to provide new prefix information for the next domain, or the new prefix information can also be get from a local segment FIB entry.

7.2. Independent Arguments in E-SRH

E-SRH can store independent elements as argument that has local meaning and is used by the the immediately preceding segment. Because the argument type of segment has only local meaning, it does not consume the global address resources. This is very helpful for supporting a large number of services in some networks with limited address space.

Note that in E-SRH a Segment can also directly contain the argument with it, i.e, in the same field. How to distinguish where the argument exist depends on the behavior of the Segment.

7.3. ICMP Error Processing

The invoking packet in the ICMP error message may contain an E-SRH. Since the destination address of a packet with an E-SRH changes as each segment is processed, it may not be the destination used by the socket or application that generated 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. The following logic is used to determine the destination address for use by protocol-error handlers.

Walk all extension headers of the invoking IPv6 packet to the routing extension header preceding the upper-layer header.

If routing header is type E-SRH

Walk to the Segment List[N] (should not be an argument or index) which may be used as the destination address of the invoking packet.

7.4. Repair Path

The mechanism defined in [I-D.ietf-rtgwg-segment-routing-ti-lfa] can be used to provide protection of nodes and links within an IGP area. The repair path can be representd as a Segment List which can be encoded in a separate E-SRH. The Point of Local Repair (PLR) maybe the headend or midpoint of the original path. According to [RFC8200], E-SRH must not be processed, inserted, or deleted by any node along a packet's delivery path, until the packet reaches the node identified in the Destination Address field of the IPv6 header. Thus, for midpoint an outer IPv6 header with its own E-SRH need to be added to the received IPv6 packet, when the packet is delivered along a repair path. However, it is possible for headend to encode multiple E-SRHs within a single IPv6 header when it initiates a packet that is delivered along a repair pat. In any case, when the outer E-SRH is finished, the inner E-SRH is continued to be processed. Whether it's the inner E-SRH or the outer E-SRH, in order to achieve better compression efficiency, the first Segment (similar to RH3) is compressed as much as possible.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] also described in some cases when the failure node or link is just the active segment, and the method is to ignore the active segment and continue to get next segment to keep as much as possible the traffic on the pre-failure path. This mechanism is also called midpoint protection. E-SRH can do this easily by simply reading the next <Type, Cmpr, Segment> tuple, or the next-next and so on, from the list. The FIB entry corresponding to the active segment should indicate whether it can be bypassed.

7.5. Binding to Outer Segment List

Any nodes (headend or midpoint) along the original Segment List may also bind the packets to an outer Segment List based on local policy. For example, the headend may steer packets along an outer Segment List to the first Segment, and multiple E-SRHs may be encoded within a single IPv6 header in the initated packets. A midpoint may also steer packets along an outer Segment List to the next Segment, and an outer IPv6 header with its own E-SRH need to be added to the received IPv6 packet. In any case, when the outer E-SRH is finished, the inner E-SRH is continued to be processed.

7.6. In-situ OAM

OAM and PM information from the segment endpoints can be piggybacked in the data packet. The OAM and PM information piggybacking in the data packets is also known as In-situ OAM (IOAM). IOAM records operational and telemetry information in the data packet while the

packet traverses a path between two points in the network. [I-D.ietf-ippm-ioam-data] defines the IOAM data, and [I-D.ietf-ippm-ioam-ipv6-options] defines how to carry IOAM data in "option data" fields using two types of extension headers in IPv6 packets - either Hop-by-Hop Options header or Destination options header. Next version of this document will define how IOAM data fields are transported as part of E-SRH.

E-SRH complies with [RFC8200], and the Segment Left field strictly represents the number of explicitly listed intermediate nodes still to be visited before reaching the final destination. Thus, Segment Left can be used as local index at which it is expected to record endpoint's data in the IOAM data space.

7.7. NAT Service Funtion

A NAT service funtion SR-MPLS or SRv6 SID can be encoded in the Segment List field of the E-SRH. The related NAT Service entity may modify the destination address in the packets they process. At the endpoint that associated with NAT service entity, the updated Destination Address need to be copied back into the Segment List[N] (i.e, the final destination) of the E-SRH. Although E-SRH contains compressed Segments, it can walk to the last <Type, Cmpr, Segment> tuple and write the updated address. Note that in this case the type of last Segment must not be an argument or index (therwise, the previous segment should be get), and must share prefix with the updated address.

7.8. Operation and Maintenance

To be convenient for operation and maintenance, E-SRH provide necessary information to enable offline tools to parse the IPv6 header and analysis Segment List, without the help of states of the control plane. No matter how the state of the control plane vibrates and changes, for example, the purpose of a Segment may be changed from service A to service B, but the parsing of the packet itself is stable and not affected. This is very convenient for network debugging, operators can clearly know the specific Segment List information encapsulated in E-SRH, check problems and locate them.

7.9. Upper-Layer Checksums

[RFC8200] specifies that any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6, to include the 128-bit IPv6 addresses. If the IPv6 packet contains a Routing header, the Destination Address used in the pseudo-header is that of the final destination. At the originating node, that address will be

in the last element of the Routing header; at the recipient(s), that address will be in the Destination Address field of the IPv6 header.

For E-SRH, at the originating node, the Destination Address used in the pseudo-header will be the last non-Arg segment that is before compression; at the recipient(s), that address will be in the Destination Address field of the IPv6 header.

8. Security Considerations

The E-SRH domain is treated as a trusted domain, and the nodes outside the E-SRH domain are not trusted. This is enforced by two levels of access control lists:

- On the ingress of E-SRH domain, any packet entering the E-SRH domain and destined to an IPv6 address within the E-SRH domain, is dropped.

- On the transit or egress of E-SRH domain, any packet with a destination address within the E-SRH domain but the source address not within, is dropped.

It will block the attacks documented in [RFC5095] from outside the E-SRH domain, including bypassing filtering devices, reaching otherwise-unreachable Internet systems, network topology discovery, bandwidth exhaustion, and defeating anycast.

Additionally, domains deny traffic with spoofed addresses by implementing the recommendations in BCP 84 [RFC3704].

[RFC6554] requires RPL routers to check for loops in the SRH and drop datagrams that contain such loops. However, for the flexibility of Segment List programming for any scenario, E-SRH doesn't do this check, but relevant security mechanisms to avoid tampering with Segment List should be adopted, such as HMAC mechanism introduced in [RFC8754].

The generation of ICMPv6 error messages may be used to attempt denial-of-service attacks by sending an error-causing E-SRH in back-to-back datagrams. An implementation that correctly follows Section 2.4 of [RFC4443] would be protected by the ICMPv6 rate-limiting mechanism.

9. Acknowledgements

TBD

10. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-04 (work in progress), November 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[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>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <https://www.rfc-editor.org/info/rfc6554>.

[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>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[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>.

Authors' Addresses

Han Jie
ZTE Corporation
China

Email: han.jie1@zte.com.cn

Wang Huilai
ZTE Corporation
China

Email: wang.huilai@zte.com.cn