IDR Working Group   Yao. Liu
Internet-Draft   Shaofu. Peng
Intended status: Standards Track   ZTE Corporation
Expires: May 27, 2021   November 23, 2020

BGP Extensions for Unified SID in TE Policy
draft-liu-idr-segment-routing-te-policy-complement-04

Abstract

This document defines extensions to BGP in order to advertise Unified SIDs in SR-TE policies.

Status of This Memo

This Internet-Draft will expire on May 27, 2021.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing [RFC8402] leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions,called segments.

[I-D.ietf-spring-segment-routing-policy] details the concepts of SR Policy and steering into an SR Policy.

[I-D.ietf-idr-segment-routing-te-policy] specifies the way to use BGP to distribute one or more of the candidate paths of an SR Policy to the headend of that policy.

With increasing requirements for a shortened identifier in a segment routing network with the IPv6 data plane, [I-D.mirsky-6man-unified-id-sr] proposed 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.

This document defines extensions to BGP in order to advertise Unified SIDs in SR-TE policies.

Firstly, we focus on how to carry 32-bits IP address format U-SID, other type of U-SID (such as 16-bits) will be considered in future version.

2. SR policy with Unified SID

As discussed in [I-D.ietf-spring-srv6-network-programming], the node with the SRv6 capability will maintain its local SID table. A Local

SID is generally composed of two parts, that is, LOC:FUNCT, or may carry arguments at the same time, that is, LOC:FUNCT:ARGS.

FUNCT indicates the local function of the packet on the node that generates the LOC.ARGS may contain information related to traffic and services, or any other information required for executing the function.LOC indicates locator. In most cases, other nodes in the network can forward packets to the node that generates this LOC according to the corresponding routing table entries.

The controller plane protocol can also use B:N to represent an LOC, where B is SRv6 SID Locator Block and N to represent node N. In other words, the structure of a complete SID is B:N:FUNCT:ARGS.

[I-D.ietf-lsr-isis-srv6-extensions] defines the extension of ISIS to support SRv6, and each node can announce the SID assigned by itself. In particular, SRv6 SID Structure Sub-Sub-TLV is defined and the specific structure of the corresponding SID is provided, including the length of SRv6 SID Locator Block, the length of SRv6 SID Locator Node, the length of SRv6 SID Function, and the length of SRv6 SID Arguments.

Similarly, [I-D.ietf-bess-srv6-services] also provide the SID structure information for L3VPN or EVPN service related SID.

Thus, it can be seen that the existing control plane protocol reveals a very intuitive method to reduce the size of SRH. That is, under the specific address planning(the SIDs allocated by all SRv6 nodes are in the same SRv6 SID Locator Block), SRH only needs to store the difference between SIDs (N:FUNCT:ARGS), and does not need to contain the SRv6 SID Locator Block information. In a 128-bit classic SRv6 SID, the highest part is SRv6 SID Locator Block, and the following 32 bits are composed of SRv6 SID Locator Node, SRv6 SID Function and SRv6 SID Arguments, and the rest bits are zeros.

As for how to obtain the SRv6 SID Locator Block information during packet forwarding, there maybe three cases:

1)For the head-end node, when the node sends a packet along the segment list to the first segment, it already knows the 128-bit classical SID before truncating. The head node copies it directly to the DA of IPv6 Header, but the SRH carries the 32-bit truncatured SIDs.

2)For the normal transit node, it can obtain the SRv6 SID Locator Block information from the DA of the received IPv6 packet.

3)For the inter-domain border node, it can obtain the new SRv6 SID Locator Block information from the local SID entry.

2.1. Advertisement of SID Attribute

The U-SID solution defined in [I-D.mirsky-6man-unified-id-sr] reply two attributes of SID, they are: SID structure attribute and Endpoint Behavior attribute. However, [I-D.ietf-idr-segment-routing-te-policy] does not provide these information now. This document discusses two options to supplement these information.

2.1.1. Option 1: Advertising SID Attribute within existing sub-TLVs

In this section, a new sub-sub-TLV is introduced in each segment sub-TLV(type B/I/J/K) [I-D.ietf-idr-segment-routing-te-policy] to offer the SID structure information.

Since the new compression information-related sub-sub-TLV is included in segment List sub-TLV, the meaning of the whole segment list will be changed, that is, the headend cannot regard this segment list as a classic segment list to process and encapsulate the classic 128 bit SRH. Therefore, the controller must know the compression capability supported by the head node when delivering SR policy to the it.

There are two ways to do this.

Opntion 1, negotiate of compression capacity through BGP session. The controller only sends the Segment List Sub-TLV with compression information to the BGP neighbors with compression capability.

It is necessary to consider the scenario with a route reflector. The BGP session is not directly established between the controller and the head node. One or more RT Extended Community can be carried in the SR policy UPDATE announcement message to contain the specific head node Router-ID information.

If the controller learns that the head node has the compression capability by some means (such as collecting through BGP-LS), but the RR does not have the abilities , then the controller can still choose to send to the RR according to the actual destination node notified by UPDATE.

If the reflector does not recognize the newly added sub-TLV / sub-sub-TLV compression information, it is necessary to decide whether to unconditionally transmit it to the head node according to the positive bit in the top-level TLV (that is, the Tunnel Encapsulation Attribute).

If the reflector recognizes the newly added sub-TLV / sub-sub-TLV, it is necessary to check whether the headend has compression capability. If not, RR will not reflect the Segment List Sub-TLV containing compressed information to the head node.

Opntion 2, the controller collects the compression capability of the head node through BGP-LS. If the head node has compression capability, the controller can deliver an segment list Sub-TLV containing compression information to the head node. Otherwise, only an Segment List sub-TLV containing 128-bit SIDs can be delivered.

If there is a RR, it only needs to decide whether to transmit it unconditionally to the head node according to transitive bit in the top-level TLV (that is, Tunnel Encapsulation Attribute).

The first method is too complicated, so option 2 is recommended.

Figure 2 uses the type B segment sub-TLV as an example, other types of segment sub-TLV are similar.

The SRv6 SID Structure Sub-Sub-TLV has the following format:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|sub-Type=STRUCT|   Length      |     Count     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  BL of SID 1  |  TL of SID 1  |  BL of SID 2  |  TL of SID 1  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ... ...                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            ... ...            |  BL of SID N  |  TL of SID N  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SRv6 SID Structure Sub-Sub-TLV format

where,

Count: 1 octet, the count of segments.The value of count must be consistent with the number of Segment Sub-TLV contained in segment list sub-TLV; otherwise, the whole segment list sub-TLV must be ignored.

BL: block length of classical 128 bit SID in bits, value: 1~ 128. If the corresponding SID is an MPLS label, BL is 0.

TL: truncated length of the compressed SID in bits, value: 1~ 128. If a 128 bit SID is compressed to 32 bits, TL is 32. If a 128 bit SID is not compressed, TL is 128. the TL of a 32-bit MPLS label is 32.

As above, if the headend does not recognize the Segment Truncated sub-TLV, the entire Segment List sub-TLV must be ignored.

A new flag is introduced in Segment Sub-TLV,

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |V|A|     |UET| |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                       SRv6 SID (16 octets)                   //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: UET-Flavor Flag in Segment sub-TLV

where,

UET: U-SID Encapsulation Type Flag, 2-bit field, it indicates the UET type of the next SID, in other words, indicates the UET domain constructed by the current segment node and the next segment node. It could be the following values:

0: the UET domain following the current segment node is UET-128 domain, that means the next SID does not need compression and remains 128 bits.

1: the UET domain following the current segment node is UET-32 domain, that means the next SID needs to be compressed to a 32-bit IP address.

2: the UET domain following the current segment node is UET-32-MPLS domain, that means the next SID needs to be compressed to a 32-bit MPLS Label.

3: the UET domain following the current segment node is UET-16 domain, that means the next SID needs to be compressed to a 16-bit IP address.

Currently, if only the 32-bit IP address compression mode is considered, the UET-Flavor value is 0 or 1.

Two factors should be considered for the value of UET-Flavor: the compression domain type composed of this segment node and the next segment node, and the structure of the next selected SID can be compressed in the way indicated by UET-Flavor.

2.1.2. Option 2: Introducing a new U-segment list sub-TLV

In order to solve the forward compatibility problem more conveniently, a new U-Segment List Sub-TLV is defined, which can contain compressed information.

Similarly, the controller decides whether to send U-Segment List Sub-TLV with compression information or Segment List Sub-TLV without compression information to the head node based on whether the head node has the compression capability.

The specific extension:

1) Add a UET-Flavor Flag to the existing Segment Sub-TLV.

2) In U-Segment List Sub-TLV, define Segment Truncated sub-TLV to describe the compression result.

2.2. Controller Processing

Controller can collect UET capability information of all nodes, see [I-D.mirsky-6man-unified-id-sr], each node can support one or more than one UET capabilities. In general, a border node that belongs to multiple UET domain will support multiple UET capabilities, while other nodes can only support a single UET capability.

Controller can also collect SID per UET of all nodes. If a node support an UET capability, it will also allocate related SIDs for this UET Flavor.

When controller computed an SR path, it can check the UET capability of each segment node within the segment list, to outline which UET domains the SR path crosses. For example, from Headend H to endpoint E, a segment list <X1, X2, X3, B, Y1, Y2, Y3, E> may cross two UET domains, the node H, X1, X2, X3, B all support UET-1, and the node B, Y1, Y2, Y3, E all support UET-2. In this case, the FSU-flag will be set to UET-1, it indicates the UET domian which the first SID X1 belongs to. At the same time, the controller will select UET related SID for each segment according to the UET domain which the segment node belongs to, i.e., the UET Flag of SID X1, X2, X3 will be set to UET-1, and the UET Flag of SID B, Y1, Y2, Y3, E will be se to UET-2. Note that in this case, SID B with UET-2 Flavor, but not UET-1 Flavor, is inserted in ths list for the purpose of seamless splicing.

Then, controller need to check the structure information of each selected SID, to ensure they can safely construct an SID list with UET information. For example, the structure information of SID X1 (with UET-1 Flavor), SID X2 (with UET-1 Flavor), SID X3 (with UET-1

Flavor), SID B (with UET-2 Flavor), MUST support to get UET-1 (because the UET of prev SID is UET-1) related truncated piece information (Node:Func:ARGS) from the original IPv6 SID. Similarly, the structure information of SID Y1 (with UET-2 Flavor), SID Y2 (with UET-2 Flavor), SID Y3 (with UET-2 Flavor), SID E (with UET-2 Flavor), MUST support to get UET-2 (because the UET of prev SID is UET-2) related truncated piece information from the original IPv6 SID.

There maybe another segment list example, <B, Y1, Y2, Y3, E> also cross two UET domains, that is, the node H, B all support UET-1, and the node B, Y1, Y2, Y3, E all support UET-2. In this case, the FSU-flag will be also set to UET-1, it indicates the UET domian which the first SID B belongs to. At the same time, the controller will select UET related SID for each segment according to the UET domain which the segment node belongs to, i.e., the UET Flag of SID B, Y1, Y2, Y3, E will be se to UET-2. Note that in this case, SID B with UET-2 Flavor, but not UET-1 Flavor, is inserted in ths list for the purpose of seamless splicing. Then, the controller check the structure information of each selected SID to ensure they can safely construct an SID list with UET information. That is, the structure information of SID B (with UET-2 Flavor), MUST support to get UET-1 (because the UET of prev SID is UET-1) related truncated piece information from the original IPv6 SID. Similarly, the structure information of SID Y1 (with UET-2 Flavor), SID Y2 (with UET-2 Flavor), SID Y3 (with UET-2 Flavor), SID E (with UET-2 Flavor), MUST support to get UET-2 (because the UET of prev SID is UET-2) related truncated piece information from the original IPv6 SID.

If a SID can not support to get UET related truncated piece according to the UET of prev SID, the controller MUST select another prev SID with UET-0 flavor.

2.3. Headend Processing

When the headend receives the SR policy, it obtains the compressed information of each SID according to the TL field in the Segment Truncated sub-TLV. The headend should identify the UET-Flavor of each SID, which can be verified with the compression result, that is, the UET-Flavor of a certain SID must be consistent with the compression result of the next SID, otherwise the entire Segment List sub-TLV must be ignored .

In particular, the UET-Flavor of the last SID can be used as a clear basis to decide what compression method should be adopted for the overlay SID, such as the VPN service.

Optionally, the headend can use reduced SRH that exclude the first SID, to further reduce the cost of SRH.

3. Security Considerations

Procedures and protocol extensions defined in this document do not affect the security considerations discussed in [I-D.ietf-idr-segment-routing-te-policy].

4. IANA Considerations

TBD

5. References

5.1. Normative References

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-11 (work in progress), November 2020.

[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-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 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-24 (work in progress), October 2020.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., and G. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-07 (work in progress), July 2020.

[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>.

5.2. Informative References

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

Authors' Addresses

Liu Yao
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: liu.yao71@zte.com.cn

Peng Shaofu
ZTE Corporation
No. 50 Software Ave, Yuhuatai Distinct
Nanjing
China

Email: peng.shaofu@zte.com.cn