

Network
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2021

C. Weiqiang
China Mobile
G. Mirsky
ZTE Corp.
P. Shaofu
L. Aihua
ZTE Corporation
G. Mishra
Verizon Inc.
July 2, 2020

Unified Identifier in IPv6 Segment Routing Networks
draft-mirsky-6man-unified-id-sr-07

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.

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 3, 2021.

Copyright Notice

Copyright (c) 2020 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

1.	Introduction	3
1.1.	Conventions used in this document	3
1.1.1.	Acronyms	3
1.1.2.	Requirements Language	4
2.	Segment Routing Extension Header: Benefits and Challenges	4
3.	Unified SIDs in IPv6 Segment Routing Extension Header	4
4.	The Use Case of Unified Segment Identifier	6
4.1.	Interworking Between SR-MPLS and SRv6 Using U-SID	6
5.	Operations with Unified Segment Identifier	9
5.1.	Procedures of SR-MPLS over IP	10
5.2.	Packet Forwarding	10
6.	Interworking Between Different UET domain Considerations	12
6.1.	Interworking Using Binding SID	12
6.2.	Interworking Using Mixing U-SID	12
6.2.1.	UET capability Advertisement	13
6.2.2.	SRv6 SID Allocated per UET	13
6.2.3.	Packets Forwarding Procedures	15
6.2.4.	SRH with mixing elements Pseudo-code	18
7.	Control Plane in Support of Unified SID	19
8.	U-SID supporting SRv6 programming	20
9.	Implementation Considerations	20
10.	IANA Considerations	20
11.	Security Considerations	21
12.	Contributors	21
13.	Acknowledgements	21
14.	Normative References	21
	Authors' Addresses	23

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) [[I-D.ietf-6man-segment-routing-header](#)].

This document extends the use of the SRH [[I-D.ietf-6man-segment-routing-header](#)] 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

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 [[I-D.ietf-6man-segment-routing-header](#)], 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 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 the necessity for operators. It 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 [[I-D.ietf-6man-segment-routing-header](#)] as presented in Figure 1

0b10 - 32-bits 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 a reference to the U-SID network programming document.]

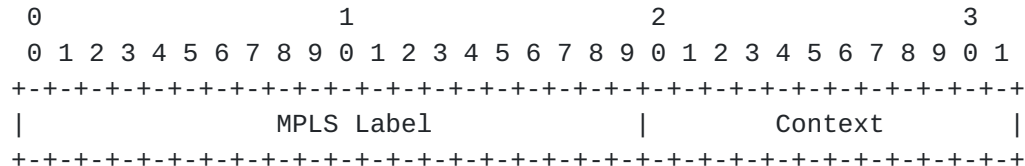


Figure 2: Format of Unified SID with MPLS Label

0b11 - Reserved for future use.

This document also introduce a compatible operation on Segment Left field, also termed as SRH.SL. That is, if SRH.UET Flag is 0b00, SRH.SL represents the count of 128bits-SID items in SRH, if SRH.UET Flag is 0b01 or 0b10, SRH.SL represents the count of 32bits-SID items in SRH.

4. The Use Case of Unified Segment Identifier

U-SID can be used for 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 tunnel.

4.1. Interworking Between SR-MPLS and SRv6 Using U-SID

SR-MPLS uses SR SIDs as MPLS label in 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 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 3.

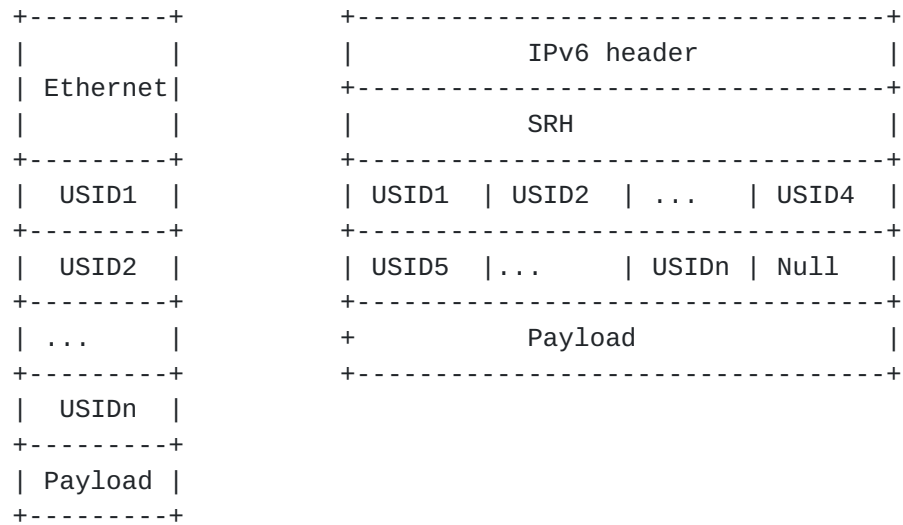


Figure 3: 32-bits long U-SIDs Encapsulation

The SR-MPLS and SRv6 interworking is illustrated in Figure 4. An end-to-end SR tunnel 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](#)].

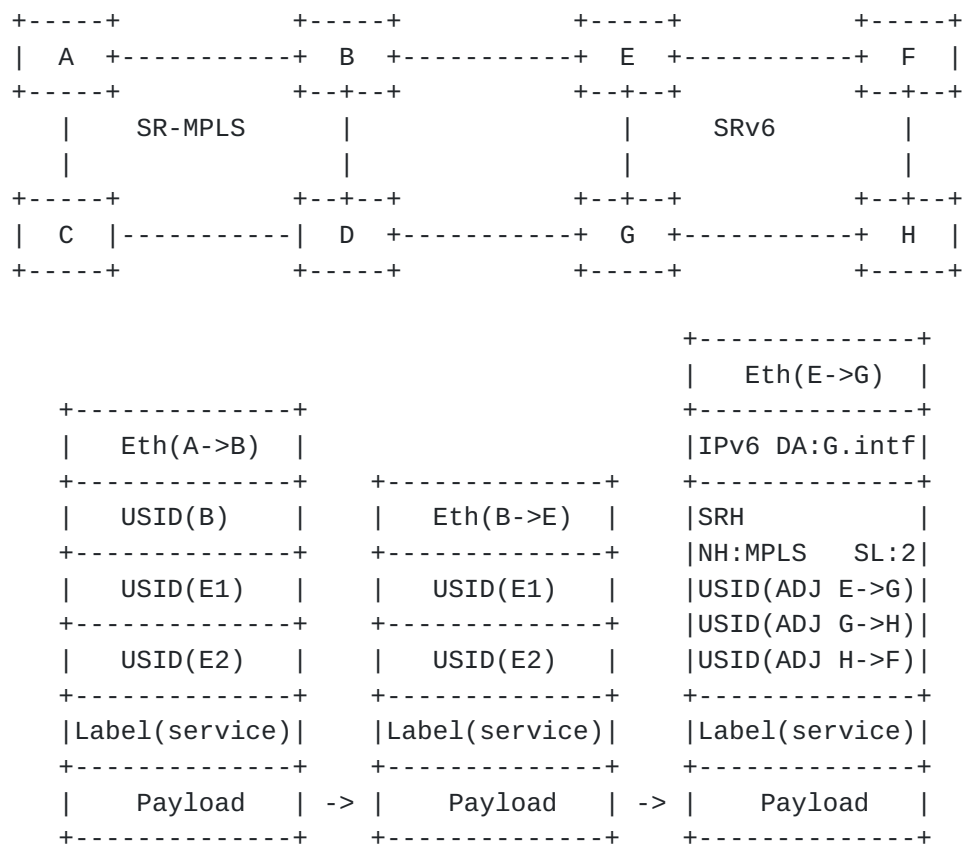


Figure 4: SR-MPLS and SRv6 interworking

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 6 demonstrates an example of the packet forwarding, where U-SID is an MPLS label.

The reverse interworking is illustrated in Figure 5. An end-to-end SR tunnel 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.

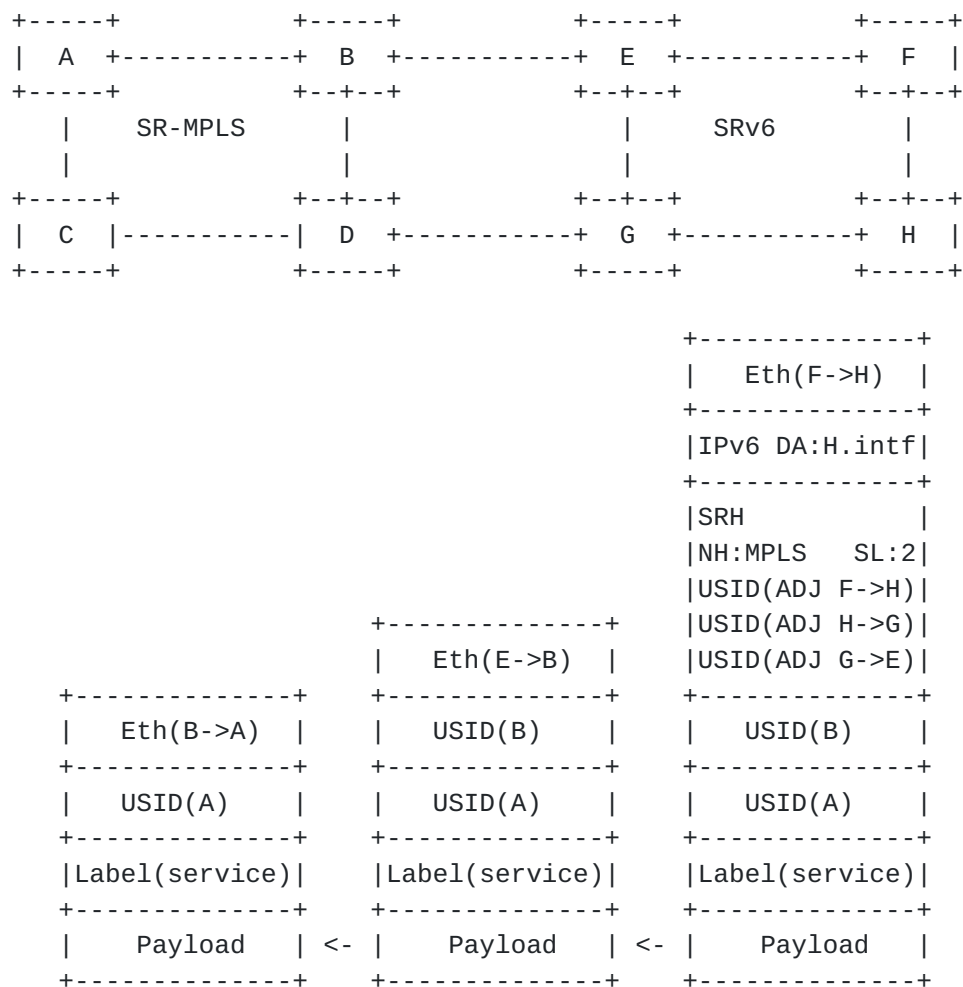


Figure 5: SR-MPLS and SRv6 reverse interworking

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. 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 [[I-D.ietf-6man-segment-routing-header](#)] respectively.

If U-SID is used to support interworking between SR-MPLS and SRv6 domains, it is beneficial that U-SID type matches to an MPLS label. In that 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 of the SR policy is not discarded, but UDP not.

[5.1.](#) Procedures of SR-MPLS over IP

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 the 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 both FEC and next-hop/interface forwarding information from the ILM entry, to avoid extra FIB lookup. An SRv6 policy chosen to encapsulate 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.

[5.2.](#) Packet Forwarding

U-SID based packet forwarding 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.

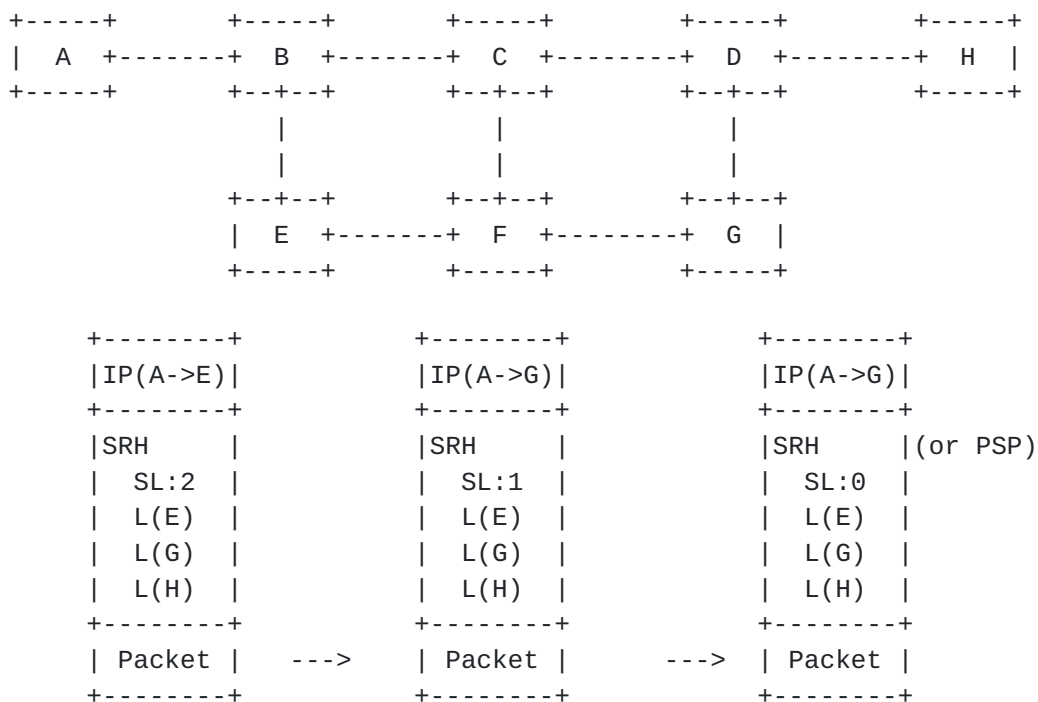


Figure 6: Packet Forwarding Example

In the example shown in Figure 6, 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. 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. 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 next segment (label) is set to PSP.

6. Interworking Between Different UET domain Considerations

6.1. Interworking Using Binding SID

In Figure 4 and Figure 5, We have seen a simple interworking solution based on Binding SID.

This document RECOMMEND this method for interworking, because Binding SID is very simple to hide the difference U-SID type of different domain. Especially, a headend only with classical SRv6 SRH encapsulation capability, i.e., no capability to put multiple short U-SIDs to a single 128bits item, 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 which is upgraded separately.

In order to do interwork, as mentioned before, an MPLS Binding SID could be allocated for an SRv6 policy, used to hide the details of UET-0b00 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 details of UET-0b10 domain for a traditional SRv6 SRH. And, an SRv6 Binding SID allocated for an SRv6 policy which enable UET-0b01 compression style, will hide the details of UET-0b01 domain for a traditional SRv6 SRH. There maybe other combinations, and not list one by one.

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

6.2. Interworking Using Mixing U-SID

U-SRH can also provide an alternate interworking scheme to support an end-to-end SR tunnel or policy using mixing type of U-SIDs, if more headend nodes have be upgraded to support encapsulating mixing U-SID in SRH. For example, an SID list could contain some 128bits-SIDs, some 32bits-SIDs, some 32bits-Labels. For this purpose, we need explicitly define U-SID types, in other words, the type is just UET.

The interworking of different UET domain is illustrated in Figure 7. An end-to-end SR tunnel or policy from S to D with segment list <X,

ABR1, Y, ABR2, Z, D>, crosses the UET-0b00 domain, UET-0b01 domain and UET-0b10 domain. Note that any order of UET domains are also possible and similar with the following illustration.

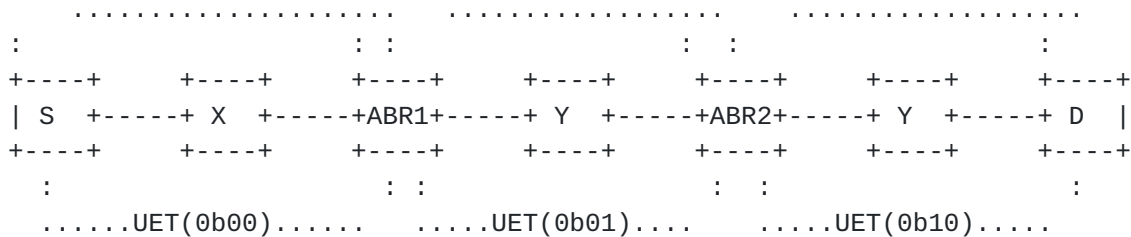


Figure 7: Interworking between different UET SID

6.2.1. UET capability Advertisement

In SRv6 network, each node can configure its UET capability, and advertise it to other nodes. Controller can also collect UET capability information of all nodes. Typical UET capability is:

UET-0: Support classical 128bits SRv6 SID.

UET-1: Support shorter 32bits IPv4 address or number.

UET-2: Support shorter 32bits MPLS label.

UET-3: Support shorter 16bits number.

Each node can support one or more than one UET capabilities. Refer to Figure 7, node S/X/ABR1 can configure to support UET-0 capability, node ABR1/Y/ABR2 can configure to support UET-1 capability, and node ABR2/Y/D can configure to support UET-2 capability.

6.2.2. SRv6 SID Allocated per UET

An SRv6 SID is allocated per UET capability. In control plane, an SRv6 SID is always 128bits, however:

An SRv6 SID that has UET-0 attribute means it is allocated for traditional 128bits encapsulation purpose, that means in SRH the next SID is also a 128bits SID.

An SRv6 SID that has UET-1 attribute means it is allocated for 32bits IPv4 encapsulation purpose, that means in SRH the next SID is a 32bits IPv4 address.

An SRv6 SID that has UET-2 attribute means it is allocated for 32bits MPLS encapsulation purpose, that means in SRH the next SID is a 32bits MPLS Label.

An SRv6 SID that has UET-3 attribute means it is allocated for 16bits number encapsulation purpose, that means in SRH the next SID is a 16bits number.

For the local SID entry of each SRv6 SID allocated per UET capability, it will explicitly give the UET attribute information.

Each node allocate its SRv6 SID per UET capability, and advertise it to other nodes with additional UET-Flavor. Controller can also collect these SIDs used for E2E SID List programming.

In order to save label resources, MPLS label is not allocated per UET. The UET information can be directly inserted in the context field of the label item in SRH.

For example, refer to Figure 7, each node allocate the following SRv6 SID per UET.

Node S: 128bits-END-SID-S for UET-0.

Node X: 128bits-END-SID-X for UET-0.

Node ABR1: 128bits-END-SID-ABR1 for UET-0, and 128bits-END-SID-ABR1' for UET-1.

Node Y: 128bits-END-SID-Y for UET-1.

Node ABR2: 128bits-END-SID-ABR2 for UET-1, and 128bits-END-SID-ABR2' for UET-2.

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

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

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

6.2.3. Packets Forwarding Procedures

Controller compute an E2E segment list <X, ABR1, Y, ABR2, Z, D>.

Controller knows that ABR1 and ABR2 are the border nodes between different UET domain for the above segment list.

So, the SID list informed to headend is:

No.1 SID: 128bits-END-SID-X (with UET-0 Flag, and BL|NBL info)

No.2 SID: 128bits-END-SID-ABR1' (with UET-1 Flag, and BL|NBL info)

No.3 SID: 128bits-END-SID-Y (with UET-1 Flag, and BL|NBL info)

No.4 SID: 128bits-END-SID-ABR2' (with UET-2 Flag, and BL|NBL info)

No.5 SID: 32bits-PREFIX-SID-Z, (with UET-2 Flag)

No.6 SID: 32bits-PREFIX-SID-D, (with UET-0 Flag)

Note: BL is Block Length, NBL is non-Block Length.

The headend analysis how to get the compressed SID List.

The No.1 SID, 128bits-END-SID-X, has UET-0 Flag, so keep 128bits.

The No.1 SID, 128bits-END-SID-X, has UET-0 Flag, that means the next SID, 128bits-END-SID-ABR1', need also keep 128bits.

The No.2 SID, 128bits-END-SID-ABR1' has UET-1 Flag, that means the next SID, 128bits-END-SID-Y, need to be compressed as 32bits IPv4 address.

The No.3 SID, 128bits-END-SID-Y, has UET-1 Flag, that means the next SID, 128bits-END-SID-ABR2', need to be compressed as 32bits IPv4 address.

The No.4 SID, 128bits-END-SID-ABR2', has UET-2 Flag, that means the next SID, 32bits-PREFIX-SID-Z, need keep 32bits.

The No.5 SID, 32bits-PREFIX-SID-Z, has UET-2 Flag, that means the next SID, 32bits-PREFIX-SID-D, need keep 32bits.

The No.6 SID, 32bits-PREFIX-SID-D, has UET-0 Flag, that means the next SID, maybe a VPN service SRv6 SID, need keep 128bits.

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

128bits-END-SID-X for UET-0

128bits-END-SID-ABR1' for UET-1

32bits of 128bits-END-SID-Y for UET-1

32bits of 128bits-END-SID-ABR2' for UET-2

32bits-PREFIX-SID-Z (with UET-2 in context.field)

32bits-PREFIX-SID-D (with UET-0 in context.field)

At 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-0 in context.field) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                32bits-PREFIX-SID-Z (with UET-2 in context.field) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ [1]
|                32bits of 128bits-END-SID-ABR2' for UET-2 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                32bits of 128bits-END-SID-Y for UET-1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                128bits-END-SID-ABR1' for UET-1                ~ [2]
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                128bits-END-SID-X for UET-0                    ~ [3]
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                                                                //
//                Optional Type Length Value objects (variable) //
//                                                                //
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 8: SRH including different UET SID

The initial SRH.SL is set to 4, that is the count of 128bits based SIDs in SRH, and the initial SRH.UET is set to 0b00, that represent the first UET domain.

During the process of packets passing through multiple UET domains, if SRH.UET change from 0b00 to 0b01 or 0b10, SRH.SL will quadruple,

i.e., $SRH.SL = SRH.SL * 4$, that is the count of 32bits based SIDs in SRH. When SRH.UET changed from 0b01 or 0b10 to 0b00, SRH.SL will revert to its original size, i.e., $SRH.SL = SRH.SL / 4$, that is the count of 128bits based SIDs in SRH.

Refer to Figure 7, 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 for UET-0, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 3, SRH.UET is 0b00.

At the No.1 segment node X, the local SID matched by DA has UET-0 attribute, that is, SRH.UET has no change. It will continue to decrement SRH.SL by 1, get the next 128bits SID from SRH.List[], 128bits-END-SID-ABR1' for UET-1, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 2, SRH.UET is 0b00.

At the No.2 segment node ABR1, the local SID matched by DA has UET-1 attribute, that is, SRH.UET has changed from UET-0 to UET-1. It will firstly let $SRH.SL * 4$, then decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits of 128bits-END-SID-Y for UET-1, convert it to a complete IPv6 SID, copy to DA, and lookup FIB to send packets. At this time, SRH.SL is 7, SRH.UET is 0b01.

At the No.3 segment node Y, the local SID matched by DA has UET-1 attribute, that is, SRH.UET has no change. It will continue to decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits of 128bits-END-SID-ABR2' for UET-2, 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 0b01.

At the No.4 segment node ABR2, the local SID matched by DA has UET-2 attribute, that is, SRH.UET has changed from UET-1 to UET-2. Because the size of SID has no change, it will continue to decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits-PREFIX-SID-Z (with UET-2 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 need update to SRH.UET again and see if it changes, no change at this time, so there is 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 matched by DA has not any further UET attribute, that is, SRH.UET has no change. It will continue to decrement SRH.SL by 1, get the next 32bits SID from SRH.List[], 32bits-PREFIX-SID-D (with UET-0 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 need update to SRH.UET again and see if it changes, it is changed from UET-2 to UET-0, so SRH.SL will be revert to its original size, i.e., let $SRH.SL / 4$. At this time, SRH.SL is 1, SRH.UET is 0b00.

At the No.6 segment node D, the normal address route entry matched by DA has not any further UET attribute, that is, SRH.UET has no change. It will continue to decrement SRH.SL by 1, 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](#)].

6.2.4. SRH with mixing elements Pseudo-code

Processing of SRH with mixing elements is demonstrated in the following pseudo-code:

Headend sending packet:

```
S01. set initial SRH.UET, respond to the UET type of the first SID;
S02. set initial SRH.SL, it is the count of 128bits-based SID;
S03. if (SRH.UET == 0b00) {
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 == 0b01) {
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 == 0b10) {
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. }
```


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 0b00 to 0b01 or 0b10,
        SRH.SL*4, when from 0b01 or 0b10 to 0b00, SRH.SL / 4;
    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 == 0b00) {
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 == 0b01) {
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 == 0b10) {
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 0b10 to 0b00, SRH.SL/4;
S20.         FIB lookup according to DA, or, directly get forwarding
            information from ILM entry, and forward packet;
S21.     }
S22. }
```

7. 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 to the existing protocols and identifies necessary extensions.

Each node in the SRv6 domain need 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 UET 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. These information can be collected and sent to central controller through BGP-LS. Then, controller can send the computed segment list to the headend through BGP or PCEP, and each segment will include an explicit UET information. All these will be defined in other documents.

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 purpose, this document suggest to use the existed mature SR-MPLS control plane and FIB entry to server the MPLS U-SID advertisement and mapping entry. However, it is possible to based on SRv6 related TLVs/sub-TLVs to advertise the MPLS U-SID, and that will be discussed in another document.

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. 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.

10. IANA Considerations

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

11. Security Considerations

This specification inherits all security considerations of [[RFC8402](#)] and [[I-D.ietf-6man-segment-routing-header](#)].

12. 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

13. Acknowledgements

TBD

14. Normative References

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", [draft-ietf-6man-segment-routing-header-26](#) (work in progress), October 2019.

[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-08](#) (work in progress), April 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-00](#) (work in progress), February 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-16](#) (work in progress), June 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>>.

[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>>.

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