SPRING
Internet-Draft
Intended status: Informational
Expires: September 7, 2020

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

March 6, 2020

SRv6 network programming using Unified Identifier
draft-mirsky-spring-unified-id-network-programming-00

Abstract

This draft describes how Unified Segment Identifier can be used to achieve the goals of SRv6 network programming.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Conventions used in this document 2
1.1.1. Terminology 3
1.1.2. Requirements Language 3
2. SRv6 Network Programming using U-SID 3
2.1. SRv6 Network Programming 3
2.2. SRv6 Network Programming Using 32bit U-SID 4
2.3. U-SID with MPLS Programming Process 5
2.3.1. U-SID with MPLS Support Programming using Flavors . . 5
2.4. U-SID with SRv6 Programming process 6
2.5. U-SID Complementary Method 6
3. IANA Considerations 7
4. Security Considerations 7
5. Acknowledgements 7
6. Normative References 7
Authors' Addresses 8

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].

Unified SID [I-D.mirsky-6man-unified-id-sr] defines an extension of SRH that enables the use of a shorter segment identifier, such as 32-bits Label format SID or 32-bits IP address format SID.

SRv6 network programming is defined [I-D.ietf-spring-srv6-network-programming]. SRv6 network programming can be supported using Unified SID.

1.1. Conventions used in this document

1.1.1. Terminology

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

SRv6: Segment Routing in IPv6

U-SID: Unified Segment Identifier

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. SRv6 Network Programming using U-SID

2.1. SRv6 Network Programming

[I-D.ietf-spring-srv6-network-programming] defines an SRv6 SID as consisting of LOC:FUNCT:ARG, where a locator (LOC) is encoded in the L most significant bits of the SID, followed by F bits of function (FUNCT) and A bits of arguments (ARG). L, the length of the locator, is flexible, and an operator is free to use the locator length of their choice. F and A may be any value as long as L + F + A <= 128. When L + F + A is less than 128, then the remainder of the SID MUST be zero.

A locator may be represented as B:N where B is the SRv6 SID block (IPv6 subnet allocated for SRv6 SIDs by the operator) and N is the identifier of the parent node instantiating the SID. The FUNCT is an opaque identification of a local behavior bound to the SID. An SRv6 endpoint behavior MAY require additional information for its processing (e.g., related to the flow or service). This information MAY be encoded in the ARG bits of the SID.

2.2. SRv6 Network Programming Using 32bit U-SID

[I-D.mirsky-6man-unified-id-sr] defines a 32 bits SID as an MPLS label or an IPv4 address or a complementary SID to a common IPv4/IPv6 prefix. If the U-SID represents an MPLS label, it could be mapped to the 128-bits SRv6 SID. And if this U-SID represents a complementary U-SID to a common IPv6 prefix, it could be associated with an SRv6 SID (a method to establish such association could use mapping, stitching, shifting, or translation). This SID can be compliant to the programming SID format as LOC:FUNCT:ARG, this means complementing the SRv6 SID of programming format to 32-bits U-SID. A U-SID with MPLS label format can support network programming, as illustrated 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           U-LOC (20bit, Label)        |P|U|D|R|U-FUNCT (U-ARG)|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                            |    Context (12bit)    |
                                            +-----------------------+
```

Figure 1: Example of U-SID supporting Network Programming with SR-MPLS

The context field can be defined as follow:

P-Flag: PSP (Penultimate Segment Pop of the SRH) Flag. If set, then the penultimate segment node MUST remove the SRH from the IPv6 extension header chain.

U-Flag: USP (Ultimate Segment Pop of the SRH) Flag. If set, then the ultimate segment node MUST remove the SRH from the IPv6 extension header chain and proceed to process the next header in the packet.

D-Flag: USD (Ultimate Segment Decapsulation) Flag. If set, then the ultimate segment node MUST skip the SRH processing and proceed to the next header.

R-Flag: Reserved Flag.

Function: 8-bits to store the short KEY for the specific table lookup. The MPLS label in the leftmost 20-bits will identify the context-specific table. For the context table that has a longer KEY than 8-bits, the next 32-bits SID could be used for this purpose.

A format of U-SID as 32-bits IP address can support network programming, as illustrated in Figure 2.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             U-LOC             |        U-FUNCT (U-ARG)        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Example of U-SID supporting Network Programming with SRv6

In this case, U-SID is split to 16-bits locator (U-LOC) and 16-bits function (U-FUNCT, U-ARG is optional). The operator can use any method to compress the 128-bits SRv6 SID to 32-bits complementary U-SID, such as mapping, stitching, shifting or translation, etc. For example, an operator can simply compress the original locator to a shorter locator. If the original SRv6 locator consists of B:N, this case the N is 16-bits. So, only the N is the U-LOC of U-SID and the B can be advertised by IGP protocol in the domain. the length of U-LOC and U-FUNCT (U-ARG) is flexible, and an operator is free to use the length of their choice. The length of U-LOC and U-FUNCT (U-ARG) may be any value as long as its sum mo more than 32. The compressing and the advertising method are out of the scope of this draft.

2.3. U-SID with MPLS Programming Process

2.3.1. U-SID with MPLS Support Programming using Flavors

[I-D.ietf-spring-srv6-network-programming] introduced the PSP, USP, and USD flavors for SRv6 SID. The U-FUNCT (U-ARG) and Flavors are combined to allocate different SRv6 SIDs, or someone can understand that each U-FUNCT (U-ARG) codepoint itself has a determined flavor. That is no problem for SRv6 SID allocation because IPv6 address resource is enough. For SR-MPLS over SRH in this document, a different MPLS label is used for each topology type of SID, such as node SID, adjacency SID, or service type of SID, etc. All these types of SIDs are equivalent to SIDs defined in SRv6. The label allocation is independent of flavors. For the use of the behavior flavor, an explicit standard Flavor codepoint could be set on the rightmost 12-bits of the SID entry (label) in SRH.

The codepoint can be used as U-FUNCT (U-ARG) to support the network programming. In this case, a 20-bits MPLS label of U-SID is interpreted as the U-LOC. The U-FUNCT in the codepoint field of U-SID can be advertised by the control plane.

Note that the flavor codepoint is different from the PHP flag of prefix-SID in SR-MPLS.

2.4. U-SID with SRv6 Programming process

Processing of SRH with elements carrying 32 bits-long SIDs closely follows SRH processing as defined in Section 4.3.1.1 [I-D.ietf-6man-segment-routing-header] and the "End" behavior is demonstrated in the pseudo-code below, but it equally applies to all SID behaviors. When N with U-SID receives a packet whose IPv6 DA is S and S is a local End SID. The lines S08 and S14 of the End processing which was, as per Section 4.1 of [I-D.ietf-spring-srv6-network-programming]:

```
S08.    max_LE = ( Hdr Ext Len * 8/ sizeof(SRH_element) ) - 1
        [...]
S14.    Get 128-bits IPv6 DA by 32-bits U-SID
                from Segment List[Segments Left]
        Update IPv6 DA
```

Note: S14. Obtaining 128-bits IPv6 DA from complementary U-SID can be done by mapping, stitching, shifting, translation, etc.

2.5. U-SID Complementary Method

The 32-bits U-SID MAY be used as complementary to a common IPv6 prefix to construct an IPv6 address SID (SRv6 SID). Many methods can be used to achieve that, including mapping, stitching, shifting, translation, etc.

Generally speaking, the relationship between 32-bits U-SID and 128-bits SRv6 SID can be established using any transformation function as long as the relationship unambiguous and reversible, i.e., there exists a transformation function that when being applied to the result produces the original value. We can use a function F as a method that produces 32-bits U-SID from 128-bits SRv6 SID. Then there must be a function F', used as the reversible method, to produce the original 128-bits SRv6 SID from the 32-bits U-SID. These functions are illustrated below:

```
U-SID = F (SRv6 SID);
SRv6 SID = F' (U-SID);
```

The details of these functions will be demonstrated in the future.

3. IANA Considerations

This draft has no requests for IANA actions. This section can be removed before the publication.

4. Security Considerations

TBD

5. Acknowledgements

TBD

6. 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-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-12 (work in progress), March 2020.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., Wan, X., Wei, C., and S. Shay, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-05 (work in progress), February 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>.

[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>.

Authors' Addresses

Cheng Weiqiang
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

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

Email: liu.aihua@zte.com.cn

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

Email: peng.shaofu@zte.com.cn