SPRING C. Filsfils, Ed.
Internet-Draft P. Camarillo, Ed.
Intended status: Standards Track Cisco Systems, Inc.
Expires: June 13, 2020 J. Leddy
Individual Contributor
D. Voyer
Bell Canada
S. Matsushima
SoftBank
Z. Li
Huawei Technologies
December 11, 2019

SRv6 Network Programming
draft-ietf-spring-srv6-network-programming-06

Abstract

The SRv6 Network Programming framework enables a network operator or an application to specify a packet 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 (Service Level Agreements).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

Familiarity with the Segment Routing Header [I-D.ietf-6man-segment-routing-header] 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, Active Segment, SR Policy, Prefix SID and Adjacency SID.

The following terms used within this document are defined in [I-D.ietf-6man-segment-routing-header]: SRH, SR Source Node, Transit Node, SR Segment Endpoint Node and Reduced SRH.

NH: Next-header field of the IPv6 header. NH=SRH means that the next-header of the IPv6 header is Routing Header for IPv6(43) with the Type field set to 4.

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 behavior: A packet processing behavior executed at an SRv6 segment endpoint. Section 4 of this document defines 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.

SRH[n]: A shorter representation of Segment List[n], as defined in [I-D.ietf-6man-segment-routing-header].

When a packet is intercepted on a wire, it is possible that SRH[SL] is different from the DA.

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 an IPv6 network as an IPv6 address.

Its processing is defined in [I-D.ietf-6man-segment-routing-header] 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 packets 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 non-local route
>
> - No Match

This document formally defines behaviors and parameters for SRv6 SIDs.

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 reminder of the SID MUST be zero.

A locator may be represented as B:N where B is the SRv6 SID block (IPv6 subnet 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.

A behavior may require additional arguments that would be placed immediately after the FUNCT. ARG may contain information related to the flow, service, or any other information required by FUNCT. 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 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 routing protocol specific aspects that are outside 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

Each FIB entry indicates the behavior associated with a SID instance and its parameters.

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 a 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.

The following sub-sections detail the behaviors, introduced in this document, that a node (N) binds to a SID (S).

Section 4.16 defines flavors of some of these behaviors.

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.      Send an ICMP Parameter Problem message 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.
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 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. VRF, 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.

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End SID, send an ICMP parameter problem message to the Source Address and discard the packet. Error code 4 (SR Upper-layer Header Error) and Pointer set to the offset of the upper-layer header.

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 it is required to express any traffic-engineering policy.

An instance of the End.X 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.   Set the packet's egress adjacency to 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 Section 6.2.

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(LAG) itself and then up to one for each Layer-2 member link.

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

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
```

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 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) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the outer IPv6 Header with all its extension headers
S05. Forward the exposed IPv6 packet to the L3 adjacency J
```

Notes:
S01. 41 refers to IPv6 encapsulation as defined by IANA allocation for Internet Protocol Numbers.
S05. 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 Section 6.2.

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) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the outer IPv6 Header with all its extension headers
S05. Forward the exposed IPv4 packet to the L3 adjacency J
```

Notes:
S01. 4 refers to IPv4 encapsulation as defined by IANA allocation for Internet Protocol Numbers
S05. 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 Section 6.2.

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 and 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) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the outer IPv6 Header with all its extension headers
S05. Set the packet's associated FIB table to T
S06. Submit the packet to the egress IPv6 FIB lookup and
        transmission to the new destination
```

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 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) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the outer IPv6 Header with all its extension headers
S05. Set the packet's associated FIB table to T
S06. Submit the packet to the egress IPv4 FIB lookup and
        transmission to the new destination
```

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) {
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) {
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.    Send an ICMP Parameter Problem message 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.
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/ EVPN[RFC7432] VPWS 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 != TBD1) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the outer IPv6 Header with all its extension headers and
        forward the Ethernet frame to the OIF I.
```

Notes:
S04. 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.

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 the End.DX2V 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:

```
S04. Remove the outer IPv6 Header with all its extension headers,
        lookup the exposed VLANs in L2 table T, and forward
        via the matched table entry.
```

Notes:
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. 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 != TBD1) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the IPv6 header and all its extension headers
S05. Learn the exposed MAC Source Address in L2 Table T
S06. Lookup the exposed MAC Destination Address in L2 Table T
S07. If (matched entry in T) {
S08.    Forward via the matched table T entry
S09. } Else {
S10.    Forward via all L2 OIFs entries in table T
S11. }
```

Notes:
S05. In EVPN, the learning of the exposed inner MAC SA is done via the control 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 BUM with ESI filtering and the EVPN ETREE use-cases.

Any SID instance of this behavior is associated with a L2 table T. Additionally the behavior MAY take an argument: "Arg.FE2". It is an argument specific to EVPN ESI filtering and EVPN-ETREE used to exclude specific OIF (or set of OIFs) from L2 table T flooding.

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.DT2M behavior except for the Upper-layer header processing which is as follows:

```
S01. If (Upper-Layer Header type != TBD1) {
S02.    Send an ICMP Parameter Problem message 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.
S03. }
S04. Remove the IPv6 header and all its extension headers
S05. Learn the exposed inner MAC Source Address in L2 Table T
S06. Forward via all L2 OIFs excluding the one specified in Arg.FE2
```

Notes:
S05. In EVPN, the learning of the exposed inner MAC SA is done via control 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 the one of the SRv6 instantiations of a Binding SID [RFC8402].

An End.B6.Encaps SID is never the last segment in a SID list. Any SID instantiation is associated with an SR Policy B[I-D.ietf-spring-segment-routing-policy] 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.      Send an ICMP Parameter Problem message 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.
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 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Push a new IPv6 header with its own SRH containing B
S15.   Set the outer IPv6 SA to A
S16.   Set the outer IPv6 DA to the first SID of B
S17.   Set the outer PayloadLength, Traffic Class, FlowLabel and
          Next-Header fields
S18.   Submit the packet to the egress IPv6 FIB lookup and
          transmission to the new destination
S19. }
```

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 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, send an ICMP parameter problem message to the Source Address and discard the packet. Error code 4 (SR Upper-layer Header Error), Pointer set to the offset of the upper-layer header.

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 [I-D.ietf-6man-segment-routing-header].

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].

An End.BM SID is never the last SID, and any SID instantiation is associated with an SR-MPLS Policy B[I-D.ietf-spring-segment-routing-policy].

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.      Send an ICMP Parameter Problem message 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.
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 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Push the MPLS label stack for B
S15.   Submit the packet to the MPLS engine for transmission to the
          topmost label.
S16. }
```

When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 End.BM SID, send an ICMP parameter problem message to the Source Address and discard the packet. Error code 4 (SR Upper-layer Header Error), Pointer set to the offset of the upper-layer header.

4.16. Flavors

The PSP, USP and USD flavors are variants of the End, End.X and End.T behaviors. For each of these behaviors these flavors MAY be supported for a SID either individually or in combinations.

4.16.1. PSP: Penultimate Segment Pop of the SRH

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 of the SRH
 S14.3.      Decrease the IPv6 header Payload Length by the Hdr Ext Len
                value of the SRH
 S14.4.      Remove the SRH from the IPv6 extension header chain
 S14.5.   }
```

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 the Hdr Ext Len
                value of the SRH
 S03.3.      Remove the SRH from the IPv6 extension header chain
 S03.4.      Proceed to process the next header in the packet
 S04.     }
```

4.16.3. USD: Ultimate Segment Decapsulation

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.      Skip the SRH processing and proceed to the next header
   S04.   }
```

Further on, 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 || 4) {
S02.    Remove the outer IPv6 Header with all its extension headers
S03.    Submit the packet to the egress IP FIB lookup and
           transmission to the new destination
S04. } Else {
S05.    Send an ICMP Parameter Problem message 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.

S06. }

End.T:
S01. If (Upper-layer Header type == 41 || 4) {
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 IP FIB lookup and
           Transmission to the new destination
S05. } Else {
S06.    Send an ICMP Parameter Problem message 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.
S07. }

End.X:
S01. If (Upper-layer Header type == 41 || 4) {
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.    Send an ICMP Parameter Problem message 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.
S06. }
```

An implementation that supports the USD flavor in conjunction with the USP flavor MAY optimize the packet processing by first looking whether the conditions for the USD flavor are met, in which case it can proceed with USD processing else do USP processing.

5. Transit Behaviors

This section describes the set of basic transit behaviors. These behaviors are not bound to a SID and they correspond to source SR nodes or transit nodes [I-D.ietf-6man-segment-routing-header].

T	Transit behavior
T.Encaps	Transit behavior with encapsulation in an SRv6 policy
T.Encaps.Red	Transit behavior with reduced encaps in an SRv6 policy
T.Encaps.L2	T.Encaps applied to received L2 frames
T.Encaps.L2.Red	T.Encaps.Red applied to received L2 frames

This list can be expanded in case any new functionality requires it.

5.1. T: Transit Behavior

As per [RFC8200], if a node N receives a packet (A, S2)(S3, S2, S1; SL=1) and S2 is neither a local address nor a local SID of N then N forwards the packet without inspecting the SRH.

This means that N treats the following two packets P1 and P2 with the same performance:

P1 = (A, S2)

P2 = (A, S2)(S3, S2, S1; SL=1)

A transit node does not need to count by default the amount of transit traffic with an SRH extension header. This accounting might be enabled as an optional behavior.

A transit node includes the outer flow label in its ECMP load-balancing hash as described in [RFC6437].

5.2. T.Encaps: Transit 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.

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

The T.Encaps transit encapsulation behavior is defined as follows:

```
S01.   Push an IPv6 header with its own SRH (S3, S2, S1; SL=2)
S02.   Set outer IPv6 SA = T and outer IPv6 DA = S1
S03.   Set outer payload length, traffic class and flow label  ;;Ref1,2
S04.   Update the Next-Header value                            ;;Ref1
S05.   Decrement inner Hop Limit or TTL                        ;;Ref1
S06.   Submit the packet to the IPv6 module for transmission to S1
```

After the T.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 T.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.

Ref 1: As described in [RFC2473] (Generic Packet Tunneling in IPv6 Specification)

Ref 2: As described in [RFC6437] (IPv6 Flow Label Specification)

5.3. T.Encaps.Red: Transit with Reduced Encapsulation

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

T.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 T.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.

5.4. T.Encaps.L2: Transit with Encapsulation of L2 Frames

The T.Encaps.L2 behavior encapsulates a received Ethernet [Ethernet] 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 TBD1.

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 or frame check sequence (FCS) from the Ethernet frame upon encapsulation and the decapsulating node MUST regenerate the preamble or FCS before forwarding Ethernet frame.

5.5. T.Encaps.L2.Red: Transit with Reduced Encapsulation of L2 Frames

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

T.Encaps.L2.Red reduces the length of the SRH by excluding the first SID in teh 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. Operation

6.1. Counters

Any SRv6 capable node SHOULD implement the following set of combined counters (packets and bytes):

- CNT-1: Per local SID entry, traffic that matched that SID and was processed correctly.

- CNT-2: Per SRv6 Policy, traffic steered into it and processed correctly.

Furthermore, an SRv6 capable node SHOULD maintain an aggregate counter CNT-3 tracking the IPv6 packets received with an IPv6 Destination Address matching a local interface address that is not a locally instantiated SID and containing an SRH with a Segments Left value different from 0.

6.2. Flow-based Hash Computation

When a flow-based selection within a set needs to be performed, the source address, the destination address and the flow label 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 [I-D.ietf-spring-segment-routing-policy].

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

6.3. OAM

[I-D.ietf-6man-spring-srv6-oam] defines OAM behaviors for SRv6. This includes the definition of the SRH Flag 'O-bit', as well as additional SR Endpoint behaviors for OAM purposes.

7. Security Considerations

The security considerations for Segment Routing are discussed in [RFC8402]. More specifically for SRv6 the security considerations and the mechanisms for securing an SR domain are discussed in [I-D.ietf-6man-segment-routing-header]. Together, they describe the required security mechanisms that allow establishment of an SR domain of trust to operate SRv6-based services for internal traffic while preventing any external traffic from accessing or exploiting the SRv6-based services.

This document introduces SRv6 Endpoint and Transit Nodes behaviors for implementation on SRv6 capable nodes in the network. As such, this document does not introduce any new security considerations.

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.

This section provides a high level overview of the control-plane protocols involved with SRv6 and their specification.

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[I-D.ietf-lsr-isis-srv6-extensions]. The IGP also advertises the support for SRv6 capabilities of the node.

The presence of SIDs in the IGP do 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 not governed neither influenced in any way by a SID advertisement in the IGP.

These SIDs provide important topological behaviors for the IGP to build TI-LFA[I-D.ietf-rtgwg-segment-routing-ti-lfa] based FRR

solutions 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 [I-D.ietf-idr-bgpls-srv6-ext]. 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 [I-D.ietf-bess-srv6-services].

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 transit capabilities are signaled in which signaling protocol.

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

Table 2: SRv6 transit behaviors signaling

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

For example, a BGP-LS advertisement of the T capability of node N would indicate that node N supports the basic transit behavior. The T.Encaps behavior would describe the capability of node N to perform a T.Encaps behavior, specifically it would describe how many SIDs could be pushed by N without significant performance degradation.

The reader should also remember that every SR policy is always assigned a Binding SID. They should remember that BSIDs are advertised in BGP-LS as shown in Table 1. Hence, it is normal that Table 2 only focuses on the generic capabilities related to T.Encaps as Table 1 advertises the specific instantiated BSID properties.

9. IANA Considerations

9.1. 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 TBD1 in the Next Header field of an IPv6 header or any extension header indicates that the payload is an Ethernet [Ethernet].

9.2. 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. 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	Invalid
1-32767	0x0001-0x7FFF	FCFS	
32768-65534	0x8000-0xFFFE	Reserved	
65535	0xFFFF	Reserved	Opaque

Table 3: SRv6 Endpoint Behaviors Registry

9.2.1. Initial Registrations

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

Value	Hex	Endpoint behavior	Reference
0	0x0000	Invalid	[This.ID]
1	0x0001	End (no PSP, no USP)	[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 (no PSP, no USP)	[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 (no PSP, no USP)	[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]
13	0x000D	Reserved	-
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]
26	0x001A	Reserved	-
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]
40-32767		Unassigned	
32768-65534		Reserved	Change control under IETF
65535	0xFFFF	Opaque	[This.ID]

Table 4: IETF - SRv6 Endpoint Behaviors

Requests for allocation from within the FCFS range must include a point of contact and preferably also a brief description of how the value will be used. This information may be provided with a reference to an Internet Draft or an RFC or in some other documentation that is permanently and readily available.

10. 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 and Saleem Hafeez.

11. 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
Ericsson
United States of America

Email: hani.elmalky@gmail.com

Prem Jonnalagadda
Barefoot Networks
United States of America

Email: prem@barefootnetworks.com

Milad Sharif
Barefoot Networks
United States of America

Email: msharif@barefootnetworks.com

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
Arista
United States of America

Email: arthi@arista.com

Satish Mynam
Innovium Inc.
United States of America

Email: smynam@innovium.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

12. References

12.1. Normative References

[Ethernet]
DigitalEquipment, Intel, and Xerox, "The Ethernet -- A Local Area Network: Data Link Layer and Physical Layer (Version 2.0)", November 1982.

[I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-26 (work in progress), October 2019.

[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>.

[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>.

12.2. 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-01 (work in progress), August 2019.

[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-02 (work in progress), November 2019.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-01 (work in progress), November 2019.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-01 (work in progress), July 2019.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-03 (work in progress), October 2019.

[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-01 (work in progress), March 2019.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-05 (work in progress), November 2019.

[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>.

[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>.

[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>.

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