SPRING
Internet-Draft
Intended status: Informational
Expires: March 30, 2022

R. Bonica
Juniper
W. Cheng
China Mobile
D. Dukes, Ed.
Cisco Systems
W. Henderickx
Nokia
C. Li
Huawei
P. Shaofu
ZTE
C. Xie
China Telecom

September 26, 2021

# Compressed SRv6 SID List Analysis
## draft-ietf-spring-compression-analysis-00

Abstract

Several mechanisms have been proposed to compress the SRv6 SID list. This document analyzes each mechanism with regard to the requirements stated in the companion requirements document.

Status of This Memo



Copyright Notice

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . 3
2. SRv6 Compression Requirements . . . . . . . . . . . . . . . . 3
  2.1. Encapsulation Header Size . . . . . . . . . . . . . . . . 4
    2.1.1. Reference Scenarios . . . . . . . . . . . . . . . . . 4
  2.2. Forwarding Efficiency . . . . . . . . . . . . . . . . . . 5
    2.2.1. Headers Parsed . . . . . . . . . . . . . . . . . . . 5
    2.2.2. Lookups Performed (LKU) . . . . . . . . . . . . . . . 7
  2.3. State Efficiency . . . . . . . . . . . . . . . . . . . . 8
3. SRv6 Specific Requirements . . . . . . . . . . . . . . . . . 10
  3.1. SRv6 Based . . . . . . . . . . . . . . . . . . . . . . . 10
  3.2. Functional Requirements . . . . . . . . . . . . . . . . . 11
    3.2.1. SRv6 Functionality . . . . . . . . . . . . . . . . . 11
    3.2.2. Heterogeneous SID Lists . . . . . . . . . . . . . . . 14
    3.2.3. SID List Length . . . . . . . . . . . . . . . . . . . 15
    3.2.4. SID Summarization . . . . . . . . . . . . . . . . . . 15
  3.3. Operational Requirements . . . . . . . . . . . . . . . . 15
    3.3.1. Lossless Compression . . . . . . . . . . . . . . . . 16
    3.3.2. Preservation of non-routing information . . . . . . . 16
    3.3.3. Address Planning . . . . . . . . . . . . . . . . . . 16
  3.4. Scalability Requirements . . . . . . . . . . . . . . . . 17
    3.4.1. Compression Levels . . . . . . . . . . . . . . . . . 18
4. Protocol Design Requirements . . . . . . . . . . . . . . . . 18
  4.1. SRv6 Base Coexistence . . . . . . . . . . . . . . . . . . 18
5. Security Requirements . . . . . . . . . . . . . . . . . . . . 19
  5.1. Security Mechanisms . . . . . . . . . . . . . . . . . . . 19
  5.2. SR Domain Protection . . . . . . . . . . . . . . . . . . 19
6. Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . 19
7. Normative References . . . . . . . . . . . . . . . . . . . . 21
Appendix A. Encapsulation analysis . . . . . . . . . . . . . . . 24
  A.1. CRH note . . . . . . . . . . . . . . . . . . . . . . . . 24
  A.2. Analysis results . . . . . . . . . . . . . . . . . . . . 24
Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . . 27

## 1. Introduction

The following mechanisms are proposed to compress the SRv6 SID list:

- o CSID - [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] - Describes two new SRv6 SID flavors, a combination of SID flavors from [I-D.filsfils-spring-net-pgm-extension-srv6-usid] and [I-D.cl-spring-generalized-srv6-for-cmpr]
- o CRH - [I-D.bonica-6man-comp-rtg-hdr] - Requires two new routing header types and a label mapping technique.
- o VSID - [I-D.decraene-spring-srv6-vlsid] - Defines a set of SID behaviors to access smaller SIDs within the SR header.
- o UIDSR - [I-D.mirsky-6man-unified-id-sr] - Extends the SRH to carry MPLS labels or IPv6 addresses.

This document analyzes each mechanism against the requirements stated in [I-D.srcompdt-spring-compression-requirement]. Each section of this document corresponds to a similarly named section in [I-D.srcompdt-spring-compression-requirement]. Each section reiterates corresponding requirements and analyzes each proposal against the those requirements.

The terms compression mechanism, compression solution, and compression proposal are used interchangeably within this document.

## 2. SRv6 Compression Requirements

An SR domain consisting of 3 sub-domains is shown to illustrate the scenarios associated with encapsulation header size, forwarding efficiency and state efficiency.

```
        + * * * * * * * * * * * * * * * * * * * * * * * * * +
        *                                                    *
        * - - - - - - - - + - - - - - - - - + - - - - - - - - *
        *                 |                 |                *
        *   [M1_0]      [B5]  [C_0]       [B7]  [M2_0]       *
[H1]--[E3]                |                 |              [E4]---[H2]
        *        [M1_i] [B6]        [C_j] [B8]       [M2_k]   *
        *                 |                 |                *
        *     Metro 1     |       Core      |     Metro 2    *
        *- - - - - - - - - - - - - - - - - - - - - - - - - -*
        *                                                    *
        *                      SR domain                     *
        + * * * * * * * * * * * * * * * * * * * * * * * * * +
```

Figure 1: Sample SR Domain

- o H1 and H2 are hosts outside the SR domain

- o E3 and E4 are SR domain edge routers
- o Metro 1, Core and Metro 2 are sub-domains with independent IGP instances
- o B5 and B6 are border routers between the Metro 1 and Core
- o B7 and B8 are border routers between the Metro 2 and Core
- o M1_1..M1_i are routers in Metro 1
- o C_1..C_j are routers in Core
- o M2_1..M2_k are routers in Metro 2
- o If Metro and Core are different AS's the border routers (B5 to B8) may be replaced by pairs of ASBRs
- o Flexible algorithms may be deployed within each sub-domain

### 2.1. Encapsulation Header Size

The compression proposal MUST reduce the size of the SRv6 encapsulation header.

Encapsulation header size is evaluated against a set of reference scenarios.

#### 2.1.1. Reference Scenarios

A service provider offers a VPN service with underlay optimization in the SR domain.

- o Hosts H1 and H2 are located in two different sites of a VPN customer.
- o Edge nodes E3 and E4 encapsulate/decapsulate traffic between H1 and H2 to provide the VPN service.
- o The encapsulation consists of a VPN SID (V) (eg END.DT etc) and an SR policy with between 0 and 15 transport segments (T) (eg END or END.X)
- o The SR domain has a block size (B) of 48 bits
- o These independent variables are used to uniquely identify each scenario. For example

    * A scenario with 48bit block size, 3 transport segments and a VPN segment is named 48B.3T.V

Proposals are evaluated against the set of scenarios to calculate the encapsulation in octets (E) and the encapsulation savings (ES) as a fraction of the SRv6 base encapsulation in octets.

E and ES were evaluated for:

- o each proposal in two variants

    * 16-bit SID

  * 32-bit SID
- 48-bit SRv6 block, 0 to 15 transport segments and a VPN segment (expressed in short form as 48B.0-15T.V)

The average encapsulation savings for each proposal is shown below. The complete analysis is recorded in Appendix:

| 16-bit SIDs | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| Average ES | 54.3% | 54.2% | 50.4% | 51.6% | 49.2% |

Table 1: Average ES, 16-bit SIDs, 48B.0-15T.V

| 32-bit SIDs | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| Average ES | 42.5% | 45.5% | 43.2% | 45.5% | 42.5% |

Table 2: Average ES, 32-bit SIDs, 48B.0-15T.V

E and ES are also evaluated for 32bit and 64bit SRv6 block sizes. The CSID 16-bit ES averages 57.4% for 32-bit blocks and 49.9% for 64-bit blocks, other proposals are unchanged.

Conclusion: All proposals meet the requirement to reduce the size of the SRv6 encapsulation header. Variances between proposals are negligible.

### 2.2. Forwarding Efficiency

The compression proposal SHOULD minimize the number of required hardware resources accessed to process a segment.

#### 2.2.1. Headers Parsed

Forwarding efficiency is calculated against the reference scenarios above, recording and summarizing the differences in header parsing for different segment lists.

The following tables indicate the number of headers parsed for each proposal.

| 16-bit | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| PRS(48B.0T).V) | IPv6 | IPv6 | IPv6 | IPv6 | IPv6 |
| PRS(48B.1-4T).V) | IPv6 | IPv6<br>CRH | IPv6<br>CRH | IPv6<br>SRH | IPv6<br>SRH |
| PRS(48B.5-15T).V) | IPv6<br>SRH | IPv6<br>CRH | IPv6<br>CRH | IPv6<br>SRH | IPv6<br>SRH |

Table 3: Headers parsed on non-decapsulating SR segment endpoint nodes, 16-bit SIDs, 48B.0-15T.V

| 16-bit | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| PRS(48B.0T).V) | IPv6 | IPv6 | IPv6 | IPv6 | IPv6 |
| PRS(48B.1-4T).V) | IPv6 | IPv6<br>CRH | IPv6<br>CRH<br>TPF | IPv6<br>SRH | IPv6<br>SRH |
| PRS(48B.5-15T).V) | IPv6<br>SRH | IPv6<br>CRH | IPv6<br>CRH<br>TPF | IPv6<br>SRH | IPv6<br>SRH |

Table 4: Headers parsed on decapsulating SR segment endpoint nodes, 16-bit SIDs, 48B.0-15T.V

| 32-bit | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| PRS(48B.0T.V) | IPv6 | IPv6 | IPv6 | IPv6 | IPv6 |
| PRS(48B.1-15T.V) | IPv6<br>SRH | IPv6<br>CRH | IPv6<br>CRH | IPv6<br>SRH | IPv6<br>SRH |

Table 5: Headers parsed on non-decapsulating SR segment endpoint nodes, 32-bit SIDs, 48B.0-15T.V

| 32-bit | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| PRS(48B.0T.V) | IPv6 | IPv6 | IPv6 | IPv6 | IPv6 |
| PRS(48B.1-15T.V) | IPv6 SRH | IPv6 CRH | IPv6 CRH TPF | IPv6 SRH | IPv6 SRH |

Table 6: Headers parsed on decapsulating SR segment endpoint nodes, 32-bit SIDs, 48B.0-15T.V

Conclusion: Overall, the CSID parses the fewest headers. When per packet state is processed per segment, CSID, VSID and UIDSR proposals may include it in the routing header, CRH may include it in a destination option preceding the CRH.

### 2.2.2. Lookups Performed (LKU)

Some proposals require a different number of lookups per packet, depending on the active segment in a segment list.

An implementation may perform lookups as longest prefix match (LPM) or exact match (EM). CSID, VSID and UIDSR describe SRv6 SID lookup from the IPv6 destination address as an LPM, however an implementation may use either an LPM or EM lookup for SRv6 SIDs. CRH implementations must always uses an exact match for CRH SID lookups.

The following table describes the number of lookups per proposal per segment type.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Adjacency and VPN Segments | LPM (a) | LPM (a)<br>EM (b)<br>EM (b,c) | LPM (a) | LPM (a) |
| Prefix Segments | LPM (a)<br>LPM (d) | LPM (a)<br>EM (b) | LPM (a)<br>LPM (d) | LPM (a)<br>LPM (d) |

Table 7: Lookups

- o [a] On active SID, appearing in the IPv6 Destination address
- o [b] On SID in CRH header

o [c] This lookup is required only when the IPv6 next hop node is not non-CRH aware
o [d] On next SID, appearing in the IPv6 destination address

Note: [I-D.filsfils-spring-net-pgm-extension-srv6-usid] Section 5 describes an optional local implementation to reduce CSID 16-bit lookups, in some cases, by adding local forwarding state. The analysis of this implementation option is not included in this version of the document.

Conclusion: CSID, VSID, and UIDSR require a single lookup to process an adjacency or VPN segment. CRH always requires 2 lookups for VPN segments, and 2 and sometimes 3 lookups for adjacency segments. All proposals require two lookups to process a prefix segment and the next segment.

## 2.3. State Efficiency

The compression proposal SHOULD minimize the amount of additional forwarding state stored at a node.

State efficiency is analyzed in a sub-domain of the SR domain, with the following parameters:

o N: the number of SRv6 nodes in the sub-domain
o I: the number of IGP algorithms [I-D.ietf-lsr-flex-algo] configured
o A: the number of local adjacency SIDs at a node
o D: the number of attached SR sub-domains at a border node
o V: the number of VPN services at edge nodes

For a sub-domain consisting of:

o 1000 SRv6 nodes (N=1000) with some number of non-SRv6 nodes
o 2 IGP algorithms (I=2)
o 100 adjacencies per SRv6 node (A=100)
o up to 10 attached sub-domains per border node (D=10)
o 1000 VPN service segments per edge (V=1000)

The number of forwarding entries at a node is calculated for any node, a border node, and an edge node.

UIDSR, CSID and VSID require the following entries:

o a FIB entry for the node's prefix segment (1), per algorithm (I=2).
o a FIB entry per local adjacency SID (A=100) **Note1
o At border nodes (or any SRv6 nodes) either:

   * A.1) a FIB entry per domain (D=10) to swap the IPv6 destination address prefix.
   * A.2) no additional FIB entries, and the SR source places a 128-bit SID in the segment list of a packet if needed.
- At edge nodes, a FIB entry per VPN segment (V=1000)

CRH requires:

- a CFIB entry per CRH node per IGP algorithm for local and remote prefix segments (N*I=2000)
- a CFIB entry per local adjacency segment (A=100) **Note1

   * When non-CRH adjacent nodes are present, additional state is required for CRH as per [I-D.bonica-6man-comp-rtg-hdr] Appendix B (note, only the second option in the appendix is considered feasible due to state explosion)

      + B.1) Up to one CFIB entry per next endpoint and an additional CFIB entry per adjacency to support non-CRH adjacent endpoints, assuming IP flex algo is not implemented on non-CRH nodes (I=1) ((N+A)*I=1200).
- At border nodes, assuming two inter-domain links per adjacent domain for redundancy, additional state is required as per [I-D.bonica-6man-comp-rtg-hdr] Appendix B (note, only the second option in the appendix is considered feasible due to state explosion):

   * C.1) In a common CRH network topology, the remote sub-domain borders support CRH: a CFIB entry per CRH node per IGP algorithm for local and remote prefix segments (N*I) plus a CFIB entry per local adjacency segment (A) plus a CFIB entry per connected remote border router (20) (N*I+A+20=2120).
   * C.2) In a poorly designed CRH network topology, the remote sub-domain borders do not support CRH: a CFIB entry per unique endpoint (N*D*I), plus a CFIB entry per local adjacency segment (A), assuming IP flex algo is not implemented on non-CRH border domain (I=1), plus inter-domain adjacency (20) (N*D*I+2=10120).
- At edge nodes, V=1000 entries for SRv6 based VPN SIDs and another V=1000 entries for CFIB and TPF VPN SIDs.

**Note1: there may be additional adjacency SIDs for protected, unprotected, and per algorithm adjacencies, resulting in some multiple of A. This is common for all compression proposals.

| 16-bit and 32-bit | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| S(N1000,I2,A100,D10) | 102 | 2100 | 102 | 102 |
| | A.1:112 | | A.1:112 | A.1:112 |
| | A.2:102 | | A.2:102 | A.2:102 |
| | | B.1:3300 | | |
| | | C.1:2120 | | |
| | | C.2:10120 | | |
| | | | | |
| S(V1000) | 1000 | 2000 | 1000 | 1000 |

Table 8: Forwarding State Maintained

Conclusion: CSID, VSID and UIDSR minimize forwarding state stored at a node. CRH moves per segment state from the packet to the FIB.

# 3. SRv6 Specific Requirements

## 3.1. SRv6 Based

A solution to compress SRv6 SID Lists SHOULD be based on the SRv6 architecture, control plane and data plane. The compression solution MAY be based on a different data plane and control plane, provided that it derives sufficient benefit.

This section records the use of SRv6 standards for compression.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| U.RFC8402 | Yes | Yes - update required for SRv6 data plane | Yes | Yes |
| U.RFC8754 | Yes | No | Yes - update required for segments left | Yes - update for flags and segments left |
| U.PGM | Yes | No | Yes - update required for SID behaviors | Yes |
| U.IGP | Yes | No | Yes | Yes - additional extensions |
| U.BGP | Yes | No | Yes | Yes |
| U.POL | Yes | No | Yes | Yes |
| U.BLS | Yes | No | Yes | Yes - additional extensions |
| U.SVC | Yes | No | Yes | Yes |
| U.ALG | Yes | Yes - Adds IP flex Algo | Yes | Yes |
| U.OAM | Yes | No | Yes | Yes |

Table 9: SRv6 Based

Conclusion: CSID is SRv6 based, requiring no updates to existing SRv6 standards, VSID and UIDSR require updates. CRH is not strictly based on SRv6 but is able to provide equivalent functionality.

## 3.2. Functional Requirements

### 3.2.1. SRv6 Functionality

A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement ensures no SRv6 functionality is lost. It is particularly important to understand how a proposal, as evaluated in section "SRv6 Based", provides this functionality.

Functional requirements and the drafts defining how a proposal provides the functionality are documented in the table below.

| Draft reference Abbreviations |
|---|
| RFC8986: [RFC8986] |
| SRV6POL: [I-D.ietf-spring-segment-routing-policy] |
| SRV6EXT: [I-D.ietf-lsr-isis-srv6-extensions] |
| SRV6BGPSVC: [I-D.ietf-bess-srv6-services] |
| SRV6BGPLS: [I-D.ietf-idr-bgpls-srv6-ext] |
| SRV6SVCP: [I-D.ietf-spring-sr-service-programming] |
| SRV6OAM: [I-D.ietf-6man-spring-srv6-oam] |
| SRV6FLEXALG: [I-D.ietf-lsr-flex-algo] |
| SRV6TILFA: [I-D.ietf-rtgwg-segment-routing-ti-lfa] |
| RFC8402: [RFC8402] |
| RFC8754: [RFC8754] |
| CRH: [I-D.bonica-6man-comp-rtg-hdr] |
| VSID: [I-D.decraene-spring-srv6-vlsid] |
| UIDSR: [I-D.mirsky-6man-unified-id-sr] |
| IPFLEXALG: [I-D.ietf-lsr-ip-flexalgo] |
| CRHEXT: [I-D.bonica-lsr-crh-isis-extensions] |
| SRM6BGPSVC: [I-D.ssangli-bess-bgp-vpn-srm6] |
| CSID: [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] |

Abbreviations

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| F.SID | RFC8402 | CRH | RFC8402 | RFC8402 1 |
| F.Scope | RFC8402 | CRH | RFC8402 | RFC8402 1 |
| F.PFX | RFC8402, RFC8986, CSID adds an END SID flavor | CRH | RFC8402, RFC8986, VSID updates the End behavior | RFC8402, RFC8986 with new flavor 1 |
| F.ADJ | RFC8402, RFC8986, CSID adds an END.X flavor | CRH | RFC8402, RFC8986, VSID updates the End.X behavior | RFC8402, RFC8986 with new flavor 1 |
| F.BIND | RFC8402, RFC8986 | CRH | RFC8402, RFC8986, VSID updates the End.B | RFC8402, RFC8986 with new flavor 1 |

| | | | | |
|---|---|---|---|---|
| | | | behaviors | |
| F.PEER | RFC8402, RFC8986, CSID adds an END.X. flavor | CRH | RFC8402, RFC8986, VSID updates the End.X behaviors | RFC8402, RFC8986 with new flavor 1,2 |
| F.SVC | RFC8986 | CRH | RFC8986, VSID updates the service segment behaviors | RFC8986 1 |
| F.ALG | SRV6FLEXALG | IPFLEXALG | SRV6FLEXALG | SRV6FLEXALG |
| F.TILFA | SRV6TILFA | SRV6TILFA | SRV6TILFA | SRV6TILFA 3 |
| F.SEC | RFC8754 | CRH | RFC8754 | RFC8754 |
| F.IGP | SRV6EXT | CRH-EXT | SRV6EXT | SRV6EXT 1,4 |
| F.BGP | SRV6BGPSVC | SRM6BGPSVC | SRV6BGPSVC | SRV6BGPSVC 1 |
| F.POL | SRV6SRPOL | SRV6SRPOL update required | SRV6SRPOL | SRV6SRPOL |
| F.BLS | SRV6BGPLS | (specification required) | SRV6BGPLS and addition for VSID Length | SRV6BGPLS 5 |
| F.SFC | SRV6SVCP | CRH | SRV6SVC | SRV6SVCP 1 |
| F.PING | SRV6OAM | CRH | SRV6OAM | SRv6OAM |

Table 10: SRv6 Functionality

1. UIDSR with Global Container SID + local index enhancement
2. draft-peng-spring-truncates-sid-inter-domain
3. For protections described in section 6.1.2.1, 6.1.2.2, and 6.2, to get next-next SID from SRH with the help of draft-pl-spring-compr-path-recover.
4. Need more extensions to advertise the capability of U-SID compression (32bits, 16bits, etc.). Note: Global Container SID + local index enhancement.
5. IGP extensions

Conclusion: CSID supports SRv6 functionality. CRH VSID and UID support SRv6 functionality or equivalent with some new specifications.

### 3.2.2. Heterogeneous SID Lists

The compression proposal SHOULD support a combination of compressed and non-compressed segments in a single path. As an example, a solution may satisfy this requirement without being SRv6 based by using a binding SID to impose an additional SRv6 header (IPv6 header plus optional SRH) with non-compressed SID.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Heterogeneous SID Lists | Yes | Yes | Yes | Yes |

Heterogeneous SID Lists

VSID require a binding SID with an additional SRv6 encapsulation to encode non-compressed segments in a single path. VSID changes the interpretation of the SRH Segments Left field, which makes it capable of carrying only compressed segments.

The CRH can include a binding SID that imposes a new IPv6 header with an SRH. This is required when the next segment endpoint in the path can process the SRH, but not the CRH. The next segment endpoint or a subsequent endpoint can execute decapsulation, removing the new IPv6 header and exposing the old one with its CRH. This is required because an IPv6 packet can carry only one routing header.

CSID and UIDSR permit the encoding of, and processing of, any combination of compressed or non-compressed segments in a segment list of an SRH.

CSID makes use of the SRH, without modification, to encode CSIDs as 128 bits, supporting the use of non-compressed segments within the SRH.

UIDSR modifies the interpretation of the SRH Segments Left field at segment endpoint nodes to allow variable segment lengths within a segment list.

Conclusion: All proposals support heterogeneous SID lists. CSID and UIDSR support heterogeneous SID lists in the SRH, while CRH and VSID require installation of binding SIDs at midpoint nodes.

### 3.2.3. SID List Length

The compression proposal MUST be able to represent SR paths that contain up to 16 segments.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| 16 Segments | Yes | Yes | Yes | Yes |

SID List Length

Conclusion: All proposals support segment lists of at least 16 segments.

### 3.2.4. SID Summarization

The solution MUST be compatible with segment summarization.

In inter sub-domain deployments with summarization:

- o Any node can reach any other node in another sub-domain via a prefix segment.
- o Prefixes are summarized for advertisement between domains.

Without summarization, border router SIDs must be leaked:

- o An additional global prefix segment is required for each domain border to be traversed.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| SID Summarization | Yes | No | Yes | Yes |

SID Summarization

Conclusion: CSID, VSID and UIDSR support segment summarization, CRH does not.

## 3.3. Operational Requirements

### 3.3.1. Lossless Compression

A path traversed using a compressed SID list MUST always be the same as the path traversed using the uncompressed SID list if no compression was applied.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Lossless Compression | Yes | Yes | Yes | Yes |

Lossless Compression

Conclusion: All proposals provide lossless compression.

### 3.3.2. Preservation of non-routing information

The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Preserves Non-Routing Information | Complies | Complies | Complies | Complies |

Preservation of non-routing information

Conclusion: All proposals preserve non-routing information.

### 3.3.3. Address Planning

Description: Network operators require addressing plan flexibility, The compression mechanism MUST support flexible IPv6 address planning, it MUST support deployment by using GUA from different address blocks.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Flexible Address Planning | Yes | Yes | Yes | Yes |

Address Planning

All compression mechanisms provide the encapsulation savings described in Tables 1 and 2. CRH provides these encapsulation savings regardless of the IPv6 addressing scheme. CSID adds a CSID container, or one compressed SID (END.X with XPS behavior), for each change in locator block in a segment list. VSID (via XPS behavior) and UIDSR add one compressed SID for each change in locator block in the segment list.

The XPS behavior draws the new address block from the control plane. At the time of publication, this control plane behavior is undefined. Therefore XPS impact on the control plane is not entirely understood. While it may be possible to define these mechanisms without impacting the control plane, specifications are not yet available.

Conclusion: All proposals support flexible IPv6 planning.

### 3.4. Scalability Requirements

The compression proposal MUST be capable of representing 65000 adjacency segments per node.

The compression proposal MUST be capable of representing 1 million prefix segments per SID numbering space.

The compression proposal MUST be capable of representing 1 million services per node.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Adjacency Segment Scale 65000 | Yes | Yes | Yes | Yes |
| Prefix Segment Scale 1000000 | Yes | Yes | Yes | Yes |
| Service Scale 1000000 | Yes | Yes | Yes | Yes |

Table 11: Scale Requirements

The 32-bit variants of all proposals support this scale of prefix, adjacency and services at a node.

Each proposals 16-bit variant supports a lesser scale. All proposals can encode 2^16 prefix, adjacency and service segments. However, each proposal has various ways of supporting some larger scale per node if required.

CRH 16-bit proposes the encoding of the ultimate segment in a TPF destination option instead of the CRH. This supports 2^32 service segments per node.

VSID proposes the combination of multiple vSIDs, by copying multiple SIDs to a destination address or looking up the next segment in the segment list. This supports more than 2^16 adjacency and service segments per node.

CSID 16-bit variant uses a LIB for adjacency and service segments, the LIB allows local definition of SIDs longer than 16-bits when needed. This supports more than 2^16 adjacency and service segments per node.

UIDSR defines a segment type that modifies the value of SRH segments left field to support variable segment sizes within the segment list. This supports 2^32 adjacency and service segments per node.

Conclusion: All proposals meet scalability requirements.

### 3.4.1. Compression Levels

The compression proposal SHOULD be able to support multiple levels of compression.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Multiple compression Levels | Yes | Yes | Yes | Yes |

Compression Levels

Conclusion: All proposals support 16-bit and 32-bit SID variants.

## 4. Protocol Design Requirements

### 4.1. SRv6 Base Coexistence

The compression proposal MUST support deployment in SRv6 networks.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| SRv6 Base Coexistence | Yes | Yes | Yes | Yes |

SRv6 Base Coexistence

Conclusion: All proposals can be deployed simultaneously with the SRv6 base solution.

## 5. Security Requirements

### 5.1. Security Mechanisms

The compression solution SHOULD be able to address security issues that it introduces, using existing security mechanisms.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| Security Mechanisms | Yes | Yes | Yes | Yes |

Security Mechanisms

Conclusion: All proposals address security issues they may introduce with existing security mechanisms.

### 5.2. SR Domain Protection

A compression solution must not require nodes outside the SR domain to know SID values within the SR domain, and it must provide the ability to block nodes outside an SR domain from accessing SIDS.

| | CSID | CRH | VSID | UIDSR |
|---|---|---|---|---|
| SR Domain Protection | Yes | Yes | Yes | Yes |

SR Domain Protection

Conclusion: All proposals protect SIDs within the SR domain.

## 6. Conclusions

Encapsulation Header Size

o All proposals meet the requirement to reduce the size of the SRv6 encapsulation header. Variances between proposals are negligible.

Forwarding Efficiency

o Overall, the CSID parses the fewest headers. When per packet state is processed per segment, CSID, VSID and UIDSR proposals may include it in the routing header, CRH may include it in a destination option preceding the CRH.

- CSID, VSID, and UIDSR require a single lookup to process an adjacency or VPN segment. CRH always requires 2 lookups for VPN segments, and 2 and sometimes 3 lookups for adjacency segments. All proposals require two lookups to process a prefix segment and the next segment.

State Efficiency

- CSID, VSID and UIDSR minimize forwarding state stored at a node. CRH moves per segment state from the packet to the FIB.

SRv6 Based

- CSID is SRv6 based, requiring no updates to existing SRv6 standards, VSID and UIDSR require updates. CRH is not strictly based on SRv6 but is able to provide equivalent functionality.

SRv6 Functionality

- CSID supports SRv6 functionality. CRH VSID and UID support SRv6 functionality or equivalent with some new specifications.

Heterogeneous SID lists

- All proposals support heterogeneous SID lists. CSID and UIDSR support heterogeneous SID lists in the SRH, while CRH and VSID require installation of binding SIDs at midpoint nodes.

SID List Length

- All proposals support segment lists of at least 16 segments.

SID Summarization

- VSID, CSID and UIDSR support segment summarization, CRH does not.

Operational Requirements

- All proposals provide lossless compression.
- All proposals preserve non-routing information.
- All proposals support flexible IPv6 planning.

Scalability Requirements

- All proposals meet scalability requirements.
- All proposals support 16-bit and 32-bit SID variants.

Protocol Design Requirements

- All proposals can be deployed simultaneously with the SRv6 base solution.

Security Requirements

- All proposals address security issues they may introduce with existing security mechanisms.
- All proposals protect SIDs within the SR domain.

## 7. Normative References

[I-D.bonica-6man-comp-rtg-hdr]
Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-26 (work in progress), May 2021.

[I-D.bonica-6man-vpn-dest-opt]
Bonica, R., Kamite, Y., Jalil, L., Zhou, Y., and G. Chen, "The IPv6 Tunnel Payload Forwarding (TPF) Option", draft-bonica-6man-vpn-dest-opt-16 (work in progress), July 2021.

[I-D.bonica-lsr-crh-isis-extensions]
Kaneriya, P., Shetty, R., Hegde, S., and R. Bonica, "IS-IS Extensions To Support The IPv6 Compressed Routing Header (CRH)", draft-bonica-lsr-crh-isis-extensions-05 (work in progress), August 2021.

[I-D.cl-spring-generalized-srv6-for-cmpr]
Cheng, W., Li, Z., Li, C., Clad, F., Liu, A., Xie, C., Liu, Y., and S. Zadok, "Generalized SRv6 Network Programming for SRv6 Compression", draft-cl-spring-generalized-srv6-for-cmpr-03 (work in progress), April 2021.

[I-D.decraene-spring-srv6-vlsid]
Decraene, B., Raszuk, R., Li, Z., and C. Li, "SRv6 vSID: Network Programming extension for variable length SIDs", draft-decraene-spring-srv6-vlsid-06 (work in progress), September 2021.

[I-D.filsfils-spring-net-pgm-extension-srv6-usid]
Filsfils, C., Garvia, P. C., 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-11 (work in progress), September 2021.

[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. N., and L. Aihua, "Compressed SRv6 Segment List Encoding in SRH", draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-03 (work in progress), May 2021.

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-11 (work in progress), June 2021.

[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-07 (work in progress), April 2021.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-08 (work in progress), June 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-17 (work in progress), July 2021.

[I-D.ietf-lsr-ip-flexalgo]
Britto, W., Hegde, S., Kaneriya, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", draft-ietf-lsr-ip-flexalgo-03 (work in progress), May 2021.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-17 (work in progress), June 2021.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-07 (work in progress), June 2021.

[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-13 (work in progress), May 2021.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-05 (work in progress), September 2021.

[I-D.mirsky-6man-unified-id-sr]
Weiqiang, C., Mirsky, G., Shaofu, P., Aihua, L., and G. S. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-10 (work in progress), September 2021.

[I-D.srcompdt-spring-compression-requirement]
Cheng, W., Xie, C., Bonica, R., Dukes, D., Li, C., Shaofu, P., and W. Henderickx, "Compressed SRv6 SID List Requirements", draft-srcompdt-spring-compression-requirement-07 (work in progress), July 2021.

[I-D.ssangli-bess-bgp-vpn-srm6]
Sangli, S. and R. Bonica, "BGP based Virtual Private Network (VPN) Services over SRm6 enabled IPv6 networks", draft-ssangli-bess-bgp-vpn-srm6-02 (work in progress), September 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>.

[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>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

## Appendix A. Encapsulation analysis

### A.1. CRH note

CRH compression efficiency statistics are derived as follows:

If an SR path contains no transport segments and a VPN segment, the SR path is encoded in a single IPv6 header (40 bytes). The destination address in the IPv6 header is a classic SRv6 SID (e.g., END.DT4, END.DT6).

If the SR path contains T transport segments and a VPN segment, and T is greater than 0, the SR path can be encoded:

- o With an IPv6 Tunnel Payload Function (TPF) Option [I-D.bonica-6man-vpn-dest-opt]
- o Without a TPF Option

If the SR path is encoded with a TPF Option, the packet includes a single IPv6 Header (40 bytes), a CRH (variable length), and a Destination Options header (8 bytes). The destination address in the IPv6 header represents the IPv6 address of an interface on the first transport segment endpoint. The CRH must be large enough to contain the subsequent T segments.

If the SR path is encoded without a TPF Option, the packet includes a single IPv6 Header (40 bytes) plus a CRH (variable length). The destination address in the IPv6 header represents the IPv6 address of an interface on the first transport segment endpoin . The CRH must be large enough to contain T+1 segments. In the CRH, SID[1] maps to the IPv6 address of the PE router. SID[0] maps to a classic SRv6 SID (e.g., END.DT4) that is instantiated on the PE router.

In some deployment scenarios, each encoding strategy yields better compression.

### A.2. Analysis results

The detailed encapsulation and encapsulation savings per proposal with one VPN segment and "T" transport segments:

| T | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| 0 | 40 | 40 | 40 | 40 | 40 |
| 1 | 40 | 48 | 56 | 56 | 64 |
| 2 | 40 | 56 | 56 | 56 | 64 |
| 3 | 40 | 56 | 64 | 56 | 64 |
| 4 | 64 | 56 | 64 | 64 | 64 |
| 5 | 64 | 56 | 64 | 64 | 64 |
| 6 | 64 | 64 | 64 | 64 | 64 |
| 7 | 64 | 64 | 72 | 64 | 64 |
| 8 | 64 | 64 | 72 | 72 | 64 |
| 9 | 80 | 64 | 72 | 72 | 80 |
| 10 | 80 | 72 | 72 | 72 | 80 |
| 11 | 80 | 72 | 80 | 72 | 80 |
| 12 | 80 | 72 | 80 | 80 | 80 |
| 13 | 80 | 72 | 80 | 80 | 80 |
| 14 | 96 | 80 | 80 | 80 | 80 |
| 15 | 96 | 80 | 88 | 80 | 80 |

Table 12: Encapsulation (E) octets, 16bit SIDS, 48B.0-15T.V

| T | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| 0 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 1 | 37.5% | 25.0% | 12.5% | 12.5% | 0.0% |
| 2 | 50.0% | 30.0% | 30.0% | 30.0% | 20.0% |
| 3 | 58.3% | 41.7% | 33.3% | 41.7% | 33.3% |
| 4 | 42.9% | 50.0% | 42.9% | 42.9% | 42.9% |
| 5 | 50.0% | 56.3% | 50.0% | 50.0% | 50.0% |
| 6 | 55.6% | 55.6% | 55.6% | 55.6% | 55.6% |
| 7 | 60.0% | 60.0% | 55.0% | 60.0% | 60.0% |
| 8 | 63.6% | 63.6% | 59.1% | 59.1% | 63.6% |
| 9 | 58.3% | 66.7% | 62.5% | 62.5% | 58.3% |
| 10 | 61.5% | 65.4% | 65.4% | 65.4% | 61.5% |
| 11 | 64.3% | 67.9% | 64.3% | 67.9% | 64.3% |
| 12 | 66.7% | 70.0% | 66.7% | 66.7% | 66.7% |
| 13 | 68.8% | 71.9% | 68.8% | 68.8% | 68.8% |
| 14 | 64.7% | 70.6% | 70.6% | 70.6% | 70.6% |
| 15 | 66.7% | 72.2% | 69.4% | 72.2% | 72.2% |

Table 13: Encapsulation Savings (ES), 16bit SIDS, 48B.0-15T.V

| T | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| 0 | 40 | 40 | 40 | 40 | 40 |
| 1 | 64 | 56 | 56 | 56 | 64 |
| 2 | 64 | 56 | 64 | 56 | 64 |
| 3 | 64 | 64 | 64 | 64 | 64 |
| 4 | 64 | 64 | 72 | 64 | 64 |
| 5 | 80 | 72 | 72 | 72 | 80 |
| 6 | 80 | 72 | 80 | 72 | 80 |
| 7 | 80 | 80 | 80 | 80 | 80 |
| 8 | 80 | 80 | 88 | 80 | 80 |
| 9 | 96 | 88 | 88 | 88 | 96 |
| 10 | 96 | 88 | 96 | 88 | 96 |
| 11 | 96 | 96 | 96 | 96 | 96 |
| 12 | 96 | 96 | 104 | 96 | 96 |
| 13 | 112 | 104 | 104 | 104 | 112 |
| 14 | 112 | 104 | 112 | 104 | 112 |
| 15 | 112 | 112 | 112 | 112 | 112 |

Table 14: Encapsulation (E) octets, 32bit SIDS, 48B.0-15T.V

| T | CSID | CRH | CRH+TPF | VSID | UIDSR |
|---|---|---|---|---|---|
| 0 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 1 | 0.0% | 12.5% | 12.5% | 12.5% | 0.0% |
| 2 | 20.0% | 30.0% | 20.0% | 30.0% | 20.0% |
| 3 | 33.3% | 33.3% | 33.3% | 33.3% | 33.3% |
| 4 | 42.9% | 42.9% | 35.7% | 42.9% | 42.9% |
| 5 | 37.5% | 43.8% | 43.8% | 43.8% | 37.5% |
| 6 | 44.4% | 50.0% | 44.4% | 50.0% | 44.4% |
| 7 | 50.0% | 50.0% | 50.0% | 50.0% | 50.0% |
| 8 | 54.5% | 54.5% | 50.0% | 54.5% | 54.5% |
| 9 | 50.0% | 54.2% | 54.2% | 54.2% | 50.0% |
| 10 | 53.8% | 57.7% | 53.8% | 57.7% | 53.8% |
| 11 | 57.1% | 57.1% | 57.1% | 57.1% | 57.1% |
| 12 | 60.0% | 60.0% | 56.7% | 60.0% | 60.0% |
| 13 | 56.3% | 59.4% | 59.4% | 59.4% | 56.3% |
| 14 | 58.8% | 61.8% | 58.8% | 61.8% | 58.8% |
| 15 | 61.1% | 61.1% | 61.1% | 61.1% | 61.1% |

Table 15: Encapsulation Savings (ES), 32bit SIDS, 48B.0-15T.V

Authors' Addresses

Ron Bonica
Juniper

Email: rbonica@juniper.net

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Darren Dukes (editor)
Cisco Systems

Email: ddukes@cisco.com

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn