SPRING
Internet-Draft
Intended status: Standards Track
Expires: May 29, 2021

C. Filsfils, Ed.
P. Camarillo, Ed.
Cisco Systems, Inc.
J. Leddy
Individual Contributor
D. Voyer
Bell Canada
S. Matsushima
SoftBank
Z. Li
Huawei Technologies
November 25, 2020

SRv6 Network Programming
draft-ietf-spring-srv6-network-programming-25

Abstract

The SRv6 Network Programming framework enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header.

Each instruction is implemented on one or several nodes in the network and identified by an SRv6 Segment Identifier in the packet.

This document defines the SRv6 Network Programming concept and specifies the base set of SRv6 behaviors that enables the creation of interoperable overlays with underlay optimization.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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 29, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 4
2. Terminology 4
2.1. Requirements Language 5
3. SRv6 SID 5
3.1. SID Format 6
3.2. SID Allocation within an SR domain 7
3.3. SID Reachability 9
4. SR Endpoint Behaviors 10
4.1. End: Endpoint 12
4.1.1. Upper-Layer Header 12
4.2. End.X: Layer-3 Cross-Connect 13
4.3. End.T: Specific IPv6 Table Lookup 14
4.4. End.DX6: Decapsulation and IPv6 Cross-Connect 14
4.5. End.DX4: Decapsulation and IPv4 Cross-Connect 15
4.6. End.DT6: Decapsulation and Specific IPv6 Table Lookup . . 16
4.7. End.DT4: Decapsulation and Specific IPv4 Table Lookup . . 17
4.8. End.DT46: Decapsulation and Specific IP Table Lookup . . 18
4.9. End.DX2: Decapsulation and L2 Cross-Connect 19
4.10. End.DX2V: Decapsulation and VLAN L2 Table Lookup 20
4.11. End.DT2U: Decapsulation and Unicast MAC L2 Table Lookup . 21
4.12. End.DT2M: Decapsulation and L2 Table Flooding 22
4.13. End.B6.Encaps: Endpoint Bound to an SRv6 Policy w/ Encaps 22
4.14. End.B6.Encaps.Red: End.B6.Encaps with Reduced SRH 24
4.15. End.BM: Endpoint Bound to an SR-MPLS Policy 24
4.16. Flavors 25
4.16.1. PSP: Penultimate Segment Pop of the SRH 25
4.16.2. USP: Ultimate Segment Pop of the SRH 28
4.16.3. USD: Ultimate Segment Decapsulation 28
5. SR Policy Headend Behaviors 29
5.1. H.Encaps: SR Headend with Encapsulation in an SRv6 Policy 30
5.2. H.Encaps.Red: H.Encaps with Reduced Encapsulation 30

5.3. H.Encaps.L2: H.Encaps Applied to Received L2 Frames . . . 31
5.4. H.Encaps.L2.Red: H.Encaps.Red Applied to Received L2 frames 31
6. Counters 31
7. Flow-based Hash Computation 32
8. Control Plane 32
8.1. IGP 32
8.2. BGP-LS 33
8.3. BGP IP/VPN/EVPN 33
8.4. Summary 33
9. Security Considerations 35
10. IANA Considerations 35
10.1. Ethernet Next Header Type 35
10.2. SRv6 Endpoint Behaviors Registry 36
10.2.1. Initial Registrations 36
11. Acknowledgements 37
12. Contributors 38
13. References 41
13.1. Normative References 41
13.2. Informative References 41
Authors' Addresses 43

1. Introduction

Segment Routing [RFC8402] leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions, called segments. Each one of these instructions represents a function to be called at a specific location in the network. A function is locally defined on the node where it is executed and may range from simply moving forward in the Segment List to any complex user-defined behavior. Network programming combines segment routing functions, both simple and complex, to achieve a networking objective that goes beyond mere packet routing.

This document defines the SRv6 Network Programming concept and specifies the main segment routing behaviors to enable the creation of interoperable overlays with underlay optimization.

The companion document [I-D.filsfils-spring-srv6-net-pgm-illustration] illustrates the concepts defined in this document.

Familiarity with the Segment Routing Header [RFC8754] is expected.

2. Terminology

The following terms used within this document are defined in [RFC8402]: Segment Routing, SR Domain, Segment ID (SID), SRv6, SRv6 SID, SR Policy, Prefix-SID, and Adj-SID.

The following terms used within this document are defined in [RFC8754]: SRH, SR Source Node, Transit Node, SR Segment Endpoint Node, Reduced SRH, Segments Left and Last Entry.

SL: The Segments Left field of the SRH

FIB: Forwarding Information Base. A FIB lookup is a lookup in the forwarding table.

SA: Source Address

DA: Destination Address

SRv6 SID function: The function part of the SID is an opaque identification of a local behavior bound to the SID. It is formally defined in Section 3.1 of this document.

SRv6 Segment Endpoint behavior: A packet processing behavior executed at an SRv6 Segment Endpoint Node. Section 4 of this document defines SRv6 Segment Endpoint behaviors related to traffic-engineering and

overlay use-cases. Other behaviors (e.g. service programming) are outside the scope of this document.

An SR Policy is resolved to a SID list. A SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SR path.

(SA,DA) (S3, S2, S1; SL) represents an IPv6 packet with:

- Source Address is SA, Destination Address is DA, and next-header is SRH.

- SRH with SID list <S1, S2, S3> with Segments Left = SL.

- Note the difference between the <> and () symbols: <S1, S2, S3> represents a SID list where S1 is the first SID and S3 is the last SID to traverse. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID. When referring to an SR policy in a high-level use-case, it is simpler to use the <S1, S2, S3> notation. When referring to an illustration of the detailed packet behavior, the (S3, S2, S1; SL) notation is more convenient.

- The payload of the packet is omitted.

Per-VRF VPN label: a single label for the entire VRF that is shared by all routes from that VRF ([RFC4364] Section 4.3.2)

Per-CE VPN label: a single label for each attachment circuit that is shared by all routes with the same "outgoing attachment circuit" ([RFC4364] Section 4.3.2)

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. SRv6 SID

RFC8402 defines an SRv6 Segment Identifier as an IPv6 address explicitly associated with the segment.

When an SRv6 SID is in the Destination Address field of an IPv6 header of a packet, it is routed through Transit Nodes in an IPv6 network as an IPv6 address.

Its processing is defined in [RFC8754] section 4.3 and reproduced here as a reminder.

> Without constraining the details of an implementation, the SR segment endpoint node creates Forwarding Information Base (FIB) entries for its local SIDs.
>
> When an SRv6-capable node receives an IPv6 packet, it performs a longest-prefix-match lookup on the packet's destination address. This lookup can return any of the following:
>
> * A FIB entry that represents a locally instantiated SRv6 SID
>
> * A FIB entry that represents a local interface, not locally instantiated as an SRv6 SID
>
> * A FIB entry that represents a nonlocal route
>
> * No Match

Section 4 of this document defines a new set of SRv6 SID behaviors in addition to that defined in [RFC8754] Section 4.3.1.

3.1. SID Format

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

A locator may be represented as B:N where B is the SRv6 SID block (IPv6 prefix allocated for SRv6 SIDs by the operator) and N is the identifier of the parent node instantiating the SID.

When the LOC part of the SRv6 SIDs is routable, it leads to the node which instantiates the SID.

The FUNCT is an opaque identification of a local behavior bound to the SID.

The term "function" refers to the bit-string in the SRv6 SID. The term "behavior" identifies the behavior bound to the SID. The behaviors are defined in Section 4 of this document.

An SRv6 Segment Endpoint Behavior may require additional information for its processing (e.g. related to the flow or service). This information may be encoded in the ARG bits of the SID.

In such a case, the semantics and format of the ARG bits are defined as part of the SRv6 endpoint behavior specification.

The ARG value of a routed SID SHOULD remain constant among packets in a given flow. Varying ARG values among packets in a flow may result in different ECMP hashing and cause re-ordering.

3.2. SID Allocation within an SR domain

Locators are assigned consistent with IPv6 infrastructure allocation. For example, a network operator may:

- Assign block B::/48 to the SR domain
- Assign a unique B:N::/64 block to each SRv6-enabled node in the domain

As an example, one mobile service provider has commercially deployed SRv6 across more than 1000 commercial routers and 1800 whitebox routers. All these devices are enabled for SRv6 and advertise SRv6 SIDs. The provider historically deployed IPv6 and assigned infrastructure addresses from ULA space [RFC4193]. They specifically allocated three /48 prefixes (Country X, Country Y, Country Z) to support their SRv6 infrastructure. From those /48 prefixes each router was assigned a /64 prefix from which all SIDs of that router are allocated.

In another example, a large mobile and fixed-line service provider has commercially deployed SRv6 in their country-wide network. This provider is assigned a /20 prefix by an RIR (Regional Internet Registry). They sub-allocated a few /48 prefixes to their infrastructure to deploy SRv6. Each router is assigned a /64 prefix from which all SIDs of that router are allocated.

IPv6 address consumption in both these examples is minimal, representing one billionth and one millionth of the available address space, respectively.

A service provider receiving the current minimum allocation of a /32 from an RIR may assign a /48 prefix to their infrastructure deploying

SRv6, and subsequently allocate /64 prefixes for SIDs at each SRv6 node. The /48 assignment is one sixty-five thousandth (1/2^16) of the usable IPv6 address space available for assignment by the provider.

When an operator instantiates a SID at a node, they specify a SID value B:N:FUNCT and the behavior bound to the SID using one of the SRv6 Endpoint Behavior codepoint of the registry defined in this document (see Table 4).

The node advertises the SID, B:N:FUNCT, in the control-plane (see Section 8) together with the SRv6 Endpoint Behavior codepoint identifying the behavior of the SID.

An SR Source Node cannot infer the behavior by examination of the FUNCT value of a SID.

Therefore, the SRv6 Endpoint Behavior codepoint is advertised along with the SID in the control plane.

An SR Source Node uses the SRv6 Endpoint Behavior codepoint to map the received SID (B:N:FUNCT) to a behavior.

An SR Source Node selects a desired behavior at an advertising node by selecting the SID (B:N:FUNCT) advertised with the desired behavior.

As an example, a network operator may:

- Assign an SRv6 SID block 2001:db8:bbbb::/48 from their in-house operation block for their SRv6 infrastructure

- Assign an SRv6 Locator 2001:db8:bbbb:3::/64 to one particular router, for example Router 3, in their SR Domain

- At Router 3, within the locator 2001:db8:bbbb:3::/64, the network operator or the router performs dynamic assignment for:

  * Function 0x0100 associated with the behavior End.X (Endpoint with cross-connect) between router 3 and its connected neighbor router, for example Router 4. This function is encoded as 16-bit value and has no arguments (F=16, A=0).
    This SID is advertised in the control plane as 2001:db8:bbbb:3:100:: with SRv6 Endpoint Behavior codepoint value of 5.

  * Function 0x0101 associated with the behavior End.X (Endpoint with cross-connect) between router 3 and its connected neighbor

router, for example Router 2. This function is encoded as 16-bit value and has no arguments (F=16, A=0). This SID is advertised in the control plane as 2001:db8:bbbb:3:101:: with SRv6 Endpoint Behavior codepoint value of 5.

These examples do not preclude any other IPv6 addressing allocation scheme.

3.3. SID Reachability

Most often, the node N would advertise IPv6 prefix(es) matching the LOC parts covering its SIDs or shorter-mask prefix. The distribution of these advertisements and calculation of their reachability are specific to the routing protocol and are outside of the scope of this document.

An SRv6 SID is said to be routed if its SID belongs to an IPv6 prefix advertised via a routing protocol. An SRv6 SID that does not fulfill this condition is non-routed.

Let's provide a classic illustration:

Node N is configured explicitly with two SIDs: 2001:db8:b:1:100:: and 2001:db8:b:2:101::.

The network learns about a path to 2001:db8:b:1::/64 via the IGP and hence a packet destined to 2001:db8:b:1:100:: would be routed up to N. The network does not learn about a path to 2001:db8:b:2::/64 via the IGP and hence a packet destined to 2001:db8:b:2:101:: would not be routed up to N.

A packet could be steered to a non-routed SID 2001:db8:b:2:101:: by using a SID list <...,2001:db8:b:1:100::,2001:db8:b:2:101::,...> where the non-routed SID is preceded by a routed SID to the same node. Routed and non-routed SRv6 SIDs are the SRv6 instantiation of global and local segments, respectively [RFC8402].

4. SR Endpoint Behaviors

Following is a set of well-known behaviors that can be associated with a SID.

End	Endpoint function The SRv6 instantiation of a Prefix SID [RFC8402]
End.X	Endpoint with Layer-3 cross-connect The SRv6 instantiation of an Adj SID [RFC8402]
End.T	Endpoint with specific IPv6 table lookup
End.DX6	Endpoint with decapsulation and IPv6 cross-connect e.g. IPv6-L3VPN (equivalent to per-CE VPN label)
End.DX4	Endpoint with decaps and IPv4 cross-connect e.g. IPv4-L3VPN (equivalent to per-CE VPN label)
End.DT6	Endpoint with decapsulation and IPv6 table lookup e.g. IPv6-L3VPN (equivalent to per-VRF VPN label)
End.DT4	Endpoint with decapsulation and IPv4 table lookup e.g. IPv4-L3VPN (equivalent to per-VRF VPN label)
End.DT46	Endpoint with decapsulation and IP table lookup e.g. IP-L3VPN (equivalent to per-VRF VPN label)
End.DX2	Endpoint with decapsulation and L2 cross-connect e.g. L2VPN use-case
End.DX2V	Endpoint with decaps and VLAN L2 table lookup e.g. EVPN Flexible cross-connect use-case
End.DT2U	Endpoint with decaps and unicast MAC L2table lookup e.g. EVPN Bridging unicast use-case
End.DT2M	Endpoint with decapsulation and L2 table flooding e.g. EVPN Bridging BUM use-case with ESI filtering
End.B6.Encaps	Endpoint bound to an SRv6 policy with encapsulation SRv6 instantiation of a Binding SID
End.B6.Encaps.RED	End.B6.Encaps with reduced SRH SRv6 instantiation of a Binding SID
End.BM	Endpoint bound to an SR-MPLS Policy SRv6 instantiation of an SR-MPLS Binding SID

The list is not exhaustive. In practice, any function can be attached to a local SID: e.g. a node N can bind a SID to a local VM or container which can apply any complex processing on the packet, provided there is a behavior codepoint allocated for the processing.

When an SRv6-capable node (N) receives an IPv6 packet whose destination address matches a FIB entry that represents a locally instantiated SRv6 SID (S), the IPv6 header chain is processed as defined in Section 4 of [RFC8200]. For SRv6 SIDs associated with an Endpoint Behavior defined in this document, the SRH and Upper-layer Header are processed as defined in the following subsections.

The pseudocode describing these behaviors details local processing at a node. An implementation of the pseudocode is compliant as long as the externally observable wire protocol is as described by the pseudocode.

Section 4.16 defines flavors of some of these behaviors.

Section 10.2 of this document defines the IANA Registry used to maintain all these behaviors as well as future ones defined in other documents.

4.1. End: Endpoint

The Endpoint behavior ("End" for short) is the most basic behavior. It is the instantiation of a Prefix-SID [RFC8402].

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

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
             header in the packet, whose type is identified by
             the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address,
             Code 0 (Hop limit exceeded in transit),
             interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.

S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Submit the packet to the egress IPv6 FIB lookup and
          transmission to the new destination
S16. }
```

Notes:
The End behavior operates on the same FIB table (i.e. identified by VRF or L3 relay id) associated to the packet. Hence the FIB lookup on line S15 is done in the same FIB table as the ingress interface.

4.1.1. Upper-Layer Header

When processing the Upper-layer Header of a packet matching a FIB entry locally instantiated as an SRv6 End SID do the following:

```
S01. If (Upper-Layer Header type is allowed by local configuration) {
S02.   Proceed to process the Upper-layer Header
S03. } Else {
S04.   Send an ICMP Parameter Problem to the Source Address,
          Code 4 (SR Upper-layer Header Error),
          Pointer set to the offset of the Upper-layer Header,
          Interrupt packet processing and discard the packet.
S05  }
```

Allowing processing of specific Upper-Layer Headers types is useful for OAM. As an example, an operator might permit pinging of SIDs. To do this they may enable local configuration to allow Upper-layer Header type 58 (ICMPv6).

It is RECOMMENDED that an implementation of local configuration only allows Upper-layer Header processing of types that do not result in the packet being forwarded (e.g. ICMPv6).

4.2. End.X: Layer-3 Cross-Connect

The "Endpoint with cross-connect to an array of layer-3 adjacencies" behavior (End.X for short) is a variant of the End behavior.

It is the SRv6 instantiation of an Adjacency-SID [RFC8402] and its main use is for traffic-engineering policies.

Any SID instance of this behavior is associated with a set, J, of one or more Layer-3 adjacencies.

When N receives a packet destined to S and S is a local End.X SID, the line S15 from the End processing is replaced by the following:

```
S15.   Submit the packet to the IPv6 module for transmission
          to the new destination via a member of J
```

Notes:
S15. If the set J contains several L3 adjacencies, then one element of the set is selected based on a hash of the packet's header (see Section 7).

If a node N has 30 outgoing interfaces to 30 neighbors, usually the operator would explicitly instantiate 30 End.X SIDs at N: one per layer-3 adjacency to a neighbor. Potentially, more End.X could be explicitly defined (groups of layer-3 adjacencies to the same neighbor or to different neighbors).

Note that if N has an outgoing interface bundle I to a neighbor Q made of 10 member links, N may allocate up to 11 End.X local SIDs: one for the bundle itself and then up to one for each Layer-2 member link. The flows steered using the End.X SID corresponding to the bundle itself get load balanced across the member links via hashing while the flows steered using the End.X SID corresponding to a member link get steered over that specific member link alone.

When the End.X behavior is associated with a BGP Next-Hop, it is the SRv6 instantiation of the BGP Peering Segments [RFC8402].

When processing the Upper-layer Header of a packet matching a FIB entry locally instantiated as an SRv6 End.X SID, process the packet as per Section 4.1.1.

4.3. End.T: Specific IPv6 Table Lookup

The "Endpoint with specific IPv6 table lookup" behavior (End.T for short) is a variant of the End behavior.

The End.T behavior is used for multi-table operation in the core. For this reason, an instance of the End.T behavior is associated with an IPv6 FIB table T.

When N receives a packet destined to S and S is a local End.T SID, the line S15 from the End processing is replaced by the following:

```
S15.1.   Set the packet's associated FIB table to T
S15.2.   Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
```

When processing the Upper-layer Header of a packet matching a FIB entry locally instantiated as an SRv6 End.T SID, process the packet as per Section 4.1.1.

4.4. End.DX6: Decapsulation and IPv6 Cross-Connect

The "Endpoint with decapsulation and cross-connect to an array of IPv6 adjacencies" behavior (End.DX6 for short) is a variant of the End.X behavior.

One of the applications of the End.DX6 behavior is the L3VPNv6 use-case where a FIB lookup in a specific tenant table at the egress

Provider Edge (PE) is not required. This is equivalent to the per-CE VPN label in MPLS [RFC4364].

The End.DX6 SID MUST be the last segment in a SR Policy, and it is associated with one or more L3 IPv6 adjacencies J.

When N receives a packet destined to S and S is a local End.DX6 SID, N does the following processing:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DX6 SID, the following is done:

```
S01. If (Upper-Layer Header type == 41(IPv6) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Forward the exposed IPv6 packet to the L3 adjacency J
S04. } Else {
S05.    Process as per Section 4.1.1
S06. }
```

Notes:
S01. 41 refers to IPv6 encapsulation as defined by IANA allocation for Internet Protocol Numbers.
S03. If the End.DX6 SID is bound to an array of L3 adjacencies, then one entry of the array is selected based on the hash of the packet's header (see Section 7).

4.5. End.DX4: Decapsulation and IPv4 Cross-Connect

The "Endpoint with decapsulation and cross-connect to an array of IPv4 adjacencies" behavior (End.DX4 for short) is a variant of the End.X behavior.

One of the applications of the End.DX4 behavior is the L3VPNv4 use-case where a FIB lookup in a specific tenant table at the egress PE

is not required. This is equivalent to the per-CE VPN label in MPLS [RFC4364].

The End.DX4 SID MUST be the last segment in a SR Policy, and it is associated with one or more L3 IPv4 adjacencies J.

When N receives a packet destined to S and S is a local End.DX4 SID, N does the following processing:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DX4 SID, the following is done:

```
S01. If (Upper-Layer Header type == 4(IPv4) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Forward the exposed IPv4 packet to the L3 adjacency J
S04. } Else {
S05.    Process as per Section 4.1.1
S06. }
```

Notes:
S01. 4 refers to IPv4 encapsulation as defined by IANA allocation for Internet Protocol Numbers
S03. If the End.DX4 SID is bound to an array of L3 adjacencies, then one entry of the array is selected based on the hash of the packet's header (see Section 7).

4.6. End.DT6: Decapsulation and Specific IPv6 Table Lookup

The "Endpoint with decapsulation and specific IPv6 table lookup" behavior (End.DT6 for short) is a variant of the End.T behavior.

One of the applications of the End.DT6 behavior is the L3VPNv6 use-case where a FIB lookup in a specific tenant table at the egress PE is required. This is equivalent to the per-VRF VPN label in MPLS [RFC4364].

Note that an End.DT6 may be defined for the main IPv6 table in which case an End.DT6 supports the equivalent of an IPv6inIPv6 decapsulation (without VPN/tenant implication).

The End.DT6 SID MUST be the last segment in a SR Policy, and a SID instance is associated with an IPv6 FIB table T.

When N receives a packet destined to S and S is a local End.DT6 SID, N does the following processing:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DT6 SID, N does the following:

```
S01. If (Upper-Layer Header type == 41(IPv6) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Set the packet's associated FIB table to T
S04.    Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S05. } Else {
S06.    Process as per Section 4.1.1
S07. }
```

4.7. End.DT4: Decapsulation and Specific IPv4 Table Lookup

The "Endpoint with decapsulation and specific IPv4 table lookup" behavior (End.DT4 for short) is a variant of the End.T behavior.

One of the applications of the End.DT4 behavior is the L3VPNv4 use-case where a FIB lookup in a specific tenant table at the egress PE is required. This is equivalent to the per-VRF VPN label in MPLS [RFC4364].

Note that an End.DT4 may be defined for the main IPv4 table in which case an End.DT4 supports the equivalent of an IPv4inIPv6 decapsulation (without VPN/tenant implication).

The End.DT4 SID MUST be the last segment in a SR Policy, and a SID instance is associated with an IPv4 FIB table T.

When N receives a packet destined to S and S is a local End.DT4 SID, N does the following processing:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DT4 SID, N does the following:

```
S01. If (Upper-Layer Header type == 4(IPv4) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Set the packet's associated FIB table to T
S04.    Submit the packet to the egress IPv4 FIB lookup and
           transmission to the new destination
S05. } Else {
S06.    Process as per Section 4.1.1
S07. }
```

4.8. End.DT46: Decapsulation and Specific IP Table Lookup

The "Endpoint with decapsulation and specific IP table lookup" behavior (End.DT46 for short) is a variant of the End.DT4 and End.DT6 behavior.

One of the applications of the End.DT46 behavior is the L3VPN use-case where a FIB lookup in a specific IP tenant table at the egress PE is required. This is equivalent to single per-VRF VPN label (for IPv4 and IPv6) in MPLS[RFC4364].

Note that an End.DT46 may be defined for the main IP table in which case an End.DT46 supports the equivalent of an IPinIPv6 decapsulation(without VPN/tenant implication).

The End.DT46 SID MUST be the last segment in a SR Policy, and a SID instance is associated with an IPv4 FIB table T4 and an IPv6 FIB table T6.

When N receives a packet destined to S and S is a local End.DT46 SID, N does the following processing:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DT46 SID, N does the following:

```
S01. If (Upper-layer Header type == 4(IPv4) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Set the packet's associated FIB table to T4
S04.    Submit the packet to the egress IPv4 FIB lookup and
           transmission to the new destination
S05. } Else if (Upper-layer Header type == 41(IPv6) ) {
S06.    Remove the outer IPv6 Header with all its extension headers
S07.    Set the packet's associated FIB table to T6
S08.    Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S09. } Else {
S10.    Process as per Section 4.1.1
S11. }
```

4.9. End.DX2: Decapsulation and L2 Cross-Connect

The "Endpoint with decapsulation and Layer-2 cross-connect to an outgoing L2 interface (OIF)" (End.DX2 for short) is a variant of the endpoint behavior.

One of the applications of the End.DX2 behavior is the L2VPN [RFC4664] / EVPN VPWS [RFC7432] [RFC8214] use-case.

The End.DX2 SID MUST be the last segment in a SR Policy, and it is associated with one outgoing interface I.

When N receives a packet destined to S and S is a local End.DX2 SID, N does:

```
S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S04.   }
S05.   Proceed to process the next header in the packet
S06. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.DX2 SID, the following is done:

```
S01. If (Upper-Layer Header type == 143(Ethernet) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Forward the Ethernet frame to the OIF I
S04. } Else {
S05.    Process as per Section 4.1.1
S06. }
```

Notes:
S01. IANA has allocated the Internet Protocol number 143 to Ethernet [IEEE.802.3_2018] (see Section 10.1).
S03. An End.DX2 behavior could be customized to expect a specific IEEE header (e.g. VLAN tag) and rewrite the egress IEEE header before forwarding on the outgoing interface.

Note that an End.DX2 SID may also be associated with a bundle of outgoing interfaces.

4.10. End.DX2V: Decapsulation and VLAN L2 Table Lookup

The "Endpoint with decapsulation and specific VLAN table lookup" behavior (End.DX2V for short) is a variant of the End.DX2 behavior.

One of the applications of the End.DX2V behavior is the EVPN Flexible cross-connect use-case. The End.DX2V behavior is used to perform a lookup of the Ethernet frame VLANs in a particular L2 table. Any SID instance of this behavior is associated with an L2 Table T.

When N receives a packet whose IPv6 DA is S and S is a local End.DX2 SID, the processing is identical to the End.DX2 behavior except for the Upper-layer header processing which is modified as follows:

```
S03. Lookup the exposed VLANs in L2 table T, and forward
        via the matched table entry.
```

Notes:
S03. An End.DX2V behavior could be customized to expect a specific VLAN format and rewrite the egress VLAN header before forwarding on the outgoing interface.

4.11. End.DT2U: Decapsulation and Unicast MAC L2 Table Lookup

The "Endpoint with decapsulation and specific unicast MAC L2 table lookup" behavior (End.DT2U for short) is a variant of the End behavior.

One of the applications of the End.DT2U behavior is the EVPN Bridging unicast [RFC7432]. Any SID instance of the End.DT2U behavior is associated with an L2 Table T.

When N receives a packet whose IPv6 DA is S and S is a local End.DT2U SID, the processing is identical to the End.DX2 behavior except for the Upper-layer header processing which is as follows:

```
S01. If (Upper-Layer Header type == 143(Ethernet) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Learn the exposed MAC Source Address in L2 Table T
S04.    Lookup the exposed MAC Destination Address in L2 Table T
S05.    If (matched entry in T) {
S06.       Forward via the matched table T entry
S07.    } Else {
S08.       Forward via all L2 OIFs entries in table T
S09.    }
S10. } Else {
S11.    Process as per Section 4.1.1
S12. }
```

Notes:
S01. IANA has allocated the Internet Protocol number 143 to Ethernet (see Section 10.1).
S03. In EVPN [RFC7432], the learning of the exposed MAC Source Address is done via control plane. In L2VPN VPLS [RFC4761] [RFC4762] reachability is obtained by standard learning bridge functions in the data plane.

4.12. End.DT2M: Decapsulation and L2 Table Flooding

The "Endpoint with decapsulation and specific L2 table flooding" behavior (End.DT2M for short) is a variant of the End.DT2U behavior.

Two of the applications of the End.DT2M behavior are the EVPN Bridging of broadcast, unknown and multicast (BUM) traffic with Ethernet Segment Identifier (ESI) filtering [RFC7432] and the EVPN ETREE [RFC8317]use-cases.

Any SID instance of this behavior is associated with a L2 table T. The behavior also takes an argument: "Arg.FE2". This argument provides a local mapping to ESI for split-horizon filtering of the received traffic to exclude specific OIF (or set of OIFs) from L2 table T flooding. The allocation of the argument values is local to the SR Endpoint Node instantiating this behavior and the signaling of the argument to other nodes for the EVPN functionality via control plane.

When N receives a packet whose IPv6 DA is S and S is a local End.DT2M SID, the processing is identical to the End.DX2 behavior except for the Upper-layer header processing which is as follows:

```
S01. If (Upper-Layer Header type == 143(Ethernet) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Learn the exposed MAC Source Address in L2 Table T
S04.    Forward via all L2OIFs excluding those associated by the
           identifier Arg.FE2
S05. } Else {
S06.    Process as per Section 4.1.1
S07. }
```

Notes:
S01. IANA has allocated the Internet Protocol number 143 to Ethernet (see Section 10.1).
S03. In EVPN [RFC7432], the learning of the exposed MAC Source Address is done via control plane. In L2VPN VPLS [RFC4761] [RFC4762] reachability is obtained by standard learning bridge functions in the data plane.

4.13. End.B6.Encaps: Endpoint Bound to an SRv6 Policy w/ Encaps

This is a variation of the End behavior.

One of its applications is to express scalable traffic-engineering policies across multiple domains. It is one of the SRv6 instantiations of a Binding SID [RFC8402].

Any SID instance of this behavior is associated with an SR Policy B and a source address A.

When N receives a packet whose IPv6 DA is S and S is a local End.B6.Encaps SID, does:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
             header in the packet, whose type is identified by
             the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.       Send an ICMP Time Exceeded message to the Source Address,
             Code 0 (Hop limit exceeded in transit),
             interrupt packet processing and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > (Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.
S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Push a new IPv6 header with its own SRH containing B
S16.   Set the outer IPv6 SA to A
S17.   Set the outer IPv6 DA to the first SID of B
S18.   Set the outer Payload Length, Traffic Class, Flow Label,
          Hop Limit and Next-Header fields
S19.   Submit the packet to the egress IPv6 FIB lookup and
          transmission to the new destination
S20. }
```

Notes:
S14. The SRH MAY be omitted when the SRv6 Policy B only contains one SID and there is no need to use any flag, tag or TLV.
S17. The Payload Length, Traffic Class, Hop Limit and Next-Header fields are set as per [RFC2473]. The Flow Label is computed as per [RFC6437].

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.B6.Encaps SID, process the packet as per Section 4.1.1.

4.14. End.B6.Encaps.Red: End.B6.Encaps with Reduced SRH

This is an optimization of the End.B6.Encaps behavior.

End.B6.Encaps.Red reduces the size of the SRH by one SID by excluding the first SID in the SRH of the new IPv6 header. Thus, the first segment is only placed in the IPv6 Destination Address of the new IPv6 header and the packet is forwarded according to it.

The SRH Last Entry field is set as defined in Section 4.1.1 of [RFC8754].

The SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag or TLV.

4.15. End.BM: Endpoint Bound to an SR-MPLS Policy

The "Endpoint bound to an SR-MPLS Policy" is a variant of the End behavior.

The End.BM behavior is required to express scalable traffic-engineering policies across multiple domains where some domains support the MPLS instantiation of Segment Routing. This is an SRv6 instantiation of an SR-MPLS Binding SID [RFC8402].

Any SID instance of this behavior is associated with an SR-MPLS Policy B.

When N receives a packet whose IPv6 DA is S and S is a local End.BM SID, does:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
             header in the packet, whose type is identified by
             the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address,
             Code 0 (Hop limit exceeded in transit),
             interrupt packet processing and discard the packet.

S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > (Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address,
             Code 0 (Erroneous header field encountered),
             Pointer set to the Segments Left field,
             interrupt packet processing and discard the packet.

S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Push the MPLS label stack for B
S16.   Submit the packet to the MPLS engine for transmission to the
          topmost label.
S17. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.BM SID, process the packet as per Section 4.1.1.

4.16. Flavors

The Penultimate Segment Pop of the SRH (PSP), Ultimate Segment Pop of the SRH (USP) and Ultimate Segment Decapsulation (USD) flavors are variants of the End, End.X and End.T behaviors. The End, End.X and End.T behaviors can support these flavors either individually or in combinations.

4.16.1. PSP: Penultimate Segment Pop of the SRH

4.16.1.1. Guidelines

SR Segment Endpoint Nodes advertise the SIDs instantiated on them via control plane protocols as described in Section 8. Different

behavior ids are allocated for flavored and unflavored SIDs (see Table 4).

An SR Segment Endpoint Node that offers both PSP and non-PSP flavored behavior advertises them as two different SIDs.

The SR Segment Endpoint Node only advertises the PSP flavor if the operator enables this capability at the node.

The PSP operation is deterministically controlled by the SR Source Node.

A PSP-flavored SID is used by the Source SR Node when it needs to instruct the penultimate SR Segment Endpoint Node listed in the SRH to remove the SRH from the IPv6 header.

4.16.1.2. Definition

SR Segment Endpoint Nodes receive the IPv6 packet with the Destination Address field of the IPv6 Header equal to its SID address.

A penultimate SR Segment Endpoint Node is one that, as part of the SID processing, copies the last SID from the SRH into the IPv6 Destination Address and decrements the Segments Left value from one to zero.

The PSP operation only takes place at a penultimate SR Segment Endpoint Node and does not happen at any Transit Node. When a SID of PSP-flavor is processed at a non-penultimate SR Segment Endpoint Node, the PSP behavior is not performed as described in the pseudocode below since Segments Left would not be zero.

The SRH processing of the End, End.X and End.T behaviors are modified: after the instruction "S14. Update IPv6 DA with Segment List[Segments Left]" is executed, the following instructions must be executed as well:

```
S14.1.   If (Segments Left == 0) {
S14.2.      Update the Next Header field in the preceding header to the
               Next Header value from the SRH
S14.3.      Decrease the IPv6 header Payload Length by 8*(Hdr Ext Len+1)
S14.4.      Remove the SRH from the IPv6 extension header chain
S14.5.   }
```

The usage of PSP does not increase the MTU of the IPv6 packet and hence does not have any impact on the PMTU discovery mechanism.

As a reminder, [RFC8754] defines in section 5 the SR Deployment Model within the SR Domain [RFC8402]. Within this framework, the Authentication Header (AH) is not used to secure the SRH as described in Section 7.5 of [RFC8754].

In the context of this specification, the End, End.X and End.T behaviors with PSP do not contravene Section 4 of [RFC8200] because the destination address of the incoming packet is the address of the node executing the behavior.

4.16.1.3. Use-case

One use-case for the PSP functionality is streamlining the operation of an egress border router.

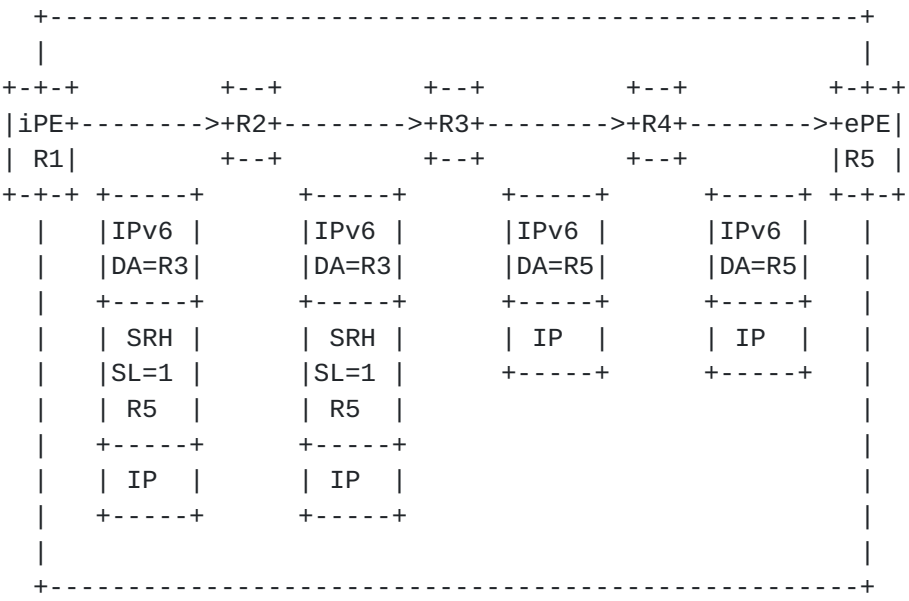


Figure 1: PSP use-case topology

In the above illustration, for a packet sent from iPE to ePE, node R3 is an intermediate traffic engineering waypoint and is the penultimate segment endpoint router; the node that copies the last segment from the SRH into the IPv6 Destination Address and decrements segments left to 0. The SDN controller knows that no other node after R3 needs to inspect the SRH, and it instructs R3 to remove the exhausted SRH from the packet by using a PSP-flavored SID.

The benefits for the egress PE are straightforward:

-as part of the decapsulation process the egress PE is required to parse and remove fewer bytes from the packet.

-if a lookup on an upper-layer IP header is required (e.g. per-VRF VPN), the header is more likely to be within the memory accessible to the lookup engine in the forwarding ASIC (Application-specific integrated circuit).

4.16.2. USP: Ultimate Segment Pop of the SRH

The SRH processing of the End, End.X and End.T behaviors are modified: the instructions S02-S04 are substituted by the following ones:

```
S02.     If (Segments Left == 0) {
S03.1.      Update the Next Header field in the preceding header to the
               Next Header value of the SRH
S03.2.      Decrease the IPv6 header Payload Length by 8*(Hdr Ext Len+1)
S03.3.      Remove the SRH from the IPv6 extension header chain
S03.4.      Proceed to process the next header in the packet
S04.     }
```

One of the applications of the USP flavor is when a packet with an SRH is destined to an application on hosts with smartNICs implementing SRv6. The USP flavor is used to remove the consumed SRH from the extension header chain before sending the packet to the host.

4.16.3. USD: Ultimate Segment Decapsulation

The Upper-layer header processing of the End, End.X and End.T behaviors are modified as follows:

```
End:
S01. If (Upper-layer Header type == 41(IPv6) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S04. } Else if (Upper-layer Header type == 4(IPv4) ) {
S05.    Remove the outer IPv6 Header with all its extension headers
S06.    Submit the packet to the egress IPv4 FIB lookup and
           transmission to the new destination
S07. Else {
S08.    Process as per Section 4.1.1
S09. }
```

```
End.T:
S01. If (Upper-layer Header type == 41(IPv6) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Set the packet's associated FIB table to T
S04.    Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S05. } Else if (Upper-layer Header type == 4(IPv4) ) {
S06.    Remove the outer IPv6 Header with all its extension headers
S07.    Set the packet's associated FIB table to T
S08.    Submit the packet to the egress IPv4 FIB lookup and
           transmission to the new destination
S09. Else {
S10.    Process as per Section 4.1.1
S11. }

End.X:
S01. If (Upper-layer Header type == 41(IPv6) ||
          Upper-layer Header type == 4(IPv4) ) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Forward the exposed IP packet to the L3 adjacency J
S04. } Else {
S05.    Process as per Section 4.1.1
S06. }
```

One of the applications of the USD flavor is the case of TI-LFA in P routers with encapsulation. The USD flavor allows the last Segment Endpoint Node in the repair path list to decapsulate the IPv6 header added at the TI-LFA Point of Local Repair and forward the inner packet.

5. SR Policy Headend Behaviors

This section describes a set of SR Policy Headend behaviors.

H.Encaps	SR Headend Behavior with Encapsulation in an SR Policy
H.Encaps.Red	H.Encaps with Reduced Encapsulation
H.Encaps.L2	H.Encaps Applied to Received L2 Frames
H.Encaps.L2.Red	H.Encaps.Red Applied to Received L2 Frames

This list is not exhaustive and future documents may define additional behaviors.

5.1. H.Encaps: SR Headend with Encapsulation in an SRv6 Policy

Node N receives two packets P1=(A, B2) and P2=(A,B2)(B3, B2, B1; SL=1). B2 is neither a local address nor SID of N.

Node N is configured with an IPv6 Address T (e.g. assigned to its loopback).

N steers the transit packets P1 and P2 into an SR Policy with a Source Address T and a Segment list <S1, S2, S3>.

The H.Encaps encapsulation behavior is defined as follows:

```
S01.   Push an IPv6 header with its own SRH
S02.   Set outer IPv6 SA = T and outer IPv6 DA = S1
S03.   Set outer Payload Length, Traffic Class, Hop Limit and
          Flow Label fields
S04.   Set the outer Next-Header value
S05.   Decrement inner IPv6 Hop Limit or IPv4 TTL
S06.   Submit the packet to the IPv6 module for transmission to S1
```

Note:
S03: As described in [RFC6437] (IPv6 Flow Label Specification).

After the H.Encaps behavior, P1' and P2' respectively look like:

- (T, S1) (S3, S2, S1; SL=2) (A, B2)

- (T, S1) (S3, S2, S1; SL=2) (A, B2) (B3, B2, B1; SL=1)

The received packet is encapsulated unmodified (with the exception of the IPv4 TTL or IPv6 Hop Limit that is decremented as described in [RFC2473]).

The H.Encaps behavior is valid for any kind of Layer-3 traffic. This behavior is commonly used for L3VPN with IPv4 and IPv6 deployments. It may be also used for TI-LFA [I-D.ietf-rtgwg-segment-routing-ti-lfa] at the point of local repair.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag or TLV.

5.2. H.Encaps.Red: H.Encaps with Reduced Encapsulation

The H.Encaps.Red behavior is an optimization of the H.Encaps behavior.

H.Encaps.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only placed in the Destination Address field of the pushed IPv6 header.

After the H.Encaps.Red behavior, P1' and P2' respectively look like:

- (T, S1) (S3, S2; SL=2) (A, B2)

- (T, S1) (S3, S2; SL=2) (A, B2) (B3, B2, B1; SL=1)

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag or TLV.

5.3. H.Encaps.L2: H.Encaps Applied to Received L2 Frames

The H.Encaps.L2 behavior encapsulates a received Ethernet [IEEE.802.3_2018] frame and its attached VLAN header, if present, in an IPv6 packet with an SRH. The Ethernet frame becomes the payload of the new IPv6 packet.

The Next Header field of the SRH MUST be set to 143.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag or TLV.

The encapsulating node MUST remove the preamble (if any) and frame check sequence (FCS) from the Ethernet frame upon encapsulation and the decapsulating node MUST regenerate, as required, the preamble and FCS before forwarding Ethernet frame.

5.4. H.Encaps.L2.Red: H.Encaps.Red Applied to Received L2 frames

The H.Encaps.L2.Red behavior is an optimization of the H.Encaps.L2 behavior.

H.Encaps.L2.Red reduces the length of the SRH by excluding the first SID in the SRH of the pushed IPv6 header. The first SID is only places in the Destination Address field of the pushed IPv6 header.

The push of the SRH MAY be omitted when the SRv6 Policy only contains one segment and there is no need to use any flag, tag or TLV.

6. Counters

A node supporting this document SHOULD implement a pair of traffic counters (one for packets and one for bytes) per local SID entry, for traffic that matched that SID and was processed successfully (i.e. packets which generate ICMP Error Messages or are dropped are not

counted). The retrieval of these counters from MIB, NETCONF/YANG or any other data structure is outside the scope of this document.

7. Flow-based Hash Computation

When a flow-based selection within a set needs to be performed, the IPv6 Source Address, the IPv6 Destination Address and the IPv6 Flow Label of the outer IPv6 header MUST be included in the flow-based hash.

This occurs when a FIB lookup is performed and multiple ECMP paths exist to the updated destination address.

This occurs when End.X, End.DX4, or End.DX6 are bound to an array of adjacencies.

This occurs when the packet is steered in an SR policy whose selected path has multiple SID lists.

Additionally, any transit router in an SRv6 domain includes the outer flow label in its ECMP flow-based hash [RFC6437].

8. Control Plane

In an SDN environment, one expects the controller to explicitly provision the SIDs and/or discover them as part of a service discovery function. Applications residing on top of the controller could then discover the required SIDs and combine them to form a distributed network program.

The concept of "SRv6 network programming" refers to the capability for an application to encode any complex program as a set of individual functions distributed through the network. Some functions relate to underlay SLA, others to overlay/tenant, others to complex applications residing in VM and containers.

While not necessary for an SDN control plane, the remainder of this section provides a high-level illustrative overview of how control-plane protocols may be involved with SRv6. Their specification is outside the scope of this document.

8.1. IGP

The End, End.T and End.X SIDs express topological behaviors and hence are expected to be signaled in the IGP together with the flavors PSP, USP and USD. The IGP should also advertise the maximum SRv6 SID depth (MSD) capability of the node for each type of SRv6 operation - in particular, the SR source (e.g. H.Encaps), intermediate endpoint

(e.g. End, End.X) and final endpoint (e.g. End.DX4, End.DT6) behaviors. These capabilities are factored in by an SR Source Node (or a controller) during the SR Policy computation.

The presence of SIDs in the IGP does not imply any routing semantics to the addresses represented by these SIDs. The routing reachability to an IPv6 address is solely governed by the non-SID-related IGP prefix reachability information that includes locators. Routing is neither governed nor influenced in any way by a SID advertisement in the IGP.

These SIDs provide important topological behaviors for the IGP to build FRR solutions based on TI-LFA [I-D.ietf-rtgwg-segment-routing-ti-lfa] and for TE processes relying on IGP topology database to build SR policies.

8.2. BGP-LS

BGP-LS provides the functionality for topology discovery that includes the SRv6 capabilities of the nodes, their locators and locally instantiated SIDs. This enables controllers or applications to build an inter-domain topology that can be used for computation of SR Policies using the SRv6 SIDs.

8.3. BGP IP/VPN/EVPN

The End.DX4, End.DX6, End.DT4, End.DT6, End.DT46, End.DX2, End.DX2V, End.DT2U and End.DT2M SIDs can be signaled in BGP.

In some scenarios an egress PE advertising a VPN route might wish to abstract the specific behavior bound to the SID from the ingress PE and other routers in the network. In such case, the SID may be advertised using the Opaque SRv6 Endpoint Behavior codepoint defined in Table 4. The details of such control plane signaling mechanisms are out of the scope of this document.

8.4. Summary

The following table summarizes behaviors for SIDs that can be signaled in which each respective control plane protocol.

	IGP	BGP-LS	BGP IP/VPN/EVPN
End (PSP, USP, USD)	X	X	
End.X (PSP, USP, USD)	X	X	
End.T (PSP, USP, USD)	X	X	
End.DX6	X	X	X
End.DX4	X	X	X
End.DT6	X	X	X
End.DT4	X	X	X
End.DT46	X	X	X
End.DX2		X	X
End.DX2V		X	X
End.DT2U		X	X
End.DT2M		X	X
End.B6.Encaps		X	
End.B6.Encaps.Red		X	
End.B6.BM		X	

Table 1: SRv6 locally instantiated SIDs signaling

The following table summarizes which SR Policy Headend capabilities are signaled in which signaling protocol.

	IGP	BGP-LS	BGP IP/VPN/EVPN
H.Encaps	X	X	
H.Encaps.Red	X	X	
H.Encaps.L2		X	
H.Encaps.L2.Red		X	

Table 2: SRv6 Policy Headend behaviors signaling

The previous table describes generic capabilities. It does not describe specific instantiated SR policies.

For example, a BGP-LS advertisement of H.Encaps behavior would describe the capability of node N to perform a H.Encaps behavior. Specifically, it would describe how many SIDs could be pushed by N without significant performance degradation.

As a reminder, an SR policy is always assigned a Binding SID [RFC8402]. BSIDs are also advertised in BGP-LS as shown in Table 1.

Hence, the Table 2 only focuses on the generic capabilities related to H.Encaps.

9. Security Considerations

The security considerations for Segment Routing are discussed in [RFC8402]. Section 5 of [RFC8754] describes the SR Deployment Model and the requirements for securing the SR Domain. The security considerations of [RFC8754] also cover topics such as attack vectors and their mitigation mechanisms that also apply the behaviors introduced in this document. Together, they describe the required security mechanisms that allow establishment of an SR domain of trust. Having such a well-defined trust boundary is necessary in order to operate SRv6-based services for internal traffic while preventing any external traffic from accessing or exploiting the SRv6-based services. Care and rigor in IPv6 address allocation for use for SRv6 SID allocations and network infrastructure addresses, as distinct from IPv6 addresses allocated for end-users/systems (as illustrated in Section 5.1 of [RFC8754]), can provide the clear distinction between internal and external address space that is required to maintain the integrity and security of the SRv6 Domain. Additionally, [RFC8754] defines an HMAC TLV permitting SR Endpoint Nodes in the SR domain to verify that the SRH applied to a packet was selected by an authorized party and to ensure that the segment list is not modified after generation, regardless of the number of segments in the segment list. When enabled by local configuration, HMAC processing occurs at the beginning of SRH processing as defined in [RFC8754] Section 2.1.2.1 .

This document introduces SRv6 Endpoint and SR Policy Headend behaviors for implementation on SRv6 capable nodes in the network. The headend policy definition should be consistent with the specific behavior used and any local configuration (as specified in Section 4.1.1). As such, this document does not introduce any new security considerations.

10. IANA Considerations

10.1. Ethernet Next Header Type

This document requests IANA to allocate, in the "Protocol Numbers" registry (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml), a new value for "Ethernet" with the following definition: The value 143 in the Next Header field of an IPv6 header or any extension header indicates that the payload is an Ethernet frame [IEEE.802.3_2018].

IANA has done a temporary allocation of Protocol Number 143.

10.2. SRv6 Endpoint Behaviors Registry

This document requests IANA to create a new top-level registry called "Segment Routing Parameters". This registry is being defined to serve as a top-level registry for keeping all other Segment Routing sub-registries.

Additionally, a new sub-registry "SRv6 Endpoint Behaviors" is to be created under top-level "Segment Routing Parameters" registry. This sub-registry maintains 16-bit identifiers for the SRv6 Endpoint behaviors. This registry is established to provide consistency for control plane protocols which need to refer to these behaviors. These values are not encoded in the function bits within a SID.

The range of the registry is 0-65535 (0x0000 - 0xFFFF) and has the following registration rules and allocation policies:

Range	Hex	Registration procedure	Notes
0	0x0000	Reserved	Not to be allocated
1-32767	0x0001-0x7FFF	First Come First Served [RFC8126]	
32768-34815	0x8000-0x87FF	Private Use [RFC8126]	
34816-65534	0x8800-0xFFFE	Reserved	
65535	0xFFFF	Reserved	Opaque

Table 3: SRv6 Endpoint Behaviors Registry

10.2.1. Initial Registrations

The initial registrations for the sub-registry are as follows:

Value	Hex	Endpoint behavior	Reference
0	0x0000	Reserved	Not to be allocated
1	0x0001	End	[This.ID]
2	0x0002	End with PSP	[This.ID]
3	0x0003	End with USP	[This.ID]
4	0x0004	End with PSP&USP	[This.ID]
5	0x0005	End.X	[This.ID]
6	0x0006	End.X with PSP	[This.ID]

7	0x0007	End.X with USP	[This.ID]
8	0x0008	End.X with PSP&USP	[This.ID]
9	0x0009	End.T	[This.ID]
10	0x000A	End.T with PSP	[This.ID]
11	0x000B	End.T with USP	[This.ID]
12	0x000C	End.T with PSP&USP	[This.ID]
14	0x000E	End.B6.Encaps	[This.ID]
15	0x000F	End.BM	[This.ID]
16	0x0010	End.DX6	[This.ID]
17	0x0011	End.DX4	[This.ID]
18	0x0012	End.DT6	[This.ID]
19	0x0013	End.DT4	[This.ID]
20	0x0014	End.DT46	[This.ID]
21	0x0015	End.DX2	[This.ID]
22	0x0016	End.DX2V	[This.ID]
23	0x0017	End.DT2U	[This.ID]
24	0x0018	End.DT2M	[This.ID]
25	0x0019	Reserved	[This.ID]
27	0x001B	End.B6.Encaps.Red	[This.ID]
28	0x001C	End with USD	[This.ID]
29	0x001D	End with PSP&USD	[This.ID]
30	0x001E	End with USP&USD	[This.ID]
31	0x001F	End with PSP, USP & USD	[This.ID]
32	0x0020	End.X with USD	[This.ID]
33	0x0021	End.X with PSP&USD	[This.ID]
34	0x0022	End.X with USP&USD	[This.ID]
35	0x0023	End.X with PSP, USP & USD	[This.ID]
36	0x0024	End.T with USD	[This.ID]
37	0x0025	End.T with PSP&USD	[This.ID]
38	0x0026	End.T with USP&USD	[This.ID]
39	0x0027	End.T with PSP, USP & USD	[This.ID]
40-32766		Unassigned	
32767	0x7FFF	The SID defined in RFC8754	[This.ID] [RFC8754]
32768-65534		Reserved	
65535	0xFFFF	Opaque	[This.ID]

Table 4: IETF - SRv6 Endpoint Behaviors

11. Acknowledgements

The authors would like to acknowledge Stefano Previdi, Dave Barach, Mark Townsley, Peter Psenak, Thierry Couture, Kris Michielsen, Paul Wells, Robert Hanzl, Dan Ye, Gaurav Dawra, Faisal Iqbal, Jaganbabu Rajamanickam, David Toscano, Asif Islam, Jianda Liu, Yunpeng Zhang,

Jiaoming Li, Narendra A.K, Mike Mc Gourty, Bhupendra Yadav, Sherif Toulan, Satish Damodaran, John Bettink, Kishore Nandyala Veera Venk, Jisu Bhattacharya, Saleem Hafeez and Brian Carpenter.

12. Contributors

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Dirk Steinberg
Lapishills Consulting Limited
Cyprus

Email: dirk@lapishills.com

Robert Raszuk
Bloomberg LP
United States of America

Email: robert@raszuk.net

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Bart Peirens
Proximus
Belgium

Email: bart.peirens@proximus.com

Hani Elmalky
Google
United States of America

Email: helmalky@google.com

Prem Jonnalagadda
Barefoot Networks
United States of America

Email: prem@barefootnetworks.com

Milad Sharif
SambaNova Systems
United States of America

Email: milad.sharif@sambanova.ai

David Lebrun
Google
Belgium

Email: dlebrun@google.com

Stefano Salsano
Universita di Roma "Tor Vergata"
Italy

Email: stefano.salsano@uniroma2.it

Ahmed AbdelSalam
Gran Sasso Science Institute
Italy

Email: ahmed.abdelsalam@gssi.it

Gaurav Naik
Drexel University
United States of America

Email: gn@drexel.edu

Arthi Ayyangar
Arrcus, Inc
United States of America

Email: arthi@arrcus.com

Satish Mynam
Arrcus, Inc
United States of America

Email: satishm@arrcus.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

Shaowen Ma
Juniper
Singapore

Email: mashao@juniper.net

Ahmed Bashandy
Individual
United States of America

Email: abashandy.ietf@gmail.com

Francois Clad
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Kamran Raza
Cisco Systems, Inc.
Canada

Email: skraza@cisco.com

Darren Dukes
Cisco Systems, Inc.
Canada

Email: ddukes@cisco.com

Patrice Brissete
Cisco Systems, Inc.
Canada

Email: pbrisset@cisco.com

Zafar Ali
Cisco Systems, Inc.
United States of America

Email: zali@cisco.com

Ketan Talaulikar
Cisco Systems, Inc.
India

Email: ketant@cisco.com

13. References

13.1. Normative References

[IEEE.802.3_2018]
IEEE, "802.3-2018", IEEE 802.3-2018,
DOI 10.1109/IEEESTD.2018.8457469, August 2018,
<https://ieeexplore.ieee.org/document/8457469>.

[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>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[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>.

[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>.

13.2. Informative References

[I-D.filsfils-spring-srv6-net-pgm-illustration]
Filsfils, C., Camarillo, P., Li, Z., Matsushima, S., Decraene, B., Steinberg, D., Lebrun, D., Raszuk, R., and J. Leddy, "Illustrations for SRv6 Network Programming", draft-filsfils-spring-srv6-net-pgm-illustration-03 (work in progress), September 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-04 (work in progress), August 2020.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <https://www.rfc-editor.org/info/rfc4664>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8214] Boutros, S., Sajassi, A., Salam, S., Drake, J., and J. Rabadan, "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, DOI 10.17487/RFC8214, August 2017, <https://www.rfc-editor.org/info/rfc8214>.

[RFC8317] Sajassi, A., Ed., Salam, S., Drake, J., Uttaro, J., Boutros, S., and J. Rabadan, "Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) and Provider Backbone Bridging EVPN (PBB-EVPN)", RFC 8317, DOI 10.17487/RFC8317, January 2018, <https://www.rfc-editor.org/info/rfc8317>.

Authors' Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Belgium

Email: cf@cisco.com

Pablo Camarillo Garvia (editor)
Cisco Systems, Inc.
Spain

Email: pcamaril@cisco.com

John Leddy
Individual Contributor
United States of America

Email: john@leddy.net

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Satoru Matsushima
SoftBank
1-9-1,Higashi-Shimbashi,Minato-Ku
Tokyo 105-7322
Japan

Email: satoru.matsushima@g.softbank.co.jp

Zhenbin Li
Huawei Technologies
China

Email: lizhenbin@huawei.com