Network
Internet-Draft
Intended status: Standards Track
Expires: July 9, 2021

C. Weiqiang
China Mobile
G. Mirsky
ZTE Corp.
P. Shaofu
L. Aihua
ZTE Corporation
G. Mishra
Verizon Inc.

January 5, 2021

Unified Identifier in IPv6 Segment Routing Networks
draft-mirsky-6man-unified-id-sr-08

Abstract

Segment Routing architecture leverages the paradigm of source routing. It can be realized in a network data plane by prepending the packet with a list of instructions, a.k.a. segments. A segment can be encoded as a Multi-Protocol Label Switching (MPLS) label, IPv4 address, or IPv6 address. Segment Routing can be applied in MPLS data plane by encoding segments in the MPLS label stack. It also can be applied to IPv6 data plane by encoding a list of segment identifiers in IPv6 Segment Routing Extension Header (SRH). This document extends the use of the SRH to unified segment identifiers encoded, for example, as MPLS label or IPv4 address, to compress the SRH, and support more detailed network programming and interworking between SR-MPLS and SRv6 domains.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing architecture [RFC8402] leverages the paradigm of source routing. It can be realized in a network data plane by prepending the packet with a list of instructions, a.k.a. segment identifiers (SIDs). A segment can be encoded as a Multi-Protocol Label Switching (MPLS) label, IPv4 address, or IPv6 address. Segment Routing can be applied in MPLS data plane by encoding 20-bits SIDs in MPLS label stack [RFC8660]. It also can be applied to IPv6 data plane by encoding a list of 128-bits SIDs in IPv6 Segment Routing Extension Header (SRH) [RFC8754].

This document extends the use of the SRH [RFC8754] to unified identifiers encoded as MPLS label or IPv4 address to support more detailed network programming and interworking between SR-MPLS and SRv6 domains.

1.1. Conventions used in this document

1.1.1. Acronyms

SR: Segment Routing

SRH: Segment Routing Extension Header

MPLS: Multiprotocol Label Switching

SR-MPLS: Segment Routing using MPLS data plane

SID: Segment Identifier

IGP: Interior Gateway Protocol

DA: Destination Address

ILM: Incoming Label Map

FEC: Forwarding Equivalence Class

FTN: FEC-to-NHLFE map

OAM: Operation, Administration and Maintenance

TE: Traffic Engineering

SRv6: Segment Routing in IPv6

U-SID: Unified Segment Identifier

PSP: Penultimate Segment Popping

FIB: Forwarding Information Base

UET: U-SID Encapsulation Type

UEC: U-SID Encapsulation Capability

1.1.2. 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.

2. Segment Routing Extension Header: Benefits and Challenges

Many functions related to Operation, Administration and Maintenance (OAM) require identification of the SR tunnel ingress and the path, constructed by segments, between the ingress and the egress SR nodes. Combination of IPv6 encapsulation [RFC8200] and SRH [RFC8754], referred to as SRv6, comply with these requirements while it is challenging when applying SR in MPLS networks, also referred to as SR-MPLS.

On the other hand, the size of IPv6 SID presents a scaling challenge to use topological instructions that define strict explicit traffic-engineered (TE) path or support network programming in combination with service-based instructions. At the same time, that is where SR-MPLS approach provides better results due to the smaller SID length. It can be used to compress the SRv6 header size when a smaller namespace of available SIDs is sufficient for addressing the particular network.

SR-MPLS is broadly used in metro networks. With the gradual deployment of SRv6 in the core networks, supporting interworking between SR-MPLS and SRv6 becomes necessary for operators. t is operationally more efficient and straightforward if SRv6 can use the same size SIDs as in SR-MPLS. The SRH can be extended to define the same as in SR-MPLS SID length to support the unified segment identifier (U-SID). As a result, end-to-end SR tunnel may use U-SIDs across SR-MPLS and SRv6 domains.

3. Unified SIDs in IPv6 Segment Routing Extension Header

SRH format has been defined in Section 3 of [RFC8754] as presented in Figure 1

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[0] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                              ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Segment List[n] (128 bits IPv6 address)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional Type Length Value objects (variable)       //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SRH format

This document defines a new field Size in the SRH Flags field as a two-bits field, termed as UET (U-SID Encapsulation Type) flag, to indicate which type of SIDs are encoded in SRH. The UET flag has the following values:

0b00 - indicate a 128-bits SID, an IPv6 address, termed as UET-128 U-SID.

0b01 - indicate a 32-bits SID, termed as UET-32 U-SID. In some environments, the context could be of IPv4 address, while in some other cases, it could represent an index of list or range of IPv4/ IPv6 addresses. Another interpretation of 32-bits SID could be as

a complementary element of an IPv4/IPv6 prefix. The setting of the interpretation might be made through the control plane based signaling and is outside the scope of this document. If this SID represents a complementary part of an IPv4/IPv6 prefix, the original IP address can be re-constructed by using, for example, mapping, stitching, shifting or translating operation. Specification of such a mechanism is outside the scope of this document.

0b10 - indicate a 32-bits SID, termed as UET-MPLS U-SID, which includes an MPLS label in the leftmost 20-bits as displayed in Figure 2. Information in the Context field could be interpreted as a flavor of a particular network programming behavior. Specification of the network programming using this type of U-SID is outside the scope of this document. [Ed.note. Replace with reference to the U-SID network programming document.]

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                MPLS Label                 |      Context      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Format of Unified SID with MPLS Label

0b11 - indicate a 16-bits SID, termed as UET-16 U-SID. It is similar to 32bits SID and suitable for scenes with higher compression efficiency

This document also introduces a compatible operation on Segment Left field, also termed as SRH.SL. The relationship between the value of SRH.UET and the interpretation of the SRH.SL is as follows:

- o if SRH.UET Flag is UET-128, SRH.SL represents the count of 128bits-SID entries in SRH;

- o if SRH.UET Flag is UET-32 or UET-MPLS, SRH.SL represents the count of 32bits-SID entries in SRH;

- o if SRH.UET Flag is UET-16, SRH.SL represents the count of 16bits-SID entries in SRH.

4. Operations with Unified Segment Identifier

When SRH is used to include 32-bits long U-SIDs, the ingress and transit nodes of an SR tunnel act as described in Section 5.1 and Section 5.2 of [RFC8754] respectively.

4.1. Procedures of 32bits MPLS Label within SRH

This section describes how the UET-MPLS type of U-SID is used to encode a compressed SRH. In this case, an ILM (Incoming Label Map) entry can be used to map a U-SID to an IPv6 address. As a result, it is not necessary to introduce a new type of index-based mapping table. For ILM entry of Adjacency-SID, the mapping result copied to DA (Destination Address) is the remote interface IPv6 address, for ILM entry of Node-SID, the mapping result that is copied into DA is a remote node loopback IPv6 address.

Operations on an MPLS label of U-SID type are the same as those defined in [RFC8663]. However, SR-MPLS over SRH has the following advantages compared with SR-MPLS over UDP:

- o SRH is flexible to extend flags or sub-TLVs for service requirements, but UDP not.

- o Labels in SRH can meet 8 bytes alignment requirements as per [RFC8200], but UDP not.

- o The source address and the complete path information of the SR policy are not discarded, but UDP not.

- o The forwarding performance of SR-MPLS over SRH is better than the UDP method because it only updates the destination address rather than frequently removing and adding outer headers.

Procedures of SR-MPLS over IP of [RFC8663] described how to construct an adjusted SR-MPLS FTN (FEC-to-NHLFE map) and ILM entry towards a prefix-SID when next-hops are IP-only routers. The action of FTN and ILM entry will steer the packet along an outer tunnel to the destination node that has originated the FEC (Forwarding Equivalence Class). UDP header is removed and put again at each segment endpoint. However, for SR-MPLS over SRH in this document, we don't try to depend on that adjusted FIB (Forwarding Information Base) entry, because there are not any actions needed to get from the FIB entry, a traditional ILM entry (maybe without out-label because of IP-only next-hop) is enough to get the FEC information, i.e., to map a U-SID to an IPv6 address and copy to DA. Note that an implementation can get FEC and next-hop/interface forwarding information from the ILM entry, avoiding extra FIB lookup. An SRv6 policy chosen to encapsulate the U-SID list within SRH is determined at the ingress node of this SRv6 policy, SRH is preserved along the SR to egress, though PSP (Penultimate Segment Popping) may be used, that is different from SR-MPLS over IP/UDP method [RFC8663], so the source address (i.e., the ingress of the SRv6 policy) is not discarded.

4.1.1. Packet Forwarding Based on UET-MPLS U-SID

The packet forwarding based on UET-MPLS U-SID is similar to the processing described in [RFC8663]. But it differs from that in FIB action and segment list processing. For completeness, we repeat the description of [RFC8663] with modification as follows.

```
      +-----+       +-----+       +-----+        +-----+        +-----+
      |  A  +-------+  B  +-------+  C  +--------+  D  +--------+  H  |
      +-----+       +--+--+       +--+--+        +--+--+        +-----+
                       |             |              |
                       |             |              |
                    +--+--+       +--+--+        +--+--+
                    |  E  +-------+  F  +--------+  G  |
                    +-----+       +-----+        +-----+

           +--------+          +--------+          +--------+
           |IP(A->E)|          |IP(A->G)|          |IP(A->G)|
           +--------+          +--------+          +--------+
           |SRH     |          |SRH     |          |SRH     |(or PSP)
           |  SL:2  |          |  SL:1  |          |  SL:0  |
           |  L(E)  |          |  L(E)  |          |  L(E)  |
           |  L(G)  |          |  L(G)  |          |  L(G)  |
           |  L(H)  |          |  L(H)  |          |  L(H)  |
           +--------+          +--------+          +--------+
           | Packet |   --->   | Packet |   --->   | Packet |
           +--------+          +--------+          +--------+
```

Figure 3: Packet Forwarding Example with UET-MPLS U-SID

In the example shown in Figure 3, assume that routers A, E, G, and H are U-SID capable (i.e., both SR-MPLS and SRv6 capable) while the remaining routers (B, C, D, and F) are only capable of forwarding IP packets. Routers A, E, G, and H advertise their Segment Routing related information via IS-IS or OSPF.

Now assume that router A (the Domain ingress) wants to send a packet to router H (the Domain egress) via an SRv6 policy with the explicit path {E->G->H}. Router A will impose an MPLS label stack within SRH on the packet that corresponds to that explicit path. Router A searches ILM entry by the top label (that indicated router E), get the FEC information and next-hop/interface forwarding information, a loopback IPv6 address of E, and then copy to DA and sends the packet. SRH.UET is set to UET-MPLS and the value of SRH.SL is 2.

When the IPv6 packet arrives at router E, router E picks the next segment (label) within SRH based on the SRH.SL value of 2, searches ILM entry by the next label, get the FEC information and next-hop/ interface forwarding information, a loopback IPv6 address of G, and then copy to DA and sends the packet. SRH.UET is set to UET-MPLS, and the value of SRH.SL is 1.

When the IPv6 packet arrives at router G, router G gets the next segment (label) within SRH based on the SRH.SL value of 1, looks up ILM entry by the next label, gets the FEC information and next-hop/ interface forwarding information, a loopback IPv6 address of H, and then copies it to IP DA and transmits the packet. Because the value of SRH.SL is 0; the SRH can be removed if the behavior flavor codepoint of the above next segment (label) is set to PSP.

4.2. Procedures of 32bits IP Address within SRH

This section describes how the UET-32 type of U-SID is used to encode a compressed SRH.

[RFC6554] specifies the Source Routing Header (to avoid confusion with Segment Routing Header, we call it SRH3 according to type 3) for use strictly between RPL (Routing Protocol for Low-Power and Lossy Networks) routers in the same RPL routing domain. 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 SRH. For each entry in Address[1..n] within the Routing header, the shared prefix octets are not carried, but only a shorter truncated piece of the original 128bits. During packet forwarding, the shorter entry gets one by one and restored to the original IPv6 address. The Segment Left field represents the number of segments remaining, i.e., the number of explicitly listed intermediate nodes still to be visited before reaching the final destination, not the number of 128bits entries.

The above described mechanism, introduced in SRH3, could also be brought to Segment Routing Header (SRH). However, unlike in SRH3, using explicit fields within the Routing header to indicate the number of prefix octets common with the IPv6 Destination Address, this document introduces a new Flavor for Endpoint Behavior, defined in [I-D.ietf-spring-srv6-network-programming], termed as UET Flavor, for SRv6 SIDs. The UET Flavor of the current active SID indicates the next SID's compressed length within SRH, thus preparing the next SID of the corresponding length.

The UET Flavor information of a SID can be stored in the local SID entry of that SID.

This section defines the following two UET Flavors for Endpoint Behavior:

UET-32 Flavor: a SID with UET-32 Flavor means in SRH that the next SID is a 32bits IPv4 address or number.

UET-16 Flavor: a SID with UET-16 Flavor means in SRH that the next SID is a 16bits address or number.

For the convenience of expression, we can use UET-128 Flavor for the case when the next SID is a traditional 128bits IPv6 address. Note that UET-128 Flavor is not defined in the document.

An SRv6 SID MUST NOT have multiple UET Flavors at the same time.

4.2.1. Packet Forwarding Based on UET-32 U-SID

This section describes the packet forwarding based on UET-32 U-SID. For UET-16 U-SID, it is similar.

```
       +-----+       +-----+       +-----+        +-----+        +-----+
       |  A  +-------+  B  +-------+  C  +--------+  D  +--------+  H  |
       +-----+       +--+--+       +--+--+        +--+--+        +-----+
                        |             |              |
                        |             |              |
                     +--+--+       +--+--+        +--+--+
                     |  E  +-------+  F  +--------+  G  |
                     +-----+       +-----+        +-----+

            +--------+          +--------+            +--------+
            |IP(A->E)|          |IP(A->G)|            |IP(A->G)|
            +--------+          +--------+            +--------+
            |SRH     |          |SRH     |            |SRH     |
            |  SL:2  |          |  SL:1  |            |  SL:0  |
            |  32-H  |          |  32-H  |            |  32-H  |
            |  32-G  |          |  32-G  |            |  32-G  |
            |  32-E  |          |  32-E  |            |  32-E  |
            +--------+          +--------+            +--------+
            | Packet |   --->   | Packet |      --->  | Packet |
            +--------+          +--------+            +--------+
```

Figure 4: Packet Forwarding Example with UET-32 U-SID

In the example shown in Figure 4, assume that routers A, E, G, and H are U-SID capable while the remaining routers (B, C, D, and F) are only capable of forwarding IP packets. Routers A, E, G, and H

advertise their Segment Routing related information via IS-IS or OSPF, especially SRv6 SIDs with SID strucuture and UET-32 Flavor information.

Suppose that router A allocates an END SID B:32-A::, router E allocates an END SID B:32-E::, router G allocates an END SID B:32-G::, and router H allocates an END SID B:32-H::. All these SIDs have the same SID structure, i.e., share the same common prefix B (also known as the SRv6 SID Locator Block), and the sum of the Node Length, Function Length, Argument Length of each SID are the same.

Now assume that router A (the Domain ingress) wants to send a packet to router H (the Domain egress) via an SRv6 policy with the explicit path {E->;G->H}. Router A will impose a UET-32 U-SID stack within SRH on the packet that corresponds to that explicit path. The U-SID stack consists of three shorter 32bits UET-32 U-SIDs, which are 32-E, 32-G, 32-H. Router A gets the first U-SID 32-E from SRH and restores it to the original IPv6 address B:32-E::, then copy it to DA and sends the packet according to IPv6 FIB lookup. SRH.UET is initially set to UET-32 and the value of SRH.SL is 2.

When the IPv6 packet arrives at router E, match the local SID entry of B:32-E::. Router E get the next U-32 32-G within SRH based on the SRH.SL value of 2, and restore it to the original IPv6 address B:32-G::, then copy it to DA and sends the packet according to IPv6 FIB lookup. SRH.UET remains unchanged, and the value of SRH.SL is 1.

When the IPv6 packet arrives at router G, match the local SID entry of B:32-G::. Router G gets the next U-32 32-H within SRH based on the SRH.SL value of 1, and restore it to the original IPv6 address B:32-H::, then copy it to DA and sends the packet according to IPv6 FIB lookup. SRH.UET remains unchanged, and the value of SRH.SL is 0. The SRH can be removed if the local SID entry of B:32-G:: has PSP Flavor.

When the IPv6 packet arrives at router H, match the local SID entry of B:32-H:: and Proceed to process the next header in the packet.

5. The Use Case of Unified Segment Identifier

In addition to being used for compression, U-SID can also be used in interworking between SR-MPLS and SRv6 domains. SR-MPLS is often used in a metro network, for example, in the backhaul metro network of CMCC. If the core network uses SRv6, for example, the core network of the same operator, U-SID can be used in the SRv6 domain to interwork with SR-MPLS in the metro network to form an end-to-end SR policy or tunnel.

5.1. Nesting Interworking Between SR-MPLS and SRv6 Using Binding U-SID

SR-MPLS uses SR SIDs as MPLS label in the MPLS stack, and the SIDs are 32-bits long. SRv6 uses SR SIDs as IPv6 extension header in SRH, and the SIDs are 128-bits long.

The type UET-MPLS of U-SID uses the same 32-bits long SIDs in MPLS stack and SRH. Thus, four 32-bits long U-SIDs can be placed in the space of a single 128-bits long header. The encapsulation is illustrated in Figure 5.

```
        +---------+         +------------------------------------+
        |         |         |           IPv6 header              |
        | Ethernet|         +------------------------------------+
        |         |         |           SRH                      |
        +---------+         +------------------------------------+
        |  USID1  |         | USID1  | USID2  | ...   | USID4  |
        +---------+         +------------------------------------+
        |  USID2  |         | USID5  |...     | USIDn | Null   |
        +---------+         +------------------------------------+
        | ...     |         +           Payload                  |
        +---------+         +------------------------------------+
        |  USIDn  |
        +---------+
        | Payload |
        +---------+
```

Figure 5: 32-bits long U-SIDs Encapsulation

This document RECOMMENDS using Binding SID for interworking because Binding SID allows hiding the difference between U-SID types of different domains Additionally, a headend with only classical SRv6 SRH encapsulation capability, i.e., no capability to put multiple short U-SIDs to a single 128bits entry, will not need to upgrade.

Although Binding SID that is allocated for the specific SR policy instance will bring more states on some domain border nodes, the SR policy instance itself maybe pre-exist due to other requirements. The SR policy is created within each UET domain that can be upgraded separately.

To interwork, an MPLS Binding SID could be allocated for an SRv6 policy, used to hide the details of the UET-128 domain (classical SRv6) for a traditional MPLS Label stack. Similarly, an SRv6 Binding SID could be allocated for an SR-MPLS policy, used to hide the UET-MPLS domain's details for a conventional SRv6 SRH. An SRv6 Binding SID allocated for an SRv6 policy that enables the UET-32 compression style will hide the details of the UET-32 domain for a traditional

SRv6 SRH. There may be other combinations that are not discussed in the document.

Note that in some cases, Binding SID will cause multiple SRH to be inserted in IPv6 header.

The SR-MPLS and SRv6 interworking is illustrated in Figure 6. An end-to-end SR path from A to F crosses the SR-MPLS and SRv6 domains. The SR-MPLS domain could be using IPv4 or IPv6 address family. The SRv6 border nodes (E/G) receive SR-MPLS packets and forward them into the SRv6 domain using an SR-MPLS Binding SID [RFC8660].

```
        +-----+           +-----+           +-----+           +-----+
        |  A  +-----------+  B  +-----------+  E  +-----------+  F  |
        +-----+           +--+--+           +--+--+           +--+--+
           |    SR-MPLS      |                 |     SRv6        |
           |                 |                 |                 |
        +-----+           +--+--+           +--+--+           +--+--+
        |  C  |-----------|  D  +-----------+  G  +-----------+  H  |
        +-----+           +-----+           +-----+           +-----+


                                                   +--------------+
                                                   |   Eth(E->G)  |
           +--------------+                        +--------------+
           |   Eth(A->B)  |                        |IPv6 DA:G.intf|
           +--------------+    +--------------+    +--------------+
           |   USID(B)    |    |   Eth(B->E)  |    |SRH           |
           +--------------+    +--------------+    |NH:MPLS   SL:2|
           |   USID(E1)   |    |   USID(E1)   |    |USID(ADJ E->G)|
           +--------------+    +--------------+    |USID(ADJ G->H)|
           |   USID(E2)   |    |   USID(E2)   |    |USID(ADJ H->F)|
           +--------------+    +--------------+    +--------------+
           |Label(service)|    |Label(service)|    |Label(service)|
           +--------------+    +--------------+    +--------------+
           |    Payload   | -> |    Payload   | -> |    Payload   |
           +--------------+    +--------------+    +--------------+
```

Figure 6: SR-MPLS and SRv6 interworking

The SRv6 edge node E assigns two SIDs, e.g., E1 and E2, E1 is an SR-MPLS Node-SID, E2 is an SR-MPLS Binding-SID, which represents an SRv6 policy (from E to F, via segment list E-G-H-F) with U-SID encapsulation. At the headend A, the end-to-end segment list could be B-E1-E2. Figure 3 demonstrates an example of the packet forwarding, where U-SID is an MPLS label.

The reverse interworking is illustrated in Figure 7. An end-to-end SR path from F to A crosses the SRv6 and SR-MPLS domains. The SRv6 border nodes (E/G) receive SRv6 packets and forward them into the SR-MPLS domain using an SR-MPLS Binding SID or normal Prefix/Adjacency SID.

```
        +-----+           +-----+           +-----+           +-----+
        |  A  +-----------+  B  +-----------+  E  +-----------+  F  |
        +-----+           +--+--+           +--+--+           +--+--+
           |    SR-MPLS      |                 |     SRv6        |
           |                 |                 |                 |
        +-----+           +--+--+           +--+--+           +--+--+
        |  C  |-----------|  D  +-----------+  G  +-----------+  H  |
        +-----+           +-----+           +-----+           +-----+

                                                   +--------------+
                                                   |   Eth(F->H)  |
                                                   +--------------+
                                                   |IPv6 DA:H.intf|
                                                   +--------------+
                                                   |SRH           |
                                                   |NH:MPLS   SL:2|
                                                   |USID(ADJ F->H)|
                               +--------------+    |USID(ADJ H->G)|
                               |   Eth(E->B)  |    |USID(ADJ G->E)|
           +--------------+    +--------------+    +--------------+
           |   Eth(B->A)  |    |   USID(B)    |    |   USID(B)    |
           +--------------+    +--------------+    +--------------+
           |   USID(A)    |    |   USID(A)    |    |   USID(A)    |
           +--------------+    +--------------+    +--------------+
           |Label(service)|    |Label(service)|    |Label(service)|
           +--------------+    +--------------+    +--------------+
           |    Payload   | <- |    Payload   | <- |    Payload   |
           +--------------+    +--------------+    +--------------+
```

Figure 7: SR-MPLS and SRv6 reverse interworking

The SRv6 edge node F assigns an SR-MPLS Binding-SID F2, which represents an SRv6 policy (from F to E, via segment list F-H-G-E) with U-SID encapsulation. At the headend F, the end-to-end segment list could be F2-B-A.

5.2. Flat Interworking Between Different UET Domain Using Mixing U-SID

U-SRH can provide a different interworking scheme to support an end-to-end SR tunnel or policy using a mixing type of U-SIDs if more headend nodes have been upgraded to support encapsulating mixing

U-SID in SRH. For example, a SID list could contain some 128bits classical SIDs, some 32bits U-SIDs (IP or Label), and some 16bits U-SIDs at the same time. For this purpose, each U-SID in SRH must meet the alignment requirement. For example, a UET-32 U-SID is stored in a 4-byte alignment, and a UET-16 U-SID is stored in a 2-byte alignment.

The interworking of different UET domain is illustrated in Figure 8. An end-to-end SR tunnel or policy from S to D with segment list <X, ABR1, Y, ABR2, Z, D>, crosses the UET-128 domain, UET-32 domain and UET-MPLS domain. Note that any order of UET domains is also possible and is similar to the case displayed in Figure 8.

```
  ......................  ....................  .....................
  :                    :  :                  :  :                   :
+-----+     +-----+     +-----+     +-----+     +-----+     +-----+     +-----+
| S  +------+ X  +------+ABR1+------+ Y  +------+ABR2+------+ Y  +------+ D  |
+-----+     +-----+     +-----+     +-----+     +-----+     +-----+     +-----+
  :                    :  :                  :  :                   :
  ........UET-128.......  .......UET-32.......  .......UET-MPLS......
```

Figure 8: Interworking between different UET SID

5.2.1. UET Capability Advertisement

In an SRv6 network, each node can configure its U-SID Encapsulation Capability (UEC), and advertise it to other nodes. A controller can collect UEC information of all nodes. Typical UEC is:

UEC-128: Support classical 128bits SRv6 SID, which is the default capability of an SRv6 node.

UEC-32: Support shorter 32bits IPv4 address or number.

UEC-MPLS: Support shorter 32bits MPLS label.

UEC-16: Support shorter 16bits number.

Each node can support one or more than one UEC. Refer to Figure 8, node S/X/ABR1 can configure to support UEC-128 capability, node ABR1/Y/ABR2 can configure to support UEC-32 capability, and node ABR2/Y/D can configure to support UEC-MPLS capability.

A UET domain is constructed by several connected SRv6 nodes with the same UEC. For example, a UET-128 domain is constructed by the connected nodes all with UEC-128.

5.2.2. SRv6 SID Allocated per UEC

An SRv6 SID is allocated per UEC. For example, an SRv6 Node can allocate different END SIDs each for UEC-128, UEC-32, UEC-MPLS, etc.

The local SID entry of each SRv6 SID allocated per UEC will explicitly have the specific UET Flavor attribute information.

For flat interworking between different UET domain purpose, in addition to the two UET Flavors, i.e., UET-32 and UET-16 Flavors that is defined in section Section 4.2, here we continue to define a third UET Flavor for SRv6 SID:

UET-MPLS Flavor: a SID with UET-MPLS Flavor means in SRH the next SID is a 32bits MPLS label.

Each node allocates its SRv6 SID per UEC and advertises it to other nodes with additional UET-Flavor. A controller can collect these SIDs to be used for E2E SID List programming.

To save label resources, an MPLS label is not allocated per UEC. The related UET-Flavor information can be directly inserted in the context field of the label item in SRH. However, to meet the SRH processing restrictions defined in [RFC8754], it is possible to allocate MPLS labels for some of the topology-related SRv6 SIDs, which will consume more label resources.

Refer to the scenario presented in Figure 8, where each node may allocate the following SRv6 SID per UEC.

Node S: 128bits-END-SID-S for UEC-128.

Node X: 128bits-END-SID-X for UEC-128.

Node ABR1: 128bits-END-SID-ABR1 for UEC-128, and 128bits-END-SID-ABR1' for UEC-32.

Node Y: 128bits-END-SID-Y for UEC-32.

Node ABR2: 128bits-END-SID-ABR2 for UEC-32, and 128bits-END-SID-ABR2' for UEC-MPLS.

Node Z: 32bits-PREFIX-SID-Z. Note that MPLS Label allocation is independent with UEC.

Node D: 32bits-PREFIX-SID-D. Note that MPLS Label allocation is independent with UEC.

Note that the above SRv6 SID itself is always a 128bits IPv6 address, with no relationship with its UET Flavor attribute. The UET Flavor attribute indicates the next SID type, i.e., 128bits classical SID, 32bits IPv4 address, or 32bits MPLS Label, etc.

5.2.3. Packets Forwarding Procedures

Consider that the controller computes an E2E segment list <X, ABR1, Y, ABR2, Z, D>.

For the above E2E segment list, the controller knows which UET domain does each segment node belongs to, especially that ABR1 and ABR2 are the border nodes between different UET domain. Controller will select appropriate SID with specific UET Flavor attribute to indicate the UET domain which the next SID belongs to, i.e., whether the next SID is a classical IPv6 address or a shorter truncated value.

The SID list informed to headend could be:

FSU: First SID UET, which indicates the compression resulst of the first SID, in this example, it is set to UET-128.

No.1 SID: 128bits-END-SID-X (with BL|TL info of itself, and UET-128 Flavor to indicate the compression result of the next SID)

No.2 SID: 128bits-END-SID-ABR1' (with BL|TL info of itself, and UET-32 Flavor to indicate the compression result of the next SID)

No.3 SID: 128bits-END-SID-Y (with BL|TL info of itself, and UET-32 Flavor to indicate the compression result of the next SID)

No.4 SID: 128bits-END-SID-ABR2' (with BL|TL info of itself, and UET-MPLS Flavor to indicate the compression result of the next SID)

No.5 SID: 32bits-PREFIX-SID-Z, (with UET-MPLS Flavor to indicate the compression result of the next SID)

No.6 SID: 32bits-PREFIX-SID-D, (with UET-128 Flavor to indicate the compression result of the next SID)

Note:

FSU indicates the compression result of the first SRv6 SID itself, while the UET Flavor attribute of the first SID just indicates the compression result of the second SRv6 SID.

BL is the Block Length of SRv6 SID. TL is the Truncated Length of SRv6 SID, i.e., the compression result.

The headend analysis of how to get the compressed SID List.

FSU is UET-128, so the first SID 128bits-END-SID-X keep 128bits.

The No.1 SID, 128bits-END-SID-X, has UET-128 Flavor, which means the next SID, 128bits-END-SID-ABR1', also needs to keep 128bits.

The No.2 SID, 128bits-END-SID-ABR1' has UET-32 Flavor, which means the next SID, 128bits-END-SID-Y, needs to be compressed as 32bits IPv4 address.

The No.3 SID, 128bits-END-SID-Y, has UET-32 Flavor, which means the next SID, 128bits-END-SID-ABR2', needs to be compressed as 32bits IPv4 address.

The No.4 SID, 128bits-END-SID-ABR2', has UET-MPLS Flavor, which means the next SID, 32bits-PREFIX-SID-Z, needs to keep 32bits.

The No.5 SID, 32bits-PREFIX-SID-Z, has UET-MPLS Flavor, which means the next SID, 32bits-PREFIX-SID-D, needs to keep 32bits.

The No.6 SID, 32bits-PREFIX-SID-D, has UET-128 Flavor, which means the next SID, maybe a VPN service SRv6 SID, needs to keep 128bits.

Note that in some cases, an overlay VPN service SRv6 SID could be compressed. At that time, the last SID within the underlay segment list may select the one with UET-32 or UET-16 Flavor attribute.

Thus, the headend can get the following compressed SID List:

128bits-END-SID-X with UET-128 Flavor

128bits-END-SID-ABR1' with UET-32 Flavor

32bits of 128bits-END-SID-Y with UET-32 Flavor

32bits of 128bits-END-SID-ABR2' with UET-MPLS Flavor

32bits-PREFIX-SID-Z (with UET-MPLS Flavor in context.field)

32bits-PREFIX-SID-D (with UET-128 Flavor context.field)

At the However, to meet the SRH processing restrictions defined in [RFC8754], it is possible to allocate MPLS labels for some of the

topology-related SRv6 SIDs, which will consume more label resources.At the headend, the encapsulated SRH could be:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Hdr Ext Len  | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Last Entry   |   Flags |UET| |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~             128bits VPN-SID                                   ~ [0]
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
|          32bits-PREFIX-SID-D (with UET-128 in context.field)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          32bits-PREFIX-SID-Z (with UET-MPLS in context.field) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ [1]
|             32bits of 128bits-END-SID-ABR2' with UET-MPLS     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             32bits of 128bits-END-SID-Y with UET-32           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
~             128bits-END-SID-ABR1' with UET-32                 ~ [2]
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ---
~             128bits-END-SID-X with UET-128                    ~ [3]
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional Type Length Value objects (variable)       //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: SRH including different UET SID

The initial SRH.SL is set to 4: the number of 128bits based SIDs in SRH, and the initial SRH.UET is set to UET-128, according to FSU, which represents the first UET domain.

During the process of packets passing through multiple UET domains, if SRH.UET change from UET-128 to UET-32 or UET-MPLS, SRH.SL will quadruple, i.e., SRH.SL = SRH.SL * 4, which is the number of 32bits based SIDs in SRH. When SRH.UET changed from UET-32 or UET-MPLS to UET-128, SRH.SL will revert to its original size, i.e., SRH.SL = SRH.SL / 4, which is the number of 128bits based SIDs in SRH.

Similarly, when SRH.UET change from UET-128 to UET-16, SRH.SL = SRH.SL * 8, from UET-32 to UET-16, SRH.SL = SRH.SL * 2, vice versa.

Refer to Figure 8, next we will describe the process of packets passing through each UET domain.

At the headend S, when packets sent to the No.1 segment node X, it will decrement SRH.SL by 1, get the first 128bits SID from SRH.List[], 128bits-END-SID-X with UET-128, copy to DA, and lookup FIB to send packets. Now, SRH.SL is 3 and SRH.UET is UET-128.

At the No.1 segment node X, the local SID matches the DA and has UET-128 Flavor attribute. Hence, SRH.UET has no change. It will decrement SRH.SL by 1, get the next 128bits SID from SRH.List[], 128bits-END-SID-ABR1' with UET-32, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 2 and SRH.UET is UET-128.

At the No.2 segment node ABR1, the local SID matches the DA has UET-32 Flavor attribute. Hence, SRH.UET has changed from UET-128 to UET-32. Tne node will firstly calculate SRH.SL * 4, then decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits of 128bits-END-SID-Y with UET-32, convert it to a complete IPv6 SID, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 7 and SRH.UET is UET-32.

At the No.3 segment node Y, the local SID matches the DA has UET-32 Flavor attribute. Thus, SRH.UET has no change. The node will decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits of 128bits-END-SID-ABR2' with UET-MPLS, convert it to a complete IPv6 SID, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 6, SRH.UET is UET-32.

At the No.4 segment node ABR2, the local SID matches the DA has UET-MPLS Flavor attribute. Hence, SRH.UET has changed from UET-32 to UET-MPLS. Because the size of SID has no change, the node will decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits-PREFIX-SID-Z (with UET-MPLS in context.field), map it to a complete IPv6 prefix FEC by ILM entry, copy to DA, and lookup FIB (or directly get forwarding information from ILM entry) to send packets. Note that the UET information in context.field needs to be compared to SRH.UET. Since values are equal no change and no additional processing. At this time, SRH.SL is 5, SRH.UET is 0b02.

At the No.5 segment node Z, the normal address route entry matches the DA and has no UET Flavor attribute. As a result, SRH.UET has no change. The node decrements SRH.SL by 1, will get the next 32bits SID from SRH.List[], 32bits-PREFIX-SID-D (with UET-128 in context.field), map it to a complete IPv6 prefix FEC by ILM entry, copy to DA, and lookup FIB (or directly get forwarding information from ILM entry) to send packets. Note that the UET information in context.field needs to be compared to SRH.UET. Because it is changed from UET-MPLS to UET-128, the SRH.SL will be reverted to its original size, i.e., let SRH.SL / 4. At this time, SRH.SL is 1, SRH.UET is UET-128.

At the No.6 segment node D, the normal address route entry matched by DA has no associated UET Flavor attribute. Hence, SRH.UET has no change. The node decrements SRH.SL by 1, will get the next 128bits SID from SRH.List[], 128bits VPN-SID, and follow the rest process described in [I-D.ietf-spring-srv6-network-programming].

6. Control Plane in Support of Unified SID

The introduction of the Unified Identifier may rely on the existing SR extensions to the routing protocols. But some enhancements in the control plane are still required. This section references the existing protocols and identifies necessary extensions.

Each node in the SRv6 domain needs to advertise its U-SID Encapsulation Capability, this information can be carried within SRv6-Capabilities sub-TLV defined in [I-D.ietf-lsr-isis-srv6-extensions] and SRv6 Capabilities TLV defined in [I-D.ietf-lsr-ospfv3-srv6-extensions]. It need also allocate SRv6 SID (Topology type and Service Function type) per UEC and advertise to other nodes, the advertisement of SRv6 END SID, END.X SID, LAN END.X SID defined in [I-D.ietf-lsr-isis-srv6-extensions] and [I-D.ietf-lsr-ospfv3-srv6-extensions] need to be extended to carry UET-Flavor information. This information can be collected and sent to the central controller through BGP-LS. The controller then can send the computed segment list to the headend through BGP or PCEP, and each segment will include explicit UET Flavor information. The detailed procedures are outside the scope of this document.

The SR-MPLS extensions to Interior Gateway Protocols (IGP), IS-IS [RFC8667], OSPF [RFC8665], and OSPFv3 [RFC8666], defined how 20-bits and 32-bits SIDs advertised and bound to SR objects and/or instructions. Extensions to BGP Link-state address family [I-D.ietf-idr-bgp-ls-segment-routing-ext] enabled propagation of segment information of variable length via BGP. The existed SR-MPLS extensions can be used to get MPLS U-SID mapping FIB entry, and it can coexist with SRv6 extensions to the same IGP/BGP-LS instance. For simplicity, this document suggests using the existing mature SR-MPLS control plane and FIB entry for the MPLS U-SID advertisement and mapping entry. However, it is possible to base it on SRv6 related TLVs/sub-TLVs to advertise the MPLS U-SID, which will be discussed in another document.

7. SRH with U-SID Pseudo-code

Processing of SRH with U-SID is demonstrated in the following pseudo-code:

Headend sending packet:

```
S01. set initial SRH.UET, respond to the FSU, i.e.,
        the compressed result of the first SID;
S02. set initial SRH.SL, it is the count of 128bits-based SIDs;
S03. if (SRH.UET == UET-128) {
S04.   SRH.SL --;
S05.   Get SRH.List[SRH.SL], 128bits, copy to IPv6 Header DA; Or,
       headend know the first SID before SRH encapsulation,
        just copy it to DA.
S06.   FIB lookup according to DA, and forward packet;
S07. }
S08. else if (SRH.UET == UET-32) {
S09.   SRH.SL =  SRH.SL * 4;
S10.   SRH.SL --;
S11.   Get SRH.List[SRH.SL], 32bits, convert to 128bits SRv6 SID, copy
       to IPv6 Header DA; Or, headend know the first SID before SRH
           encapsulation, just copy it to DA;
S12.   FIB lookup according to DA, and forward packet;
S13. }
S14. else if (SRH.UET == UET-MPLS) {
S15.   SRH.SL =  SRH.SL * 4;
S16.   SRH.SL --;
S17.   Get SRH.List[SRH.SL], 32bits, lookup ILM entry and map it to 128
       IPv6 address, copy it to IPv6 Header DA; Or, headend know
       the first SID before SRH encapsulation, just copy it to DA;
S18.   FIB lookup according to DA, or, directly get forwarding
       information from ILM entry, and forward packet;
S19. }
S20. else if (SRH.UET == UET-16) {
S21.   SRH.SL =  SRH.SL * 8;
S22.   SRH.SL --;
S23.   Get SRH.List[SRH.SL], 16bits, convert to 128bits SRv6 SID, copy
       to IPv6 Header DA; Or, headend know the first SID before SRH
           encapsulation, just copy it to DA;
S24.   FIB lookup according to DA, and forward packet;
S25. }
```

Transit/Egress receive packets:

```
S01. If DA matched local SID entry, copy the UET attr of local SID entry
     to SRH.UET, check when SRH.UET changed from UET-128 to UET-32 or
         UET-MPLS, SRH.SL*4, when from UET-32 or UET-MPLS to UET-128,
          SRH.SL / 4, similar treatment of UET-16 related SRH.SL update;
         Else If DA matched normal address route entry,
            SRH.UET no update;
S02. if (SRH.SL == 0) {
S03.   process the inner payload;
S04. }
S05. else {
S06.   if (SRH.UET == UET-128) {
S07.     SRH.SL -- ;
S08.     Get SRH.List[SRH.SL], 128bits, copy it to IPv6 Header DA;
S09.     FIB lookup according to DA, and forward packet;
S10.   }
S11.   else if (SRH.UET == UET-32) {
S12.     SRH.SL -- ;
S13.     Get SRH.List[SRH.SL], 32bits, convert to 128bits SRv6 SID, copy
         to IPv6 Header DA;
S14.     FIB lookup according to DA, and forward packet;
S15.   }
S16.   else if (SRH.UET == UET-MPLS) {
S17.     SRH.SL --
S18.     Get SRH.List[SRH.SL], 32bits, lookup ILM entry, map it to
         128bits IPv6 address, copy it to IPv6 Header DA;
S19.     Get UET info from SRH.List[SRH.SL] Context Field, copy it to
         SRH.UET. Check if  SRH.UET changed from UET-MPLS to UET-128,
         SRH.SL/4;
S20.     FIB lookup according to DA, or, directly get forwarding
         information from ILM entry, and forward packet;
S21.   }
S22.   else if (SRH.UET == UET-16) {
S23.     SRH.SL -- ;
S24.     Get SRH.List[SRH.SL], 16bits, convert to 128bits SRv6 SID, copy
         to IPv6 Header DA;
S25.     FIB lookup according to DA, and forward packet;
S26.   }
S27. }
```

8. U-SID supporting SRv6 programming

U-SID can support SRv6 programming defined by [I-D.ietf-spring-srv6-network-programming]. The details will be described in another document.

9. Benefits

To be discussed in the next version.

10. Implementation Considerations

The Unified SID solution has been already implemented and tested by two companies:

- o Centec has conducted its PoC, and the report is available at https://cloud.tencent.com/developer/article/1540023.

- o Broadcom, in its lab, also conducted PoC testing of the U-SID solution.

11. IANA Considerations

IANA is requested to allocate from the Segment Routing Header Flags registry the two-bits long field referred to as Size.

12. Security Considerations

This specification inherits all security considerations of [RFC8402] and [RFC8754].

13. Contributors

Wan Xiaolan
New H3C Technologies Co. Ltd
No.8, Yongjia Road, Haidian District
Beijing
China

Email: wxlan@h3c.com

Cheng Wei
Centec
Building B, No.5 Xing Han Street, Suzhou Industrial Park
Suzhou
China

Email: Chengw@centecnetworks.com

S.Zadok
Broadcom
Israel

Email: shay.zadok@broadcom.com

14. Acknowledgements

TBD

15. Normative References

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "BGP Link-State extensions for Segment Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-16 (work in progress), June 2019.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-11 (work in progress), October 2020.

[I-D.ietf-lsr-ospfv3-srv6-extensions]
Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak, "OSPFv3 Extensions for SRv6", draft-ietf-lsr-ospfv3-srv6-extensions-01 (work in progress), August 2020.

[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.

[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>.

[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>.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8663] Xu, X., Bryant, S., Farrel, A., Hassan, S., Henderickx, W., and Z. Li, "MPLS Segment Routing over IP", RFC 8663, DOI 10.17487/RFC8663, December 2019, <https://www.rfc-editor.org/info/rfc8663>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[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

Cheng Weiqiang
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Peng Shaofu
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: peng.shaofu@zte.com.cn

Liu Aihua
ZTE Corporation
Zhongxing Industrial Park, Nanshan District
Shenzhen
China

Email: liu.aihua@zte.com.cn

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com