         Generalized SRv6 Network Programming for SRv6 Compression
                 draft-cl-spring-generalized-srv6-for-cmpr-02

Abstract

   This document proposes Generalized Segment Routing over IPv6 (G-SRv6)
   Networking Programming for SRv6 compression.

   G-SRv6 can reduce the overhead of SRv6 by encoding the Generalized
   SIDs(G-SID) in SID list, and it also supports to program SRv6 SIDs
   and G-SIDs in a single SRH to support incremental deployment and
   smooth upgrade.

   G-SRv6 is fully compatible with SRv6 with no modification of SRH, no
   new address consumption, no new route creation, and even no
   modification of control plane.

   G-SRv6 for Compression is designed based on the Compressed SRv6
   Segment List Encoding in SRH
   [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] framework.

Status of This Memo

   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 May 20, 2021.

Copyright Notice

Table of Contents

1. Introduction

   Segment routing (SR) [RFC8402] is a source routing paradigm that
   explicitly indicates the forwarding path for packets at the ingress
   node by inserting an ordered list of instructions, called segments.

   When segment routing is deployed on the IPv6 data plane, it is called
   SRv6 [RFC8754]. For support of SR, a new routing header called
   Segment Routing Header (SRH), which contains a list of SIDs and other
   information, has been defined in [RFC8754]. In use cases like
   Traffic Engineering, an ordered SID List with multiple SIDs is
   inserted into the SRH to steer packets along an explicit path.

   However, the size of SIDs (16 bytes per SID) in SRH proposes
   challenges for packet processing and payload efficiency
   [I-D.cheng-spring-shorter-srv6-sid-requirement]. In order to solve
   this problem, this document proposes Generalized Segment Routing over
   IPv6 (G-SRv6) Networking Programming for SRv6 compression.

   G-SRv6 supports to encode multiple types of Segments in an SRH,
   called Generalized SRH (G-SRH). In SRv6 Compression, the G-SRH can
   carry multiple SRv6 SID and G-SID(Generalized Segment Identifier)
   containers in the SID list. A G-SID container may include an SRv6
   SID or multiple G-SIDs and optional padding. A G-SID can be a
   32-bits value of the original SRv6 SID, which contains the node ID
   and function ID. By carrying G-SIDs instead of 128 bits SRv6 SID,
   the problem of SRv6 header size can be solved, and the solution is
   compatible with SRv6.

2. Terminology

   This document makes use of the terms defined in [RFC8754], [RFC8402]
   and [RFC8200], and the reader is assumed to be familiar with that
   terminology. This document introduces the following terms:

   Compressible SRv6 SID: It is the 128-bit SRv6 SID whose format can be
   compressed. It is composed by Common Prefix and Generalized Segment
   Identifier (G-SID) and optional arguments and padding.

   Common Prefix: It is the same prefix shared by multiple SIDs.

   G-SRv6: Generalized SRv6 Network Programming

   G-SRH: Generalized Segment Routing Header. It keeps the same format
   and code point with original SRH, which can carry multiple G-SIDs and
   original SIDs.

G-SID: Generalized Segment Identifier.It is a Compressed SID(C-SID)
[I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc].

G-SID Container: Generalized Segment Identifier Container.It is a
C-SID container [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc].

2.1. 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.

3. Concepts of G-SRv6

This section describes the concepts of G-SRv6.

3.1. G-SID

In an SRv6 domain, the SIDs are allocated from an address block,
called SID space. Therefore, the SIDs allocated from the same SID
space share the common prefix. Also, if the length of the SID is
less than 128 bits, then padding is required. In an SID List, the
common prefix and padding are redundant. Reducing the redundant
information can reduce the overhead of SRv6.

This document defines a Generalized SID (G-SID) to carry the
different part of the original SRv6 SID in the SRH to reduce the size
of the SRH. The G-SID can be a 32-bits value following the common
prefix in the original SRv6 SID. An SRv6 SID with this format is
called compressible SRv6 SID. The format of a compressible SRv6 SID
with 32-bits G-SID is shown in Figure 1.

```
  0              Variable Length          32 bits             128 bits
   +-------------------------------------------------------------+
   |             Common Prefix    |  G-SID   | Args/padding  |
   +-------------------------------------------------------------+
  |<-------- Locator ---------------->|
```
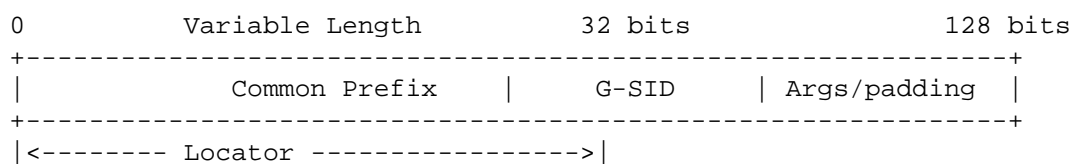
Figure 1. 32 bits G-SID in SRv6 SID

In order to indicate the format of the SRv6 SID is compressible,
control plane extension may be considered. This is out of scope of
this document, and can be described in other documents.

3.2. G-SID Container

   In order to align with 128 bits, a 128 bit G-SID Container is
   defined. A G-SID Container is a 128 bits value, and it may contain
   different type of SIDs:

   o an SRv6 SID: A G-SID Container contains a single SRv6 SID.

   o A Micro SID Carrier: A G-SID Container contains a Micro SID
      carrier [I-D.filsfils-spring-net-pgm-extension-srv6-usid].

   o Multiple G-SIDs: A G-SID Container contains multiple G-SIDs and
      optional padding. When G-SID is a 32-bits value, a G-SID
      Container can consist of 4 G-SIDs. If the length of G-SIDs in a
      G-SID Container is less than 128 bits, then padding is required.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 0                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 1                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 2                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 3                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                (a)

    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Padding                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Padding                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 0                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            G-SID 1                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                (b)
```

               Figure 2. G-SID Container for Compression

3.3. G-SID Index

   In order to locate the G-SID within the G-SID Container, this section
   defines Generalized SID Index (SI) to indicate the location of the
   G-SID within the current G-SID Container.

   SI is a location argument of the G-SID, which is the least bits in
   the argument part. When G-SID is a 32 bits value, the SI is the
   least 2 bits in Argument.

```
0            Variable Length          32 bits            128 bits
+---------------------------------------------------------------+
|            Common Prefix   | G-SID        |SI|  Padding  |
+---------------------------------------------------------------+
```
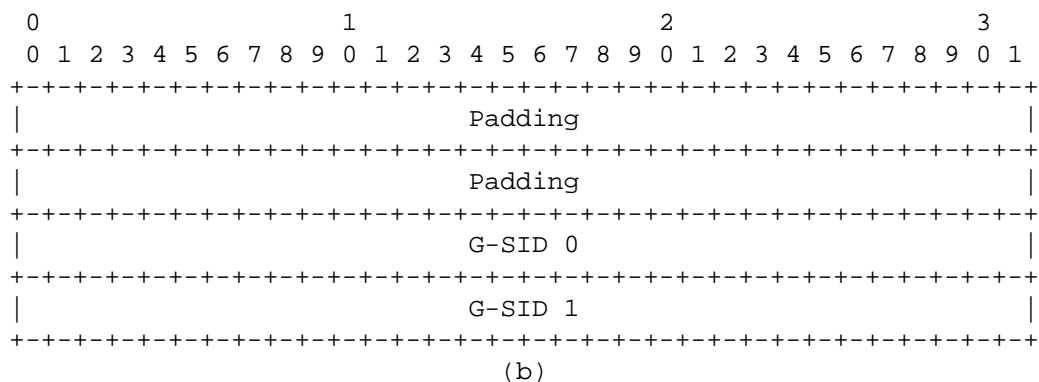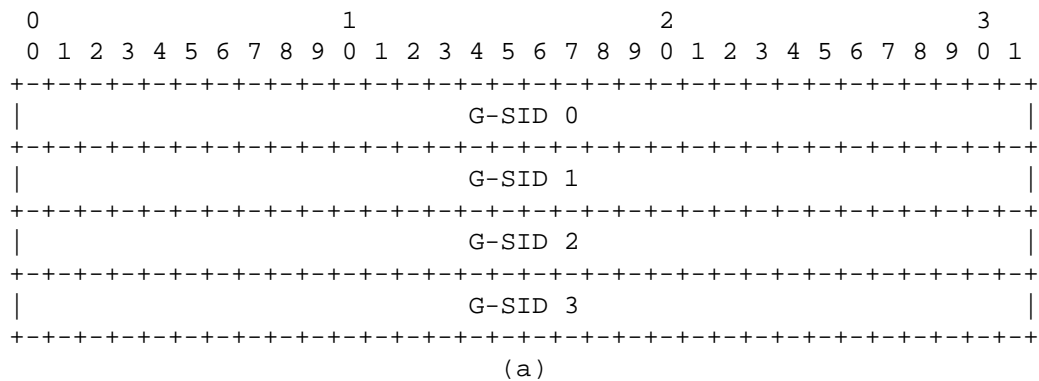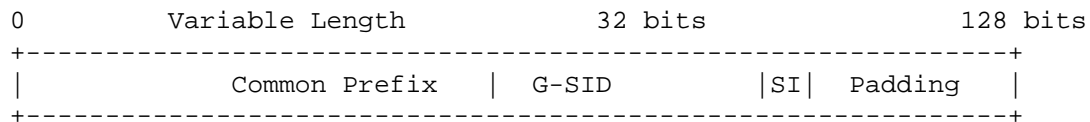
                      Figure 4. SI in the IPv6 DA

3.4. COC Flavor

   In order to indicate the SRv6 compression processing, updating the
   next 32-bits G-SID to the IPv6 DA, this section defines COC(Continue
   of Compression) Flavor.

   When a node receives an SID with COC Flavor, it indicates to update
   the G-SID part in IPv6 DA with the next 32 bits G-SID.

   When a node receives an SID without COC Flavor, the node processes
   the packet as a normal SRv6 packet
   [I-D.ietf-spring-srv6-network-programming], for example, update the
   IPv6 DA with the next 128 bits SID if SL >0.

   Therefore, if the behavior of the last G-SID in the G-SID list has no
   COC Flavor, then the next 128 bits SID will be updated to the DA, so
   it indicates the end of the compression sub-path.

   When COC Flavor applies to END, END.X and END.T, the SIDs can be
   advertised via the IS-IS [I-D.ietf-lsr-isis-srv6-extensions], and the
   SRv6 SID Structure Sub-Sub-TLV MUST be carried to indicate the format
   of the SRv6 SID. The Locator.Block length indicates the length of
   the common prefix, and the G-SID is the following 32-bits value after
   the Block, which contains the Node ID and Function ID.

4. G-SRH

   G-SRH supports to encode different types of segment in a single SRH
   without modifying the encapsulation format of SRH.

When an SRv6 path travels normal SRv6 nodes and compressed SRv6
nodes, the SRv6 SID and G-SIDs can be encoded in a single G-SRH.

For easier understanding, this document assumes that the Compressible
SRv6 SID consists of 64 bits common prefix and 32 bits G-SID. The
encoding can be shown as follows.

```
      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    |   Flag        |                Tag           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    Other G-SID Container                     |
     .                        ...                                   .
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    ---
     |                    Optional Padding                         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         G-SID 0                             | G-SID Container 0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     .                        ...                                   .      ...
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         G-SID 3                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    ---
     |                         G-SID 0                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         G-SID 1                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         G-SID 2                             | G-SID Container j
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         G-SID 3                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    ---
     |                     Common Prefix                           |
     |                                                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ G-SID Container k
     |                         G-SID 0                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                         Padding                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    ---
     |                        ...                                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                             |
     |       Generalized Segment List[n] (128 bits SRv6 SID)       |
     |                                                             |
     |                                                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     //         Optional Type Length Value objects (variable)      //
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
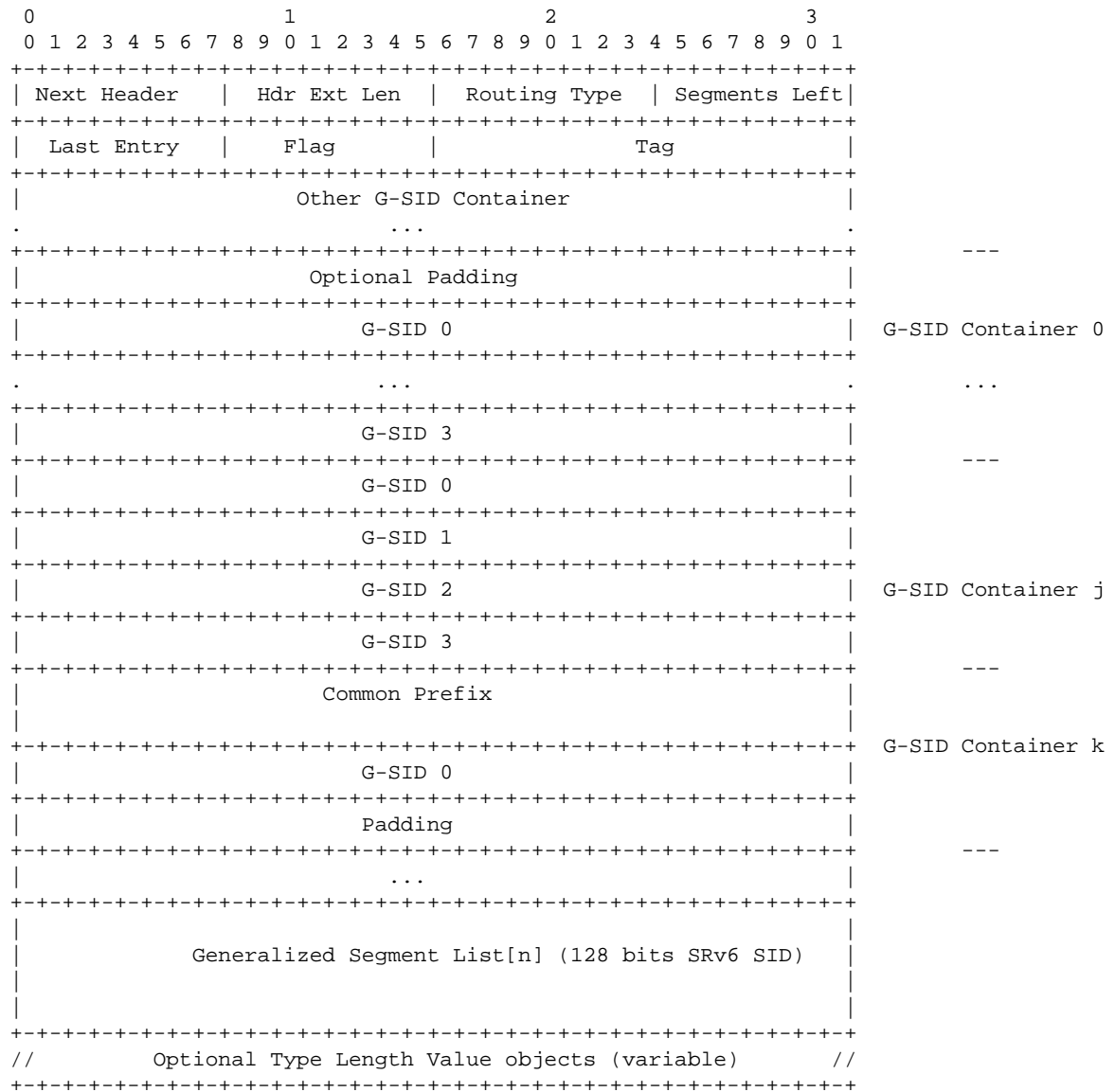
                   Figure 3. G-SRH for SRv6 Compression

     Where:

   o Common Prefix: the common prefix shared by the Compressible SRv6
      SIDs in the current compression sub-path. Usually, it is the
      prefix of the SID space, called Locator Block in control plane
      [I-D.ietf-spring-srv6-network-programming]. Operators are free to
      configure the length and the value of the common prefix based on
      the address planning of their networking.

   o G-SID: 32-bits Generalized SID.

   o Padding: Must be zero. When the length of G-SIDs within the G-SID
      Container is less than 128 bits, then padding is needed.

5. Packet Processing

   This section describes the pseudo code of COC Flavor, and it replaces
   the S13 and S14 of End, End.X, and End.T's pseudo code
   [I-D.ietf-spring-srv6-network-programming]. The pseudo code is shown
   below.

   When N receives a packet whose IPv6 DA is S and S is a local SID with
   COC Flavor, N does:

   1.   If (DA.SI != 0) {   //ref1
   2.   Decrement DA.SI by 1.
   3.   } Else {
   4.   Decrement Segments Left by 1.
   5.   Set DA.SI to 3 in the IPv6 Destination Address
   6.   }
   7.   Copy Segment List[Segments Left][DA.SI] into the bits //ref2
        [B..B+31] of the IPv6 Destination Address.

   o Ref1: an SID with COC flavor indicates the SRv6 compression
      processing that the node needs to update the next 32 bits G-SID to
      the IPv6 DA.

      * When the SI is greater than 0, the next G-SID is the next G-SID
         in the current G-SID Container.

      * Otherwise, the next G-SID is the first G-SID in the next G-SID
         Container.

   o Ref2: B is the length of the Locator Block
      [I-D.ietf-spring-srv6-network-programming].

   An SID without COC Flavor will be processed following the SRv6
   processing. The node will update the next 128 bit SID to the IPv6 DA
   if the SL > 0.

6. Illustration

   This section describes a simple example of G-SRv6 for compression.

   The reference topology is shown below.

```
               *-------------------*
               *     SRv6 Domain    *
               *                    *
     Tenant10 CE1--0-1-2-3-4-5-6-7-8-9-10--CE2  Tenant10
               *                    *
               *-------------------*
```
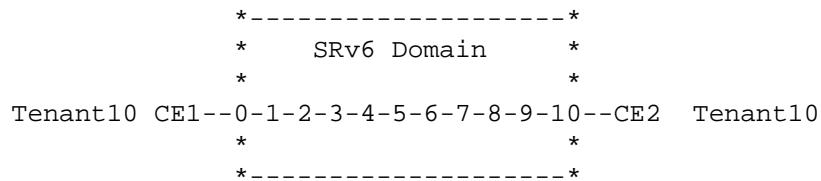
                    Figure 5. Reference topology

   Nodes 0 - 10 are G-SRv6 enabled nodes within the SRv6 domain, and
   node 0 is the ingress node of the G-SRv6 path while the node 10 is
   the egress node.

   Nodes CE1 and CE2 are tenants of VPN 10, and they are outside of the
   SRv6 domain.

   In order to ease the reading of the example, this section introduces
   a simplified SID allocation schema.

   o 2001:db8::/64 is dedicated to the internal SRv6 SID space, which
      is the common prefix for the SIDs as well.

   o Node k has 2001:db8:0:0:k::/80 for its local SID space. Its SIDs
      will be explicitly allocated from that block.

   o 2001:db8:0:0:k:1:: represents the End.X SID with COC allocated by
      node K, and it is associated with interface N of node K. For
      instance, 2001:db8:0:0:1:1:: represents the End.X with COC flavor
      allocated by node 1.

   o 2001:db8:0:0:k:2:: represents the End.X SID without COC allocated
      by node K, and it is associated with interface N of node K. For
      instance, 2001:db8:0:0:1:2:: represents the End.X without COC
      flavor allocated by node 1.

   o 2001:db8:0:0:10:10:: is an END.DT4 SID initiated by node 10, which
      is associated with the VRF10.

   Therefore, the SID 2001:db8:0:0:1:1::, 2001:db8:0:0:2:1::,
   2001:db8:0:0:3:1::, 2001:db8:0:0:4:1::, 2001:db8:0:0:5:1::,
   2001:db8:0:0:6:1::, 2001:db8:0:0:7:1::, 2001:db8:0:0:8:1:: are SRv6

End.X SIDs with COC Flavor, and 2001:db8:0:0:9:2:: is a Compressible
SRv6 End.X SID.

The SID list [2001:db8:0:0:1:1::, 2001:db8:0:0:2:1::,
2001:db8:0:0:3:1::, 2001:db8:0:0:4:1::, 2001:db8:0:0:5:1::,
2001:db8:0:0:6:1::, 2001:db8:0:0:7:1::, 2001:db8:0:0:8:1::,
2001:db8:0:0:9:2::, 2001:db8:0:0:10:10::] is calculated for a strict
TE path from Node 1 to Node 10 for the VPN traffic of tenant 10.

In G-SRv6, the SID list can be encoded as [2:1, 3:1, 4:1, 5:1, 6:1,
7:1, 8:1, 9:2, 2001:db8:0:0:10:10::] in reduced mode. The G-SID
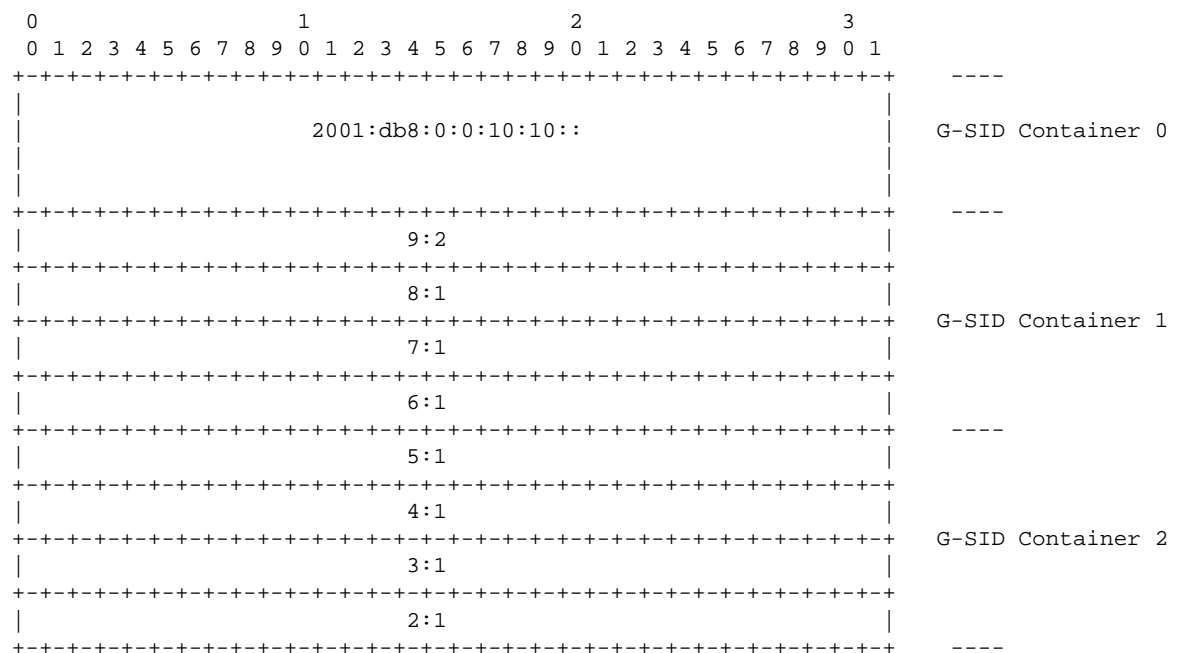Container encoding is shown below.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ----
 |                                                               |
 |                   2001:db8:0:0:10:10::                        | G-SID Container 0
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ----
 |                             9:2                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             8:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   G-SID Container 1
 |                             7:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             6:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ----
 |                             5:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             4:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   G-SID Container 2
 |                             3:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             2:1                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ----
```

Figure 6. G-SID Container Encoding for G-SRv6

The packets forwarding procedures:

o Node 0 sends the SRv6 packet with G-SRH to the node 1.The SL is 3.
   The Active SID in IPv6 DA is 2001:db8:0:0:1:1::.

o When node 1 receives the packet, the IPv6 DA is
   2001:db8:0:0:1:1::, which is a Local End.X with COC Flavor SID.

The SRH.SL is 3, and DA.SI is 0. The node processes the packet:
SL = SL-1,DA.SI =3, pointing to the next G-SID 2:1, and updates
SRH[SL=2][DA.SI=3] to the IPv6 DA[CP:CP+31], where CP is the
length of the common prefix. The packet is forwarded with the new
IPv6 DA 2001:db8:0:0:2:1:C::, to the node 2.

o When node 2 receives the packet, the IPv6 DA is
   2001:db8:0:0:2:1:C::, which is a Local End.X with COC Flavor SID.
   The SRH.SL is 2, and DA.SI is 3. The node processes the packet:
   DA.SI --, pointing to the next G-SID 3:1, and updates
   SRH[SL=2][DA.SI=2] to the IPv6 DA[CP:CP+31]. The packet is
   forwarded with the new IPv6 DA 2001:db8:0:0:3:1:8::, to the node
   3.

o Similar to node 1 and 2, the node 3,4,5,6,7,8 process the packet
   and forward with the new IPv6 DA.

o When node 9 receives the packet, the IPv6 DA is
   2001:db8:0:0:9:2::, which is a Local End.X SID. The SRH.SL is 1.
   The node updates the next SID 2001:db8:0:0:10:10:: to the IPv6 DA
   and forwards the packet to the node 10.

o Node 10 receives the packet, and the IPv6 DA is an VPN SID
   allocated by itself, the node processes the SRv6 VPN SID.

This illustration shows that 70 % overhead of SID list is removed in
G-SRv6(10 x 16 Bytes to 3 x 16 Bytes), also, it shows the
capabilities of encoding G-SIDs and SRv6 SIDs in a single G-SRH.

7. Benefits

o G-SRv6 is fully compatible with SRv6

   * No SRH encapsulation modification.

   * No new address consumption: Compressible SRv6 SIDs can be
      allocated from the Locator allocated to the node.

   * No new route advertisements: Compressible SRv6 SIDs can share
      the same locator with the normal SRv6 SID.

   * No security policy modification: when reusing the Locator with
      SRv6 SIDs, no security policy need to be updated.

   * No control plane modification: Controller can install the SR
      policy with 128-bits G-SID Containers, and the ingress treats
      the G-SID Container as an opaque 128-bits SID without
      understanding the structure of it. G-SRv6 capable nodes

        understand the COC flavor behaviors, while Compression disable
        SRv6 nodes are unaware of Compression.

   o G-SRv6 reduces the SRv6 encapsulation size.

      * 128 bits to 32 bits, up to 75 % overhead is reduced. More
         overhead is reduced when the G-SID is a 16-bits value.

   o G-SRv6 has efficient address consumption and easy to deploy

      * Operators are free to allocate an SID space from their address
         space.

      * No affect of networking(i.e. routes and ACL security policies)
         by using the existing Locator to allocate compressible SRv6
         SIDs.

   o G-SRv6 is hardware friendly

      * Same SRv6 processing flow with a new IPv6 DA update method

      * Leverages the mature hardware capabilities (DA update, DA
         longest match)

      * Avoids extra lookup in indexed mapping table

   o G-SRv6 supports incremental deployments, which can be deployed on
      demand.

8. Protocol Extensions Requirements

   This section describes the protocol extension requirements.

8.1. Data Plane

   REQ1-01: An SRv6 compression path can be represented as a G-SID
   Container list consists of a compressible SRv6 SID and G-SID
   Containers.

   REQ1-02: A G-SID Container consists of at most 4 (32-bits) G-SIDs, if
   the number of G-SID is less than 4, then padding is required to align
   with 128 bits.

   REQ1-03: If the first Compressible SRv6 SID is copied to the IPv6 DA,
   then following G-SIDs should be updated to the IPv6 DA by the nodes
   along the SRv6 compression sub-path accordingly.

   REQ1-04: The last G-SID in the G-SID Container for the SRv6
   compression sub-path is the a G-SID without COC flavor.

   REQ1-05: When process the G-SID with COC flavor in the IPv6 DA, the
   next G-SID is updated to the IPv6 DA.

8.2. Control Plane

   REQ1-11: ISIS/OSPF/BGP-LS/PCEP extensions for advertising the
   capabilities of supporting G-SRv6 for SRv6 compression.

   REQ1-12: ISIS/OSPF/BGP-LS/BGP extensions for advertising Compressible
   SRv6 SIDs.

   REQ1-13: ISIS/OSPF/BGP-LS/BGP extensions for advertising the
   Continue-of-compression(COC) flavor SID.

   REQ1-21: BGP SR Policy extensions for programming a G-SRv6 path
   combining with Compressible SRv6 SIDs and SRv6 SIDs.

   REQ1-31: PCEP SR Policy extensions for programming a G-SRv6 path
   combining with G-SIDs and SRv6 SIDs.

   REQ1-32: PCEP extensions for programming a G-SRv6 path combining with
   G-SIDs and SRv6 SIDs.

9. IANA Considerations

   This document requests IANA to allocate the following codepoints for
   COC flavor behaviors within the "SRv6 Endpoint Behaviors" sub-
   registry under the top-level "Segment Routing Parameters" registry.

```
+-------+--------+---------------------------+-----------+
| Value | Hex    |     Endpoint behavior     | Reference |
+-------+--------+---------------------------+-----------+
| 101   | 0x0065 |       End with COC        | [This.ID] |
| 102   | 0x0066 |     End with PSP&COC      | [This.ID] |
| 104   | 0x0068 |   End with PSP&USP&COC    | [This.ID] |
| 105   | 0x0069 |       End.X with COC      | [This.ID] |
| 106   | 0x006A |    End.X with PSP&COC     | [This.ID] |
| 108   | 0x006C |  End.X with PSP&USP&COC   | [This.ID] |
| 109   | 0x006D |       End.T with COC      | [This.ID] |
| 110   | 0x006E |    End.T with PSP&COC     | [This.ID] |
| 112   | 0x0070 |  End.T with PSP&USP&COC   | [This.ID] |
| 130   | 0x0082 |    End with PSP&USD&COC   | [This.ID] |
| 131   | 0x0083 |  End with PSP&USP&USD&COC | [This.ID] |
| 133   | 0x0085 |   End.X with PSP&USD&COC  | [This.ID] |
| 135   | 0x0087 | End.X with PSP&USP&USD&COC | [This.ID] |
| 137   | 0x0089 |   End.T with PSP&USD&COC  | [This.ID] |
| 139   | 0x008B | End.T with PSP&USP&USD&COC | [This.ID] |
+-------+--------+---------------------------+-----------+
```
                Table 1: IETF - SRv6 Endpoint Behaviors

10. Security Considerations

   The security considerations described in [RFC8754], and [RFC8402] are
   applicable to this specification. No additional security measure is
   required.

11. Contributors

   TBD

12. Acknowledgements

   TBD

13. References

13.1. Normative References

   [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>.

   [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>.

   [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>.

   [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.filsfilscheng-spring-srv6-srh-comp-sl-enc]
              Cheng, W., Filsfils, C., Li, Z., Cai, D., Voyer, D., Clad,
              F., Zadok, S., Guichard, J., and L. Aihua, "Compressed
              SRv6 Segment List Encoding in SRH", draft-filsfilscheng-
              spring-srv6-srh-comp-sl-enc-01 (work in progress), May
              2020.

13.2. Informative References

   [I-D.filsfils-spring-net-pgm-extension-srv6-usid]
              Filsfils, C., Camarillo, P., Cai, D., Voyer, D., Meilik,
              I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman,
              D., Liu, Y., and J. Guichard, "Network Programming
              extension: SRv6 uSID instruction", draft-filsfils-spring-
              net-pgm-extension-srv6-usid-08 (work in progress),
              November 2020.

   [I-D.cheng-spring-shorter-srv6-sid-requirement]
              Cheng, W., Xie, C., Pang, R., Li, Z., Chen, R., Zu, L.,
              Duan, X., Mirsky, G., Dukes, D., and S. Zadok, "Shorter
              SRv6 SID Requirements", draft-cheng-spring-shorter-srv6-
              sid-requirement-02 (work in progress), July 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.

Authors' Addresses

   Weiqiang Cheng (editor)
   China Mobile
   No.32 Xuanwumen west street
   Beijing 100053
   China

   Email: chengweiqiang@chinamobile.com

   Zhenbin Li
   Huawei Technologies
   Huawei Campus, No. 156 Beiqing Rd.
   Beijing 100095
   China

   Email: lizhenbin@huawei.com

   Cheng Li (editor)
   Huawei Technologies
   Huawei Campus, No. 156 Beiqing Rd.
   Beijing 100095
   China

   Email: c.l@huawei.com

   Francois Clad
   Cisco Systems, Inc
   France

   Email: fclad@cisco.com

   Aihua Liu
   ZTE Corporation
   Shenzhen
   China

   Email: liu.aihua@zte.com.cn

Chongfeng Xie
China Telecom
Technology Innovation park, Changping District
Beijing
China

Email: xiechf@chinatelecom.cn

Yisong Liu
China Mobile
No.32 Xuanwumen west street
Beijing

Email: liuyisong@chinamobile.com

Shay Zadok
Broadcom
Israel

Email: shay.zadok@broadcom.com