RIFT Working Group W. Cheng
Internet Draft China Mobile
Intended status: Standards Track C. Lin
Expires: September 03, 2025 New H3C Technologies
R. Wang
China Mobile
March 3, 2025

RIFT extensions for SRv6
draft-cheng-rift-srv6-extensions-04

Abstract

The Segment Routing (SR) architecture allows a flexible definition of the end-to-end path by encoding it as a sequence of topological elements called segments. It can be implemented over an MPLS or IPv6 data plane. This document describes the RIFT extensions required to support Segment Routing over the IPv6 data plane (SRv6).

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Segment Routing (SR) architecture [RFC8402] specifies how a node can steer a packet using an ordered list of instructions, called segments. These segments are identified using Segment Identifiers (SIDs).

Segment Routing can be instantiated on the IPv6 data plane through the use of the Segment Routing Header (SRH) defined in [RFC8754].Segment Routing instantiation on the IPv6 dataplane is referred to as SRv6.

The network programming paradigm for SRv6 is specified in [RFC8986].It describes how any behavior can be bound to a SID and how any network program can be expressed as a combination of SIDs. It also describes several well-known behaviors that can be bound to SRv6 SIDs.

Using SRv6 in data center networks brings several advantages:

1) Improved Scalability: SRv6 simplifies the network architecture by reducing the amount of state that must be maintained in network devices, which leads to increased scale and reduced management overhead.

2) Traffic Engineering: SRv6 enables more granular control over traffic paths within the network, allowing operators to direct traffic along specific paths to better utilize network resources and reduce congestion.

3) Service Function Chaining: With SRv6, network operators can easily define paths that traverse multiple service functions, including firewalls, load balancers and other middleboxes. This enables the creation of more flexible and scalable service chains to support a wide range of applications.

4) Path Segment Identification: SRv6 allows for the identification of specific path segments within the network, making troubleshooting and fault isolation simpler and faster.

5) Simplified Tunneling: SRv6 supports tunneling for traffic across an IPv6 network without the need for an IP-in-IP or GRE encapsulation. This simplifies the network architecture and reduces complexity.

This document describes the RIFT extensions required to support Segment Routing over the IPv6 data plane (SRv6).

At a high level, the extensions to RIFT are comprised of the following:

1) Add new SRv6 KV TIE using Well-Known KV TIE that is used to advertise an SRv6 Locator, its attributes, and SIDs.

2) Add new SRv6 Locator TIE, which is used to advertise Locator information southbound and northbound, as well as End SID information.

3) In the neighbor information of Node TIE, Add the new End.X SID information in the neighbor information of Node TIE to advertise the corresponding connection of End.X.

2. SRv6 KV TIE

For RIFT network, Each node is provisioned with a Locator. The Locator address are communicated to each node out-of-band and stored as configuration information. Communication could be done via primitive pen and paper or via modern signaling (Netconf/YANG) from a configuration management system. Then the node can use his own locator address to generate End-SID, End.X-SID etc.

To support Zero Touch Provisioning (ZTP), we can define a flexible Key-Value (K-V) format for transmitting SRv6 information such as SRv6 Locator information in RIFT SRv6 network, in order to enable TOF to pass the Locator address information to all nodes via KV-TIE. In this way, TOF can package the Locator address information into KV-TIE and transmit it to all nodes in the SRv6 network to support Zero Touch Provisioning. With this approach, all nodes in the network can obtain the Locator address information and use the addresses for path programming and other operations in the SRv6 network.

This section requests an entry from the RIFT Well-Known Key-Type Registry for networks that use SRv6 along with suggested values inaccordance with RIFT-KV-REGISTRY [RIFT-KV-REGISTRY].

```
 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              |   Key-Len     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             "SRv6 Locator" + AdvSystemID + SystemID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             SRv6 Locator Entry                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             SRv6 End.X Entry                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: SRv6 Locator Key/Value Pair

where:

Type: SRv6, TBD.

"SRv6 Locator": indicate that this is a Locator KV TIE.
AdvSystemID: The Advertiser’s 64-bit RIFT System ID. Typically, only the TOF publishes SRv6 KV TIEs, using its SystemID as the AdvSystemID.
System ID: A node’s 64-bit RIFT System ID.
SRv6 Locator Entry: A node’s SRv6 Locator Entry.

The SRv6 Locator Entry has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Metric                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Loc Size     |                     Resv                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//              Locator (continued, variable)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sub-TLVs (variable) . . .              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SRv6 Locator Entry

o Metric: 4 octets, as described in Section 4 of [RFC5305].

o Loc-Size: 1 octet. Number of bits in the SRv6 Locator field, which MUST be from the range (1-128). The entire TLV MUST beignored If the Loc-Size is outside this range.

o Resv: 3 octets, Not defined.

o Locator: 1-16 octets. This field encodes the advertised SRv6 Locator. The SRv6 Locator is encoded in the minimal number of octets for the given number of bits. Trailing bits MUST be set to zero and ignored when received.

o Optional Sub-TLVs: The SRv6 SID LBLN Information and SRv6 End SID Entry SHOULD be included in the Locator Entry.

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    LB Length  |  LN Length    | Fun. Length   |  Arg. Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: SRv6 SID LBLN Information

Where:

LB Length: 1-octet field. SRv6 SID Locator Block length in bits.

LN Length: 1-octet bit field. SRv6 SID Locator Node length in bits.

Function Length: 1-octet field. SRv6 SID Function length in bits.

Argument Length: 1-octet field. SRv6 SID Argument length in bits.

The SRv6 End SID Entry has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Endpoint Behavior       |      Flags                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SID (128 bits) . . .                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SID (cont . . .)                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SID (cont . . .)                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SID (cont . . .)                                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: SRv6 End SID Entry

where:

o Flags: 2 octet. No flags are currently defined. All bits are reserved for future use. They MUST be set to zero on transmission and MUST be ignored on receipt.

o Endpoint Behavior: 2 octets, as defined in [RFC8986].

o SID: 16 octets. This field encodes the advertised SRv6 SID.

The SRv6 End.X SID Entry has the following format:

```
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          link id                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          End.X SID                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: SRv6 End.X Entry in SRv6 Locator KV TIE

```
struct Srv6EndXSidElement {
   1: required common.LinkIDPair link_id,
   2: required Srv6SidElement srv6_endx_sid,
}
```

The Format of End.X SID:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |   Flags       |   Algorithm   |   Weight      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |        Endpoint Behavior      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | SID (128 bits) . . .                                          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | SID (cont . . .)                                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | SID (cont . . .)                                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | SID (cont . . .)                                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | SRv6 SID LBLN Information (optional)                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Format of End.X SID

where:

Flags: 1 octet.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|B|S|P|Reserved |
+-+-+-+-+-+-+-+-+
```

where:

B-Flag: Backup flag. If set, the SID is eligible for protection, e.g., using IP Fast Reroute (IPFRR) [RFC5286] as described in [RFC8355].

S-Flag: Set flag. When set, the S-flag indicates that the SID refers to a set of adjacencies (and therefore MAY be assigned to other adjacencies as well).

P-Flag: Persistent flag. When set, the P-flag indicates that the SID is persistently allocated, i.e., the SID value remains consistent across router restart and/or interface flap.

Reserved bits: Reserved bits MUST be zero when originated and MUST be ignored when received.

Algorithm: 1 octet, as defined in the "IGP Algorithm Types" registry [RFC8665].

Weight: 1 octet. The value represents the weight of the SID for the purpose of load balancing. The use of the weight is defined in [RFC8402].

Endpoint Behavior: 2 octets, as defined in [RFC8986].

SID: 16 octets. This field encodes the advertised SRv6 SID.

SRv6 SID LBLN Information: describe SRv6 SID LBLN Information.

3. SRV6 Locator TIE

In the RIFT network, in order to support SRv6, SRv6 Locator need to be distributed in SRv6 Locator TIEs.

The Locator address and End-SID advertised in SRv6 Locator TIE.

When a neighbor node receives a Locator Address Advertisement in SRv6 Locator TIE, it adds a route for the Locator address to its routing table.

The format of the SRv6 Locator TLV is the same as the SRv6 Locator TLV in SRv6 KV TIE described in Section 2 of this document.

When supporting the compression mode, the SRv6 SID LBLN Information needs to be carried in the End-SID information. The format of SRv6 SID LBLN Information is described in Figure 3.

Name	Value	Schema Version	Description
Illegal	0	6.1	

TIETypeMinValue	1	6.1	
NodeTIEType	2	6.1	
PrefixTIEType	3	6.1	
...			
LocatorTIEType	TBD	6.1	This Document

```
/** Description of a Locator TIE. */

struct LocatorTIEElement {

   /* SRv6 Locator information */

   1: common_srv6.SRv6Locator srv6_locator;

}
```

4. Advertise End.X SID in Node Neighbor

End.X SID is advertised via Node Neighbor in Node TIE, where the Neighbor information is extended in the Expanded Neighbor section.

In Registry RIFT_v6/encoding/NodeNeighborsTIEElement:

Name	Value	Schema Version	Description
level	1	6.1	Level of neighbor
cost	3	6.1	Cost to neighbor. Ignore anything larger than ‘infinite_distance‘ and ‘invalid_distance‘
link_ids	4	6.1	Can carry description of multiple parallel links in a TIE
bandwidth	5	6.1	Total bandwith to neighbor as sum of all parallel links
End.X	TBD	6.1	SRv6 End.X SID

Table 2: Requested Neighbors TIE Element

The format of End.X SID is as Figure 6 described.

5. Example

In a RIFT network, SRv6 is used to enable path computation and traffic engineering. The RIFT control plane uses SRv6 Segment Routing to program paths through the network. Each node in the network is assigned a unique IPv6 SID, which is used to represent the node and its attached topology. When a source node wants to send traffic to a destination node, it simply specifies a list of SIDs that correspond to the nodes that the traffic must traverse. Intermediate nodes forward the traffic based on the specified SIDs, ensuring that it follows the desired path. SRv6 is used in conjunction with RIFT’s Topology Information Base (TIB) to enable efficient path computation and fast rerouting in the event of a topology change. With SRv6, RIFT networks can support end-to-end traffic engineering, service chaining, and other advanced network functions, while also providing fast and efficient routing within the network.

Assuming the Locator Block is 2001:0db8::/32, denoted as LB; ToF1’s Locator is 2001:0db8:1::/64, denoted as LB:1::/64, NodeSid is 2001:0db8:1::1, denoted as LB:1::1; other nodes follow the similar pattern.

```
             +----------------+      +----------------+     Level 2
             |ToF 1           |      |ToF 2           |
             |NodeSid: LB:1:: |      |NodeSid: LB:2:: |
             |Loc: LB:1::/64  |      |Loc: LB:2::/64  |
             +---+--+--+--+---+      +-+--+---+--+--+--+
                 |  |  |  |            |  |   |  |
                 |  |  |  |            |  |   |  |
                 |  |  |  |            |  |   |  |
     +-----------+  |  +--)------------)--)-+ |  +------------+
     |              |     |            |  | | |               |
     |    +---------)-----)------------+  | | |               |
     |    |         |     |               | | |               |
     |    |         |     +-----------------)-- -----------+  |
     |    |         |                     | | |            |  |
     |    |         +--+     +------------+ | |            |  |
     |    |            |     |              | |  Level 1   |  |
   +-+----+-----+  +---+-----+---+  +-------+-+--+  +------+--+--+
   |Spin11      |  |Spin12       |  |Spin21      |  |Spin22      |
   |NodeSid:    |  |NodeSid:     |  |NodeSid:    |  |NodeSid:    |
   |  LB:3::    |  |   LB:4::    |  |     LB:5:: |  |    LB:6::  |
   |Loc:        |  |Loc:         |  |Loc:        |  |Loc:        |
   |  LB:3::/64 |  |   LB:4::/64 |  |   LB:5::/64|  |   LB:6::/64|
   +--+---+-----+  +---+-----+---+  +--+---+-----+  +---+-----+--+
      |   |            |      |        |   |            |       |
      |   +------------)-+    |        |   +------------)-+     |
      |                | |    |        |                | |     |
      |   +------------+ |    |        |    +-----------+ |     |
      |   |         |    |    |        |    |    Level 0  |     |
   +--+---+-----+   +----+----+--+   +-+----+-----+   +---+-----+----+
   |Leaf11      |   |Leaf12      |   |Leaf21      |   |Leaf22        |
   |NodeSid:    |   |NodeSid:    |   |NodeSid:    |   |NodeSid:      |
   |   LB:7::   |   |   LB:8::   |   |     LB:9:: |   |     LB:A::   |
   |Loc:        |   |Loc:        |   |Loc:        |   |Loc:          |
       LB:7::/64|   |   LB:8::/64|   |  LB:9::/64 |   |    LB:A::/64 |
   +--+---------+   +-+----------+   +-+----------+   ++--------------+
      +               +                +               +
      Prefix11          Prefix12           Prefix21          Prefix22
```

Figure 6: SRv6 information in RIFT network

During network initialization, the controller distributes all SRv6 Locator configurations to ToF. Then ToF generates SRv6 KV TIE for SRV6 Locator and extends it to all nodes by ZTP. After receiving this information, each node generates its own SRv6 Locator configurations.

Each node generates its Node SID based on its own SRv6 Locator and advertise its SRv6 Locator and Node SID information through Node TIE.

After establishing neighbor relationships with other adjacent nodes, the node uses the SRv6 Locator information to generate End.X SID and carries it in the Neighbors information of Node TIE to be transmitted out.

Assuming Leaf11 sends a packet to Leaf21, when congestion occurs on the westward link of TOF1, the controller can specify the packet path NodeSid_Spin12, NodeSid_TOF2, NodeSid_Spin21, NodeSid_Leaf21, for Leaf11, Leaf11 add an SRv6 header with a Segment List of (SA,DA) (2001:0db8:9::, 2001:0db8:5::, 2001:0db8:2::, 2001:0db8:4::, SL=3), so that the packet will pass through Leaf11, Spin12, ToF2, Spin21, Leaf21 node in order.

When compression mode is supported, it is not necessary to add the SRv6 header, and route can be arranged through the destination address. In the path mentioned above, using compression mode, the destination address is 2001:0db8:0004:0002:0005:0009:0000:0000 The destination address generated through this compression method is called C-SID.

When there are multiple paths available, Controller can select a specific path by specifying the End.X SID.

6. Security Considerations

TBD.

7. IANA Considerations

7.1. SRv6 Locator KV TIE

This document requests an entry from the RIFT Key-Types Registry for Locator KV TIE in accordance with "RIFT Key-Types" registry [RIFT-KV-REGISTRY]

Value	Key-Type	Description	Status/ Reference
0	Illegal	Not allowed.	
1	Experimental	Indicates that the Key-Type is Experimental.	

2	Well-Known	Indicates that the Key-Type is Well-Known.	
3	OUI	Indicates that the Key-Type is OUI (vendor specific).	
TBD	SRv6	Indicates that the Key-Type is SRv6.	This document

Table 3: Requested Key-Type

SRv6-Entry-Type in RIFT SRv6-Locator-Entry:

Name	Value	Schema Version	Description
SRv6Locator	1	6.1	This Document
End.X	2	6.1	This Document

7.2. SRv6 Locator TIE

This document makes the following registration in the " RIFT/common/TIETypeType" registry:

Name	Value	Schema Version	Description
srv6_locator	TBD	8.0	This document

Table 3: IANA Requested srv6 locator TIE

7.3. SRv6 End.X SID

This document makes the following registration in the"NodeNeighborsTIEElement" registry of a Node TIE:

Value	Name

TBD	SRv6 End.X SID

Table 4: IANA Requested Neighbours TIE Element

8. References

8.1. Normative References

[draft-ietf-rift-sr] Z. Zhang,"SRIFT: Segment Routing in Fat Trees", 12 January 2023,<https://www.ietf.org/archive/id/draft-ietf-rift-sr-00.txt>.

[draft-ietf-rift-rift] A. Przygienda, Ed., "RIFT: Routing in Fat Trees", 14 September 2023, <https://www.ietf.org/archive/id/draft-ietf-rift-rift-17.txt>

[RFC8665], P. Psenak, Ed., "OSPF Extensions for Segment Routing", December 2019, <https://datatracker.ietf.org/doc/rfc8665>

[RFC8402], C. Filsfils, Ed., "Segment Routing Architecture", July 2018, <https://datatracker.ietf.org/doc/rfc8402/>

[RFC8754], C. Filsfils, Ed., "IPv6 Segment Routing Header (SRH)", March 2020, <https://datatracker.ietf.org/doc/rfc8754/>

[RFC8986], C. Filsfils, Ed., "Segment Routing over IPv6 (SRv6) Network Programming", February 2021, <https://datatracker.ietf.org/doc/rfc8986/>

[RFC5305], T. Li., "SIS-IS Extensions for Traffic Engineering", October 2008, <https://datatracker.ietf.org/doc/rfc5305/>

[RFC5286], A. Atlas, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", September 2008, <https://datatracker.ietf.org/doc/rfc5286/>

[RFC8355], C. Filsfils, Ed., "Resiliency Use Cases in Source Packet Routing in Networking (SPRING) Networks", March 2018, <https://datatracker.ietf.org/doc/rfc8355/>

[draft-ietf-lsr-ospfv3-srv6-extensions], Z. Li, "OSPFv3 Extensions for SRv6", 8 June 2023, < https://datatracker.ietf.org/doc/draft-ietf-lsr-ospfv3-srv6-extensions/ >

Appendix A. Thrift Models

This section contains the normative Thrift models required to support SRv6. Per the main RIFT [RIFT] specification, all signed values MUST be interpreted as unsigned values.

A.1. common.thrift

This section specifies extensions to RIFT common.thrift model.

These extensions are REQUIRED in order to support SRv6.

TBD.

```
/** Type of TIE.

*/

enum TIETypeType {
```

```
        Illegal                                      = 0,
        TIETypeMinValue                              = 1,
        /** first legal value */
        NodeTIEType                                  = 2,
        PrefixTIEType                                = 3,
        PositiveDisaggregationPrefixTIEType          = 4,
        NegativeDisaggregationPrefixTIEType          = 5,
        PGPrefixTIEType                              = 6,
        KeyValueTIEType                              = 7,
        ExternalPrefixTIEType                        = 8,
        PositiveExternalDisaggregationPrefixTIEType = 9,
        Srv6LocatorTIEType                           = TBD,
        TIETypeMaxValue                              = TBD+1,
    }

    enum   KVTypes {
        Experimental = 1,
        WellKnown    = 2,
        OUI          = 3,
        Srv6         = TBD,
    }
```

A.2. encoding.thrift

This section specifies extensions to RIFT encoding.thrift model.

These extensions are REQUIRED in order to support SRv6.

```
   /** Description of a Locator TIE. */

   struct LocatorTIEElement {

      1: SRv6Locator srv6_locator;

   }

/* neighbor of a node */
struct  NodeNeihborsTIEElement {
   ...

   /* endx sid of neighbor */

   6: optional common_srv6.Srv6SidElement endx,

}
```

A.3. common_srv6.thrift

```
Struct Srv6LBLNElement {
    1: required i8 lb_len,
    2: required i8 ln_len,
    3: required i8 func_len,
    4: required i8 arg_len,
}

struct Srv6SidElement {
   1: required i16 endpoint_behavior,
   2: required i16 flags,
   3: required common.Ipv6Addr sid_addr,
   4: optional Srv6LBLNElement lbln_attrib,
```

```
}

struct Srv6LocatorSubElement {
    /** metric */
    1: required   i32 metric,
    /** loc size*/
    2: require    i8  loc_size,
    /* locator address */
    3: required common.Ipv6Addr locator_addr,
}

struct SRv6LocatorSubTlv {
    /** locator */
    1: required   Srv6LocatorSubElement       srv6_locator,
    /*  lbln information of locator*/
    2: optional   Srv6LBLNElement             locator_lbln,
    /* node sid */
    3: optional   list<Srv6SidElement>     srv6_sid_list,
}

struct Srv6EndXSidElement {
   1: required common.LinkIDPair link_id,
   2: required Srv6SidElement srv6_endx_sid,
}

struct SRv6Locator {
    /* locator info */
    1: required   SRv6LocatorSubTlv            srv6_locator,
    2: optional   list<Srv6EndXSidElement> srv6_endx_sid_list,
}
```

Authors' Addresses

Weiqiang Cheng
China Mobile
China

Email: chengweiqiang@chinamobile.com

Changwang Lin
New H3C Technologies
China

Email: linchangwang.04414@h3c.com

Ruixue Wang
China Mobile
China

Email: wangruixue@chinamobile.com