SRv6 and MPLS interworking
draft-agrawal-spring-srv6-mpls-interworking-08

Abstract

This document describes SRv6 and MPLS/SR-MPLS interworking and co-existence procedures.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 September 8, 2022.
Copyright Notice

Copyright (c) 2022 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 ............................................. 3
   1.1. Requirements Language ............................. 3
2. Interworking(IW) scenarios .............................. 3
   2.1. IW scenarios ........................................ 4
       2.1.1. Transport IW ................................... 4
       2.1.2. Service IW ...................................... 5
3. Terminology .............................................. 5
4. SRv6 SID behavior ....................................... 6
   4.1. End.DTM ............................................ 6
   4.2. End.DPM ............................................ 7
5. SRv6 Policy Headend Behaviors .......................... 8
   5.1. H.Encaps.M: H.Encaps applied to MPLS label stack ... 8
6. Interconnecting Binding SIDs ............................ 8
7. Interworking Procedures ................................ 8
   7.1. Transport IW ........................................ 9
       7.1.1. SR-PCE multi-domain On Demand Nexthop .......... 9
       7.1.2. BGP inter domain routing procedures ............ 11
       7.2. Service IW ........................................ 17
       7.2.1. Gateway Interworking ............................ 17
       7.2.2. Translation between Service labels and SRv6 service SID .................. 18
8. Migration and co-existence ................................ 18
9. Availability ............................................. 18
10. IANA Considerations .................................. 19
    10.1. BGP Prefix-SID TLV Types registry ................. 19
    10.2. SRv6 Endpoint Behaviors .......................... 19
11. Security Considerations ............................... 19
12. Acknowledgements ..................................... 19
13. References ............................................. 19
    13.1. Normative References ............................. 19
1. Introduction

The incremental deployment of SRv6 into existing networks require SRv6 to interwork and co-exist with SR-MPLS/MPLS. This document introduces interworking scenarios and building blocks for solutions to inter connect them.


1.1. 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.

2. Interworking (IW) scenarios

A multi-domain network (Figure 1) can be generalized as a central domain C with many leaf domains around it. Specifically, document look at a service flow from an ingress PE in an ingress leaf domain (LI), through the C domain and up to an egress PE of the egress leaf domain (LE). Each domain runs its own IGP instance. A domain has a single data plane type applicable both for its overlay and its underlay.

There are various SRv6 and SR-MPLS-IPv4 interworking scenarios possible.

Below scenarios cover various cascading of SRv6/MPLS network, e.g., SR-MPLS-IPv4 <-> SRv6 <-> SR-MPLS-IPv4 <-> SRv6 <-> SR-MPLS-IPv4, etc.

2.1. IW scenarios

2.1.1. Transport IW

Provider edge run MPLS based [RFC4364] or SRv6 Service SID based [I-D.ietf-bess-srv6-services] BGP L3(e.g.VPN) or L2(e.g.EVPN) services through service Route Reflectors. Service endpoint signaling and forwarding state provide interworking over intermediate transport.

- SRv6 over MPLS (6oM)
  - LI and LE domains are SRv6 data plane, C is MPLS data plane

Figure 1: Reference multi-domain network topology
* L3/L2 BGP SRv6 services [I-D.ietf-bess-srv6-services] between PEs. The ingress PE encapsulates the payload in an outer IPv6 header where the SRv6 Service SID the is last segment or destination address (DA).

* Forward SRv6 encapsulated traffic destined to egress PE over MPLS C domain.

  o MPLS over SRv6 (Mo6)

    * LI and LE domains are MPLS data plane, C is SRv6 data plane

    * L3/L2 BGP MPLS services [RFC4364], [RFC7432]. The ingress PE encapsulates the payload in an MPLS service label and sends it MPLS LSP to egress PE.

    * Forward encapsulated label stack to egress PE over SRv6 C domain.

Note: Easiest and most probable deployment is ship in the night i.e. supporting dual stack and IPv4 MPLS in each domain.

2.1.2. Service IW

L3/L2 service signaling discontinuity i.e. SRv6 service SID based PE interworks with BGP MPLS based PE for service connectivity. L3/L2 service BGP signaling and forwarding state provide interworking over intermediate domain.

  o SRv6 to MPLS (6toM): The ingress PE encapsulates the payload in an outer IPv6 header where the destination address is the SRv6 Service SID [I-D.ietf-bess-srv6-services]. Payload is delivered to egress PE with MPLS service label [RFC4364] that it advertised with service prefixes.

  o MPLS to SRv6 (Mto6): The ingress PE encapsulates the payload in an MPLS service label. Payload is delivered to egress PE with IPv6 header with destination address as SRv6 service SID that it advertised with service prefixes.

3. Terminology

The following terms used within this document are defined in [RFC8402]: Segment Routing, SR-MPLS, SRv6, SR Domain, Segment ID (SID), SRv6 SID, Prefix-SID.
Domain: Without loss of the generality, domain is assumed to be instantiated by a single IGP instance or a network within IGP if there is clear separation of data plane.

Node k has a classic IPv6 loopback address Ak::1/128.

A SID at node k with locator block B and function F is represented by B:k:F::

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:

IPv6 header with source address SA, destination addresses DA and SRH as next-header

SRH with SID list <S1, S2, S3> with SegmentsLeft = SL

Note the difference between the <> and () symbols: <S1, S2, S3> represents a SID list where S1 is the first 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.

4. SRv6 SID behavior

This document introduces a new SRv6 SID behavior. This behavior is executed on border routers between the SRv6 and MPLS domain.

4.1. End.DTM

The "Endpoint with decapsulation and MPLS table lookup" behavior.

The End.DTM SID MUST be the last segment in a SR Policy, and a SID instance is associated with an MPLS table.

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

S01. If (Upper-Layer Header type == 137(MPLS) ) {
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 MPLS FIB lookup for transmission according to the lookup result.
S05. } Else {
S06.   Process as per [ietf-spring-srv6-network-programming] section 4.1.1
S07. }

4.2. End.DPM

The "Endpoint with decapsulation and MPLS label push" behavior.

The End.DPM SID MUST be the last segment and a SID instance is associated with label stack.

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

S01. Remove the outer IPv6 Header with all its extension headers
S02. Push the MPLS label stack associated with S
S03. Submit the packet to the MPLS engine for transmission
5. SRv6 Policy Headend Behaviors

5.1. H.Encaps.M: H.Encaps applied to MPLS label stack

The H.Encaps.M behavior encapsulates a received MPLS Label stack
[RFC3032] packet in an IPv6 header with an SRH. Together MPLS label
stack and its payload becomes the payload of the new IPv6 packet.
The Next Header field of the SRH MUST be set to 137 [RFC4023].


The H.Encaps.M.Red behavior is an optimization of the H.Encaps.M
behavior. H.Encaps.M.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. 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. In such case, the Next Header field of the IPv6 header MUST be
set to 137 [RFC4023].

6. Interconnecting Binding SIDs

Binding Segment (BSID) is bound to SR policy [RFC8402]. Further an
SR-MPLS label can be bound to an SRv6 Policy and an SRv6 SID can be
bound to an SR-MPLS Policy. The IW SR-PCE solution Section 7.1.1
leverage these BSIDs as segments of SR policy on headend domain to
represent intermediate domain of different dataplane type. In
summary, an intermediate domain of different data plane type is
represented by BSID of ingress domain data plane type in SID list.

7. Interworking Procedures

Figure 1 shows reference multi-domain network topology and Section 2
its description. The procedure in this section are illustrated using
the topology.

Following is assumed for data plane support of various nodes:

- Nodes 2,3,5,6,8,9 are provider(P) routers which need to support
  single data plane type.
- 1 and 10 are PEs. They support single data plane type in overlay
  and underlay.
- Border routers 4 and 7 need to support both the SRv6 and SR-MPLS-
  IPv4 data plane.
A VPN route is advertised via service RRs (S-RR) between an egress PE (node 10) and an ingress PE (node 1).

For illustrations, the SRGB range starts from 16000 and prefix SID of a node is 16000 plus node number

7.1. Transport IW

As described in Section 2.1.1, transport IW requires:

- For 6oM, tunnel traffic destined to SRv6 Service SID of egress PE over MPLS C domain.
- For Mo6, Tunnel MPLS label stack bound to IPv4 loopback address of egress PE over SRv6 C domain.

This draft enhances two well-known solutions to achieve above:

- An SR-PCE [RFC8664] multi-domain On Demand Next-hop (ODN) SR policy [I-D.ietf-spring-segment-routing-policy] stitching end to end across different data plane domains using interconnecting binding SIDs. These procedures can be used when overlay prefixes are signaled with a color extended community [I-D.ietf-idr-tunnel-encaps].
- BGP Inter-Domain routing procedures advertising PE locator or IPv4 Loopback address for best effort end to end connectivity.

7.1.1. SR-PCE multi-domain On Demand Nexthop

This procedure provides a best-effort as well as a path that satisfies the intent (e.g. low latency), across multiple domains. Service routes (VPN/EVPN) are received on ingress PE with color extended community from egress PE. A Color is a 32-bit numerical value that associates an SR Policy with an intent [I-D.ietf-spring-segment-routing-policy]. Ingress PE does not know how to compute the traffic engineered path through the multi-domain network to egress PE and request SR-PCE for it. The SR-PCE is aware of interworking requirement at border nodes as its fed with BGP-LS topological information from each domain. It programs intermediate domain data plane specific policy on border nodes for the given intent and represents it in end to end path SID list on ingress PE leveraging Section 6.

Below sections describe 6oM and Mo6 IW with SR-PCE
7.1.1.1. 6oM

Service prefix (e.g. VPN or EVPN) is received on head-end (node 1) with color extended community (C1) from egress PE (node 10) with SRv6 service SID. The PCE computes (C1,10) path via node 2, 5 and 8. It programs an SR policy at border node 4 with segment list node 5 and 7 bounded to an End.BM BSID [RFC8986]. SR-PCE responds back to node 1 with SRv6 segments along required SLA including End.BM at node 4 to traverse SR-MPLS-IPv4 C domain.

For example, SR-PCE create SR-MPLS policy (C1,7) at node 4 with segments <16005,16007>. It is bound to End.BM behavior with SRv6 BSID as B:4:BM-C1-7::

The data plane operations for the above-mentioned interworking example are described in the following:

Node 1 performs SRv6 function H.Encaps.Red with VPN service SID and SRv6 Policy (C1,10):
Packet leaving node 1 IPv6 ((A:1::, B:2:E::) (B:10::DT4, B:8:E::, B:4:BM-C1-7::; SL=3))

Node 2 performs End function
Packet leaving node 2 IPv6 ((A:1::, B:4:BM-C1-7::) (B:10::DT4, B:8:E::, B:4:BM-C1-7::; SL=2))

Node 4 (border router) performs End.BM function
Packet leaving node 4 MPLS (16005,16007,2)((A:1::, B:8:E::) (B:10::DT4, B:8:E::) (B:4:BM-C1-7::; SL=1)).

Node 7 performs a native IPv6 lookup on due PHP behavior for 16007
Packet leaving node 7 IPv6 ((A:1::, B:8:E::) (B:10::DT4, B:8:E::, B:4:BM-C1-7::; SL=1))

Node 8 performs End(PSP) function
Packet leaving node 8 IPv6 ((A:1::, B:10::DT4))

Node 10 performs End.DT function and lookups IP in VRF and send traffic to CE.

7.1.1.2. Mo6

Refer Section 2.1.1 for Mo6 scenario. MPLS Service prefix (e.g. VPN or EVPN) is received on head-end (node 1) with color extended community (C1) from egress PE (node 10). The PCE computes color-aware C1 path via node 2, 5 and 8. It programs a SRv6 policy bound to MPLS BSID at border node 4 with SRv6 segment list along required color-aware path with last segment of behavior End.DTM Section 4.1. SR-PCE
responds back to node 1 with MPLS segment list including MPLS BSID of SRv6 policy at node 4 to traverse SRv6 core domain.

For example, SR-PCE create SRv6 policy (C1,7) at node 4 with segments <B:5:E::,B:7:DTM::>. It is bound to MPLS BSID 24407.

The data plan operations for the above-mentioned interworking example are described in the following:

1. Node 1 performs MPLS label stack encapsulation with VPN label and SR-MPLS Policy (C1,10):
   Packet leaving node 1 towards 2 (Note: PHP of node 2 prefix SID):
   MPLS packet (16004,24407,16008,16010,vpn_label)

2. Node 2 forwards traffic towards 4 (PHP of 16004)
   Packet leaving node 2 MPLS packet (24407,16008,16010,vpn_label)

3. Node 4 steers MPLS traffic into SRv6 policy bound to 24407
   Packet leaving node 4 IPv6(A:4::, B:5:E::) (B:7:DTM:: ; SL=1)NH=137) MPLS((16008,16010,vpn_label)

   Packet leaves node 7 towards node 8(PHP of 16008) MPLS packet (16010,vpn_label)

5. Node 8 forwards traffic towards 10 (PHP of 16010)
   Packet leaving node 8 MPLS packet (vpn_label)

6. Node 10 performs vpn_label lookup and send traffic to CE.

7.1.2. BGP inter domain routing procedures

Procedures similar to BGP 3107 [I-D.ietf-mpls-seamless-mpls] to advertise PE locators or IPv4 loopbacks transport reachability in multi-domain network. Also Next hop self on border routers which provide independence of intra domain tunnel technology.

Below sections describe 6oM and Mo6 IW with BGP procedures

7.1.2.1. 6oM

Refer Section 2.1.1 for 6oM scenario. SRv6 based L3/L2 BGP services are signaled with SRv6 Service SID between PEs through Service RRs with no color extended community. Ingress PEs need reachability to remote locator to send traffic to SRv6 service SID.
o Egress border router learns local PE locators through IGP. These should be redistributed in BGP like any IPv6 global prefixes. Alternatively, locator is advertised by PE in the BGP IPv6 unicast address family (AFI=2,SAFI=1) to border nodes.

o Egress border router advertise LE domain PE locators in BGP IPv6 LU[AFI=2/SAFI=4] with local label (explicit NULL) to ingress border router with IPv4 next hops. These next hops have SR-MPLS-IPv4 LSP paths built in C domain. It may advertise summary prefix covering all locators in LE domain.

o If ingress border router advertise remote locators in LI domain to ingress PE in BGP address family (AFI=2,SAFI=1), it attaches local End behavior as SRv6 SID in Prefix-SID attribute TLV type 5 [I-D.ietf-bess-srv6-services]. Alternatively, it may leak remote locators in LI IGP domain such that P routers also have reachability.

o Ingress PE learn remote locator over BGP IPv6 address family AFI=2, SAFI=1 or through LI IGP. When learnt through BGP, SRv6 SID carried in Prefix-SID attribute TLV 5 tunnels traffic to ingress border node in LI domain as P routers (node 2 and 3) will not be aware of remote locator.

Control plane example:

1. Routing Protocol(RP) @10:
   
   * In ISIS advertise locator B:10::/48
   
   * BGP AFI=1,SAFI=128 originates a VPN route RD:V/v via B:10::1 and Prefix-SID attribute B:10:DT4::. This route is advertised to service RR.

2. RP @ 7:
   
   * ISIS redistribute B:10::/48 into BGP
   
   * BGP Originates B:10::/48 in AFI=2/SAFI=4 with next hop node 7 and label explicit null among border routers.

3. RP @ 4:
   
   * BGP learns B:10::/48 with next hop node 7 and outgoing label.
   
   * BGP advertise B:10::/48 in AFI=2/SAFI=1 with next hop B:4::1 and Prefix-SID attribute tlv type 5 carrying local End behavior function B:4:END:: to node 1
* Alternatively, BGP redistributes remote locator or summary route in LI domain IGP.

4. RP @ 1:

* BGP learns B:10::/48 via B:4::1 and Prefix-SID attribute TLV type 5 with SRv6 SID B:4:END::

* Alternatively, B:10::/48 or summary route reachability is learned through ISIS

* BGP AFI=1, SAFI=128 learn service prefix RD:V/v, next hop B:10::1 and PrefixSID attribute TLV type 5 with SRv6 SID B:10:DT4

FIB state

@1: IPv4 VRF V/v => H.Encaps.red <B:4:END::, B:10:DT4::> with SRH, SRH.NH=IPv4

@4: IPv6 Table: B:4:END:: => Update DA with B:10:DT4::, set IPv6.NH=IPv4, pop the SRH

@4: IPv6 Table: B:10::/48 => push MPLS label 2 (Explicit NULL), push MPLS Label 1 6007

@7: MPLS label 2 => pop and lookup next IPv6 DA

@7: IPv6 Table B:10::/48 => forward via ISIS path to 10

@10: IPv6 Table B:10:DT4:: => pop the outer header and lookup the inner IPv4 DA in the VRF

7.1.2.2. Mo6

Refer Section 2.1.1 for Mo6 scenario. MPLS based L3/L2 BGP services are signaled with IPv4 next-hop of PE through Service RRs with no color extended community. Ingress PE need labelled reachability to remote PE IPv4 loopback address advertised as next hop with service routes.


Following are options and protocol extensions to tunnel IPv4 PE loopback LSP through SRv6 C domain

7.1.2.2.1. Tunnel BGP LU LSP across SRv6 C domain

Intuitive solution for an MPLS-minded operator

o Existing BGP-LU label cross-connect on border routers for each PE IPv4 loopback address.

o The lookups at the ingress border router are based on BGP3107 label as usual
o Just the SR-MPLS IGP label to next hop is replaced by an IPv6 tunnel with DA = SRv6 SID associated with DTM behavior in C domain.

o Ingress border router forwarding perform 3107 label swap and H.Encaps.M with DA = SRv6 SID associated with DTM behavior

o Similar to MPLS-over-IP

Following section describes how existing BGP LU updates between border routers may carry SRv6 SID associated with DTM behavior to tunnel LSP across SRv6 C domain

7.1.2.2.1.1. SRv6 label route tunnel TLV

This document introduces a new TLV called "SRv6 label route tunnel" TLV of the BGP Prefix-SID Attribute to achieve signaling of SRv6 SIDs to tunnel MPLS packet with label in NLRI at the top of its label stack through SRv6/IPv6 domain. Behavior which may be encoded but not limited to is End.DTM. SRv6 label route tunnel TLV signals "AND" semantics i.e. push label signaled in NLRI and perform H.Encaps.M with DA as SRv6 SID signaled in TLV.

o Reminder: RFC 8669 introduced Prefix-SID attribute with TLV type 1 for label index and TLV type 3 for Originator SRGB for AFI=1/2 and SAFI 4 (BGP LU)

o This document extends the BGP Prefix-SID attribute [RFC8669] to carry new "SRv6 label route tunnel" TLV. This document limits the usage of this new TLV to AFI=1/2 SAFI 4. The usage of this TLV for other AFI/SAFI is out of scope of this document.

o "SRv6 label route tunnel" TLV is encoded exactly like SRv6 Service TLVs in Prefix-SID Attribute [I-D.ietf-bess-srv6-services] with following modification:

1. TLV Type (1 octet): This field is assigned values from the IANA registry "BGP Prefix-SID TLV Types". It is set to 7 for "SRv6 label route tunnel" TLV.

2. No transposition scheme is allowed i.e. transposition length MUST be 0 in SRv6 SID Structure Sub-Sub-TLV

o Possibility of label encapsulation when dataplane has LSP to next hop irrespective of SRv6 SID signaled in "SRv6 label route tunnel" of Prefix-SID attribute. This allows existing MPLS data plane to keep operating.
Control plane example

1. Routing Protocol (RP) @10:
   * ISIS originates its IPv4 PE loopback with Node SID 16010
   * BGP AFI=1, SAFI=4 originate IPv4 loopback address with next hop node 10 and optionally label index=10 in Label-Index TLV of Prefix-SID attribute.
   * BGP AFI=1, SAFI=128 originates a VPN route RD:V/v next hop node 10. This route is advertised to service RR.

2. RP @ 7:
   * ISIS v6, advertise locator B:7::/48 in C domain
   * BGP learns node 10 IPv4 loopback address with outgoing label. It allocates local label (based on label index if present) and programs label swap to outgoing label and MPLS LSP to next hop.
   * BGP AFI=1, SAFI=4 advertise IPv4 loopback address of node 10 to node 4. NLRI label is set to local label and SRv6 SID B:7:DTM:: carried in SRv6 SID Information Sub-TLV of "SRv6 label route tunnel" TLV in Prefix-Sid attribute. If received, label index=10 in Label-Index TLV of Prefix-SID attribute is also signaled.

3. RP @ 4:
   * ISIS v4 originates its IPv4 loopback with prefix SID 16004 in L1 domain.
   * BGP learns node 10 IPv4 loopback address from node 7 with outgoing label. It allocate local label (based on label index if present) and programs label swap and H.Encaps.M.red with IPv6 header destination address as SRv6 SID received in "SRv6 label route tunnel" TLV of Prefix-Sid attribute i.e. B:7:DTM::.
   * BGP AFI=1, SAFI=4 advertise IPv4 Loopback address of node 10 to node 1. NLRI label is set to local label and do not signal "SRv6 label route tunnel" TLV in Prefix-SID attribute.

4. RP @ 1:
* BGP learns IPv4 loopback address of node 10 from node 4 with outgoing label. It programs route to push outgoing label and MPLS LSP to next hop i.e. node 4

* BGP AFI=1, SAFI=128 learn service prefix RD:V/v, next hop IPv4 loopback address of node 10 and service label.

Forwarding state at different nodes:

- @1: IPv4 VRF: V/v => out label=vpn_label, next hop=IPv4 address of node 10
- @1: IPv4 table: IPv4 address of node 10 => out label=16010, next hop=node4
- @1: IPv4 table: IPv4 address of node 4 => out label=16004, next hop=interface to reach 2
- @4: MPLS Table: 16010 => out label=16010, H.Encaps.M.red with DA=B:7:DTM::
- @4: IPv6 table: B:7::/48 => next hop=interface to reach 5
- @7: SRv6 My SID table: B:7:DTM:: => decaps IPv6 header and lookup top label.
- @7: MPLS table: 16010 => out label=16010, next hop=interface to reach 8
- @10: MPLS table: vpn label => pop label and lookup the inner IPv4 DA in the VRF

7.1.2.2.2. Label and SRv6 SID translation per BGP LU route

- For each PE IPv4 loopback address, existing BGP-LU label cross-connect on area border routers is replaced by label to SRv6 SID cross-connect or vice versa on border routers.

- Advertise SRv6 SID associated with End.DPM behavior for each BGP LU route (IPv4 loopback address of PE) received from LE domain on egress border router

- Lookup of SRv6 SID result in decaps of IPv6 header and push of BGP LU outgoing label and MPLS LSP to next hop.

- Advertise BGP LU route with SRv6 SID to ingress border router.

- Ingress border router allocate local label and advertise to LI domain.

- The lookups at the ingress border router are based on BGP3107 label as usual. Lookup results SRv6 SID of DPM behavior signaled by egress border node. Decap BGP3107 label and perform H.Encaps.M with DA = SRv6 SID.

Following section describes how existing BGP LU updates between border routers may carry SRv6 SID associated with DPM behavior

- Reminder: RFC 8669 introduced Prefix-SID attribute with TLV type 1 for label index and TLV type 3 for Originator SRGB for AFI=1/2 and SAFI 4 (BGP LU)
This document extends the BGP Prefix-SID attribute [RFC8669] to carry "SRv6 L3 Service TLV" defined in [I-D.ietf-bess-srv6-services] with AFI=1/2 SAFI 4 route.

TLV is encoded exactly like SRv6 Service TLVs in Prefix-SID Attribute [I-D.ietf-bess-srv6-services]

7.2. Service IW

As described in Section 2.1.2 Service IW need BGP SRv6 based L2/L3 PE interworking with BGP MPLS based L2/L3 PE.

There are a number of different ways of handling this scenario as detailed below.

7.2.1. Gateway Interworking

Gateway is router which supports both BGP SRv6 based L2/L3 services and BGP MPLS based L2/L3 services for a service instance (e.g. L3 VRF, EVPN EVI). It terminates service encapsulation and perform L2/L3 destination lookup in service instance.

- A border router between SRv6 domain and SR-MPLS-IPv4 domain is suitable for Gateway role.
- Transport reachability to SRv6 PE and gateway locators in SRv6 domain or MPLS LSP to PE/gateway IPv4 Loopbacks can be exchanged in IGP or through mechanism detailed in Section 2.1.1.
- Gateway exchange BGP L2/L3 service prefix with SRv6 based Service PEs via set of service RRs. This session will learn/advertise L3/L2 service prefixes with SRv6 service SID in prefix SID attribute [I-D.ietf-bess-srv6-services].
- Gateway exchange BGP L2/L3 service prefix with MPLS based Service PEs via set of distinct service RRs. This session will learn/advertise L3/L2 service prefixes with service labels [RFC4364] [RFC7432].
- L2/L3 prefix received from a domain is locally installed in service instance and re advertised to other domain with modified service encapsulation information.
- Prefix learned with SRv6 service SID from SRv6 PE is installed in service instance with instruction to perform H.Encaps. It is advertised to MPLS service PE with service label. When gateway receives traffic with service label from MPLS service PE, it perform destination lookup in service instance. Lookup result in
instruction to perform H.Encaps with DA being SRv6 Service SID learnt with prefix from SRv6 PE.

- Prefix learned with MPLS service label from MPLS service PE is installed in service instance with instruction to perform service label encapsulation and send to MPLS LSP to nexthop. It is advertised to SRv6 service PE with SRv6 service SID of behavior (e.g. DT4/DT6/DT2U) [RFC8986]. When gateway receives traffic with SRv6 Service SID as DA of IPv6 header from SRv6 service PE, it perform destination lookup in service instance after decaps of IPv6 header. Lookup result in instruction to push service label and send it to nexthop.

Couple of border routers can act as gateway for redundancy. It can scale horizontally by distributing service instance among them.

7.2.2. Translation between Service labels and SRv6 service SID

This is similar to inter-as option B control plane procedures described in [RFC4364].

This would be described in future version of draft.

8. Migration and co-existence

In addition, the draft also addresses migration and coexistence of the SRv6 and SR-MPLS-IPv4. Co-existence means a network that supports both SRv6 and MPLS in a given domain. This may be a transient state when brownfield SR-MPLS-IPv4 network upgrades to SRv6 (migration) or permanent state when some devices are not capable of SRv6 but supports native IPv6 and SR-MPLS-IPv4.

These procedures would be detailed in a future revision.

9. Availability

- Failure within domain are taken care by existing FRR mechanisms [I-D.ietf-rtgwg-segment-routing-ti-lfa].


- Convergence on failure of border routers can be achieved by well known methods for BGP inter domain routing approach:

  * BGP Add Path provide diverse path visibility
* BGP backup path pre-programming
* Sub-second convergence on border router failure notified by local IGP.

10. IANA Considerations

10.1. BGP Prefix-SID TLV Types registry

This document introduces a new TLV Type of the BGP Prefix-SID attribute. IANA is requested to assign Type value in the registry "BGP Prefix-SID TLV Types" as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>SRv6 label route tunnel TLV</td>
<td>&lt;this document&gt;</td>
</tr>
</tbody>
</table>

10.2. SRv6 Endpoint Behaviors

This document introduces a new SRv6 Endpoint behavior "End.DTM". IANA is requested to assign identifier value in the "SRv6 Endpoint Behaviors" sub-registry under "Segment Routing Parameters" registry.

<table>
<thead>
<tr>
<th>Value</th>
<th>Hex</th>
<th>Endpoint behavior</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>TBD</td>
<td>End.DTM</td>
<td>&lt;this document&gt;</td>
</tr>
</tbody>
</table>

11. Security Considerations

12. Acknowledgements

The authors would like to acknowledge Kamran Raza, Dhananjaya Rao, Stephane Litkowski, Pablo Camarillo, Ketan Talaulikar

13. References

13.1. Normative References

[I-D.ietf-bess-srv6-services]


13.2. Informative References

[I-D.ietf-idr-tunnel-encaps]

[I-D.ietf-mpls-seamless-mpls]

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

Authors’ Addresses

Swadesh Agrawal (editor)
Cisco Systems

Email: swaagraw@cisco.com

Zafar ALI
Cisco Systems

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems

Email: cfilsfil@cisco.com
Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Gaurav Dawra
LinkedIn
USA
Email: gdawra.ietf@gmail.com

Zhenbin Li
Huawei Technologies
China
Email: lizhenbin@huawei.com
Abstract

Path Tracing provides a record of the packet path as a sequence of interface ids. In addition, it provides a record of end-to-end delay, per-hop delay, and load on each egress interface along the packet delivery path.

Path Tracing allows to trace 14 hops with only a 40-bytes IPv6 Hop-by-Hop extension header.

Path Tracing supports fine grained timestamp. It has been designed for linerate hardware implementation in the base pipeline.

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 5 September 2022.

Filsfils, et al. Expires 5 September 2022
1. Introduction

Path Tracing provides a record of the packet path as a sequence of interface ids. In addition, it provides a record of end-to-end delay, per-hop delay, and load on each egress interface along the packet delivery path.
Path Tracing allows to trace 14 hops with only a 40 bytes IPv6 Hop-by-Hop header. The overhead is lower than [INT], [I-D.ietf-ippm-ioam-data], [I-D.song-opsawg-ifit-framework], and [I-D.kumar-ippm-ifa].

Path Tracing supports fine-grained timestamps. It has been designed for linerate hardware implementation in the base pipeline.

Path Tracing is applicable to both SR-MPLS [RFC8660], as well as SRv6 [RFC8986]. This document defines the Path Tracing specification for the SRv6 dataplane. The SR-MPLS dataplane will be detailed in a separate document.

The specification proposed in this document has been demonstrated successfully in different interoperable hardware platforms at linerate (Section 10).

2. Terminology

The following terms used within this document are defined in [RFC8402], [RFC8754] and [RFC8986]: Segment Routing (SR), SR Domain, Segment ID (SID), SRv6, SRv6 SID, SR Policy, Segment Routing Header (SRH), SR source node, transit node, SR Endpoint, SA, DA.

The following terms are used in this document as defined below:

PT: Path Tracing

MCD: Midpoint Compressed Data (MCD). Information that every transit router adds to the packet for PT purposes. Defined in Section 3 of this document.

HbH-PT: IPv6 Hop-by-Hop [RFC8200] Path Tracing Option used for PT. It contains a stack of MCDs. It is defined in Section 8.1 of this document.

SRH PT-TLV: SRH TLV defined in Section 8.2 of this document.

PT Source: A Source node that starts a PT Probing Instance (defined in Section 4) and generates PT probes.

PT Midpoint: A transit node that performs plain IPv6 forwarding (or SR Endpoint processing) and in addition records PT information in the HbH-PT.
PT Sink: A node that receives PT probes sent from the SRC containing the information recorded by every PT Midpoint along the path, and forwards them to a regional collector after recording its PT information.

RC: Regional collector that receives PT probes, parses, and stores them in TimeSeries Database. It uses the information in the HbH-PT and the SRH PT-TLV to construct the packet delivery path as well as the timestamp at each node.

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. Midpoint Compressed Data

Every PT Midpoint along the packet delivery path -from Source to Sink- records its PT information into the HbH-PT header. This information is known as Midpoint Compressed Data (MCD). It contains the following information:

* **MCD.OIF (Outgoing Interface ID):** An 8-bit or 12-bit interface ID associated with the egress physical port of the router
  
  - The interface ID is assigned by an operator. The Interface IDs are not globally unique across the entire network. Indeed the same Interface ID may be repeated multiple times in the network as long as the end-to-end path can be deterministically inferred based on the chain of Interface IDs.
  
  - The programming of the Interface ID in the device may be done by CLI/NETCONF or any other means, and it is out of the scope of this document.
  
  - The usage of an 8-bit or 12-bit Interface ID is an operator choice, but the Interface ID size MUST be consistent across the entire network.
  
  - In case of Link Aggregation Groups (LAG/bundle) [LAG], each one of the members is configured with a different interface ID.

* **MCD.OIL (Outgoing Interface Load):** A 4-bit representation of the egress interface load (i.e., current throughout relative to the port bandwidth).
- The load is represented using a 4-bit value in logarithmic scale. This allows more granular information as the load is higher.

* MCD.TTS (Truncated Timestamp): An 8-bit timestamp encoding the time at which the packet egress the router.

- The 8-bit TTS has various possible significance depending on the link type. This is known as Time Template, and it is configured by the operator. For example, if the link is intercontinental, the 8-bit TTS encodes 7 bits of milliseconds and 1 bit of microseconds; whereas if the link is within an DC, the 8-bit TTS encodes 2 bits of milliseconds and 6 bits of microseconds.

- Each egress port in the device is configured with one Time Template.

- Note: all routers across the network MUST have time-synchronization. The mechanism used for time synchronization is out of the scope of this draft.

4. PT Probing Instance

The controller configures a PT Probing Instance at the source node. A PT Probing Instance is configured with the following parameters:

* SA: the source address of the PT probe. Typically, it is the loopback address of the PT SRC.

* Session ID: A 16-bit value.

* Probe-rate: Number of probes per second to generate as part of this PT Probing Instance. The probe-rate is the aggregate of the probes generated across all the sweeping ranges.

* SRv6 SID List: The SRv6 SID list associated with the packet. The last SID is the Sink node.

* DSCP value

* Hop-limit Value

* IPv6 Flow-Label sweeping range:
- If set, different Flow-Label values must be used in the probe packets. It may be specified as a range of specific Flow-Label values to enumerate, or it may be specified as the number of different random Flow-Label values to use in a round-robin.

* HbH-PT size

* MTU sweeping range:
  - If set, payload must be included at the end of the packet to test different packet sizes.

5. PT Source Node Dataplane Behavior

For each configured PT Probing Instance, according to the probe-rate, the PT SRC generates a PT probe packet as follows:

S01. Generate a new IPv6 packet
S02. Set the IPv6 SA as per PT Probing Instance configuration
S03. Set the IPv6 DA to the first SID from the SRv6 SID List
S04. Set the IPv6 Next Header field to 43 (SRH)
S05. Set the DSCP and Flow Label values as per PT Probing Instance configuration
S06. Append an IPv6 Hop-by-Hop header with the Hop-by-Hop Path Tracing option (HbH-PT)
S07. Set all bits of the HbH-PT MCD Stack to zero
S08. Append an SRH
S09. Set the SRH Next Header field to 59 (IPv6 No Next Header)
S10. Write the SID list in the SRH
S11. Append the SRH PT-TLV
S12. Add padding bytes after the SRH to reach the desired packet size as per the MTU sweeping range configuration
S13. Set the session ID field of the SRH PT-TLV as per PT Probing Instance configuration
S14. Set the Sequence Number field of SRH PT-TLV and increase local counter
S15. Perform an IPv6 FIB lookup to determine the Outgoing Interface (IFACE-OUT) on which packet will be forwarded
S16. Record Transmit 64-bit timestamp (SRC.T64) in the T64 field of the SRH PT-TLV
S17. Record IFACE-OUT ID (SRC.OIF) in the IF_ID field of the SRH PT-TLV
S18. Record IFACE-OUT Load (SRC.OIL) in the IF_LD field of the SRH PT-TLV
S19. Forward the packet via IFACE-OUT

Notes:
The pseudocode describes local processing at a node. An implementation of the pseudocode is compliant as long as the externally observable wire protocol is as described in the pseudocode.

6. PT Midpoint Node Dataplane Behavior

When a midpoint node receives an IPv6 packet that contains an IPv6 HbH-PT option, the node processes the HbH-PT as follows:

S01. When processing HbH-PT option {
S02.    Compute the MCD information as per Section 3
S03.    HbH-PT.MCD_Stack[MCD_Size:HbH-PT.OPT_Data_Len-1] = HbH-PT.MCD_Stack[0:HbH-PT.OPT_Data_Len-(MCD_Size+1)]
            //Shift HbH-PT MCD Stack to the right by MCD_Size bytes
S04.    HbH-PT.MCD_Stack[0:MCD_Size-1] = MCD[0:MCD_Size-1]
            //Push the MCD at the beginning of the Stack
S05. }

Notes:

* The PT Midpoint behavior MUST be implemented in the normal pipeline to experience the regular datapath (i.e., linerate). Offloading the processing of this option to either the slow-path or a co-processors is not acceptable and yields invalid results.

7. PT Sink Node Dataplane Behavior


It is a Binding SID instantiated, at Sink nodes, that encapsulates the packet with a new IPv6 header, an SRH that contains the SID list associated to End.B6.TEF SID and an SRH PT-TLV that is used to carry Path Tracing information of Sink node.

When N receives a packet whose IPv6 DA is S and S is a local End.B6.TEF SID, N does the following:
S01. Record Rx 64-bit timestamp (SNK.T64)
S02. Record incoming interface ID (SNK.IIF)
S03. Record incoming interface Load (SNK.IIL)
S04. Push a new IPv6 header
S05. Set the IPv6 SA to the Sink node loopback
S06. Set the IPv6 DA to the first SID in the SRv6 SID List
S07. Set the IPv6 Next Header field to 43 (SRH)
S08. Append an SRH
S09. Set the SRH Next Header field to 41 (IPv6)
S10. Write the SID list in the SRH
S11. Append the SRH PT-TLV
S12. Set the session ID field of the SRH PT-TLV to zero
S13. Set the Sequence Number field of the SRH PT-TLV to zero
S14. Write SNK.T64 in the T64 field of the SRH PT-TLV
S15. Write SNK.IIF in the IF_ID field of the SRH PT-TLV
S16. Write SNK.IIL in the IF_LD field of the SRH PT-TLV
S17. Submit the packet to the egress IPv6 FIB lookup for transmission to the new destination

Notes:
* The pseudocode describes local processing at a node. An implementation of the pseudocode is compliant as long as the externally observable wire protocol is as described in the pseudocode.

8. PT Headers

8.1. IPv6 Hop-by-Hop Path Tracing Option

This document defines a new IPv6 Path Tracing option to be carried in the IPv6 Hop-by-Hop Header. The option has the following format:

```
+----------------+-----------------+
|   Option Type  | Opt Data Len    |
+----------------+-----------------+
|                 | MCD Stack       |
+----------------+-----------------+
```

Figure 1: IPv6 Hop-by-Hop Path Tracing Option Format

Where:
* Option Type: TBA1-1
  - The 3 high-order bits of the option must be set to 001
- 00: Skip HbH for nodes that don’t support the HbH-PT Option
- 1: update HbH-PT for nodes that support the HbH-PT Option

- Opt Data Len: the length of the MCD stack in bytes.

Note: The IPv6 Path Tracing Option has a variable length. It is
RECOMMENDED that implementations support a 38-octet HbH-PT Option.
The operator, upon configuring the Source node behavior, MUST select
an option length that is supported by all the routers in the network.

8.2. SRH Path Tracing TLV

We define a new SRH TLV, called "Path Tracing TLV" ("SRH PT-TLV" for
short). It has the following format:

```
+----------+----------+----------+----------+
| Type     | Length   | IF_ID    | IF_LD    |
+----------+----------+----------+----------+
|          |          |          |          |
| T64      |          |          |          |
+----------+----------+----------+----------+
|          |          |          |          |
| Session ID |         | Sequence Number |
+----------+----------+----------+----------+
```

Figure 2: SRH Path Tracing TLV Format

Where:

* Type: TBA2-1
* Length: 14
* IF_ID: 12-bit Interface ID
* IF_LD: 4-bit Interface Load
* T64: 64-bit PTP Timestamp
* Session ID: Session identifier set by SRC node generating the
  probes. Used to co-relate probes of the same session. Value of
  zero means unset.
* Sequence Number: the sequence number of the probe set by SRC node
  generating the probes. Value of zero means unset.
Note: The SRH PT-TLV is generated by both the PT SRC and the PT SNK. When used at the PT SNK node, the Session ID, and Sequence Number fields MUST be set to zero.

9. Benefits

* Low overhead:
  - A 40Byte Hop-By-Hop header allows for 14 hops path measurements: 1 at the PT SRC, 12 at PT Midpoint routers and 1 at the PT SNK
  - PT has the lowest MTU overhead compared to alternative solutions such as [INT], [I-D.ietf-ippm-ioam-data], [I-D.song-opsawg-ifit-framework], and [I-D.kumar-ippm-ifa].

* Linerate and HW friendliness:
  - Implemented at linerate in current hardware, using the regular forwarding pipeline. No offloading to co-processors or slow-path whose databases might defer from forwarding pipeline.
  - Leverages mature hardware capabilities (basic shift operation); no packet resizing at every node along the path
  - High number of diverse linerate interoperable hardware Implementations (see Section 10)

* Scalable Fine-grained Timestamp:
  - 64bit at PT SRC and PT SNK
  - 8bit at PT Midpoint leveraging flexible per-outgoing-link template allowing diverse link types in the same measurement (e.g., DC, metro, WAN)

* Scalable Load measurement

10. Implementation Status

Editorial note: Please remove this section prior publication.

The following routing platforms have participated in an interop testing:

* Cisco 8802 (based Cisco Silicon One Q200)
* Cisco ASR9904 with Lightspeed linecard
* Cisco NCS5508 (based on Broadcom Jericho2 platform)

* Cisco Nexus N3K-C3464C (based on Barefoot Tofino)

* Marvel Prestera Falcon

The following open-source software networking stacks have also participated in the interop:

* FD.io VPP

* Linux Kernel

The following opensource applications also have extensions to support Path Tracing:

* Wireshark

* Tcpdump

* P4 implementation for software switch

11. 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 to 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.

This document defines the Path Tracing architecture, which is deployed on a secured SRv6-domain. As such, all the security considerations defined in [RFC8754], [RFC8402], and [RFC8986] are applicable.

In addition, any border router in an SR Domain network where Path Tracing is enabled, MUST support the configuration of the following ACLs:
* If there is a packet coming from an external interface destined towards an internal interface that contains an IPv6 Hop-by-Hop header with a Path Tracing option, then such packet is silently dropped.

* If there is a packet coming from an internal interface destined towards an external interface that contains an IPv6 Hop-by-Hop header with a Path Tracing option, then such packet is silently dropped.

These ACLs SHOULD be enabled by default. An operator MAY disable them individually based on local configuration.

The processing of IPv6 Hop-by-Hop headers could sometimes be used as an attack vector to overload the CPU of the router. As defined in Section 6 of this document, the HBH-PT option MUST be processed at line rate. Therefore there is no impact on the router’s CPU.

12. IANA Considerations

This document has two actions for IANA:

12.1. Destination Options and Hop-by-Hop Options

This I-D requests IANA to allocate a new entry in the "Destination Options and Hop-by-Hop Options" sub-registry under the top-level registry "Internet Protocol Version 6 (IPv6) Parameters":

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA1-1</td>
<td>Path Tracing</td>
<td>[This.ID]</td>
</tr>
</tbody>
</table>

Note: The 3 high-order bits must be 001.

12.2. Segment Routing Header TLV

This I-D requests IANA to allocate a new entry in the "Segment Routing Header TLVs" sub-registry under the top-level registry "Internet Protocol Version 6 (IPv6) Parameters":

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA2-1</td>
<td>Path Tracing TLV</td>
<td>[This.ID]</td>
</tr>
</tbody>
</table>
13. Acknowledgements

The authors of this document would like to thank the team that has collaborated on the design and implementation of the Path Tracing framework at Cisco, Broadcom, Marvel, Swisscom, Alibaba, Softbank, University of Rome "Tor Vergata", and ETH Zurich. In particular: Eyal Dagan, Guy Caspary, Elad Naor, Aviran Kadosh, Eli Stein, Oren Yabo, Aviad Behar, Anand Sridharan, Anju Dey, John Bettink, Kamran Raza, Asif Islam, Yue Gao, Jakub Horn, Sam Kheirallah, Shelly Cadora, Kris Michielsen, Francois Clad, Stefano Salsano, Andrea Mayer, Paolo Lungaroni, Giulio Sidoretti, Leonardo Rodoni, Marco Tollini.

14. Contributors

Jisu Bhattacharya, Cisco Systems; jisu@cisco.com

Rakesh Gandhi, Cisco Systems; rgandhi@cisco.com

Serguei Bezverkhi, Cisco Systems; sbezverk@cisco.com

Sonia Ben Ayed, Cisco Systems; sbenayed@cisco.com

Israel Meilik, Broadcom; israel.meilik@broadcom.com

Shay Zadok, Broadcom; shay.zadok@broadcom.com

Weiqiang Cheng, China Mobile; chengweiqiang@chinamobile.com

15. References

15.1. Normative References


15.2. Informative References


Authors’ Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Belgium
Email: cf@cisco.com

Ahmed Abdelsalam (editor)
Cisco Systems, Inc.
Italy
Email: ahabdels@cisco.com

Pablo Camarillo Garvia (editor)
Cisco Systems, Inc.
Spain
Email: pcamaril@cisco.com

Mark Yufit
Broadcom
Israel
Email: mark.yufit@broadcom.com

Thomas Graf
Swisscom
Switzerland
Email: thomas.graf@swisscom.com

Yuanchao Su
Alibaba, Inc
China
Email: yitai.syc@alibaba-inc.com

Satoru Matsushima
SoftBank
Japan
Email: satoru.matsushima@g.softbank.co.jp
Abstract

This document uses the Multiprotocol Label Switching (MPLS) Entropy Label (EL) extensions defined in draft-decraene-mpls-slid-encoded-entropy-label-id or a new Special Purpose Label to indicate the presence of MPLS Extension Header (MEH) in an MPLS label stack. It defines different MPLS Extension Header encoding formats to carry additional data in the MPLS label stack that can influence forwarding decision and to carry additional data after the Bottom of the MPLS label stack.

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 3 September 2022.
Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction .................................................. 3
   1.1. Requirements .............................................. 4
2. Conventions Used in This Document ............................. 4
   2.1. Requirements Language ..................................... 4
   2.2. Terminology ................................................ 4
3. Overview ........................................................ 6
   3.1. Option 1 - ELC as MPLS Extension Header Indicator ....... 7
       3.1.1. Advantages with ELC ................................. 8
   3.2. Option 2 - New SPL as MPLS Extension Header Indicator ... 9
   3.3. Option 3 - NPL as MPLS Extension Header Indicator ....... 9
4. In-Stack MPLS Extension Header Encoding ....................... 10
5. Bottom Of Stack MPLS Extension Header Encoding ............... 12
6. MPLS Extension Header Encoding Example Use-case-1.a - Carrying
   FI without data in the MPLS label stack ..................... 14
7. MPLS Extension Header Encoding Example Use-case-1.b - Carrying
   FI with data in the MPLS label stack ......................... 15
8. MPLS Extension Header Encoding Example Use-case-2 - Carrying
   FI with data after the MPLS label stack ...................... 16
9. MPLS Extension Header Encoding Example Use-case-3 - Carrying
   use-case 1.a, 1.b and 2 in MPLS packet ....................... 17
10. Node Capability Signaling ...................................... 18
11. Security Considerations ....................................... 18
12. Backward Compatibility ....................................... 18
    12.1. Backward Compatibility With ELC as MEH Indicator .... 18
    12.2. Backward Compatibility With SPL as MEH Indicator .... 19
13. Processing In-Stack MPLS Extension Header .................... 19
14. Processing BOS MPLS Extension Header ........................ 20
15. IANA Considerations .......................................... 20
    15.1. IANA Considerations for Forwarding Instruction Flags .. 20
    15.2. IANA Considerations for IS-FI Opcode .................. 21
    15.3. IANA Considerations for BOS-FI Opcode ................ 21
    15.4. IANA Considerations for New Special Purpose Label .... 22
1. Introduction

[RFC3032] defines MPLS Header for carrying a stack of MPLS labels which are used to forward packets in an MPLS network. Today’s new applications require the MPLS packets to carry some additional indicators and associated ancillary data that would be used in MPLS packet forwarding decision or for OAM purpose.

Each application requires a separate Extended Special Purpose Label (eSPL) to address its problem that adds 2 extra labels (extension label 15 + eSPL) in the MPLS label stack. This approach does not scale, as it increases the label stack depth with multiple eSPLs that need to be imposed by the encapsulation node and scanned by the intermediate nodes. Also, currently there are no solutions defined to add ancillary data in a label stack or add multiple ancillary data after the Bottom Of Stack (BOS) in an MPLS packet. Ancillary data can be used to carry additional information, for example, a network slice identifier, In-Situ OAM (IOAM) data presence indicator, etc. Some of these use-cases are described in [I-D.saad-mpls-miad-usecases].

This document defines a new MPLS data plane extension header format to efficiently encode forwarding and OAM instructions those are easy to process in hardware. The instructions are encoded in the form of flags and opcodes and can be carried without associated ancillary data or with short in-stack ancillary data or with one or more ancillary data after the BOS.
MPLS Entropy Label (EL) standard is defined in [RFC6790]. This document uses the Entropy Label extensions defined in [I-D.decrane-mpls-slid-encoded-entropy-label-id] or a new Special Purpose Label (SPL) to indicate the presence of MPLS Extension Header (MEH) in an MPLS label stack. It defines different MPLS Extension Header encoding formats to carry additional data in the MPLS label stack that can influence forwarding decision and to carry additional data after the Bottom of the MPLS label stack.

1.1. Requirements

This document defines different MPLS Extension Header encoding formats to support the following requirements:

1. MPLS packet to carry additional data in the MPLS label stack to influence forwarding. This can be of two types:
   1a. Forwarding Instruction Flags (FIF) that does not use additional data.
   1b. Forwarding Instruction (FI) that needs additional data.

2. MPLS packet to carry additional data after the Bottom of the MPLS Label Stack.

3. Any combination of (1) and (2) in the same MPLS packet.

When MPLS Extension Header is added in an MPLS Label stack, the extension header MUST NOT contain the label field that can conflict with any previously allocated reserved label value. [I-D.bocci-mpls-miad-adi-requirements] describes additional requirements for MPLS Extension Header.

2. Conventions Used in This Document

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Terminology

BOS (Bottom Of Stack): Bottom of the MPLS label stack.

BOS-FI (Bottom Of Stack Forwarding Instruction): This is the Forwarding Instruction that is encoded after Bottom of MPLS Stack.
BPI (Bottom of the Stack MPLS Extension Header Presence Indicator): This is the flag to indicate the presence of MPLS Extension Header after the bottom of the MPLS label stack.

E2E (Edge-To-Edge): Edge to Edge.

EL (Entropy Label): Entropy Label defined as per [RFC6790].

ELC (Entropy Label Control): EL TTL field re-purposed to carry Entropy Label control bits defined in [I-D.decreaene-mpis-slid-encoded-entropy-label-id].

FI (Forwarding Instruction): Forwarding Instruction is the instruction that expresses the forwarding behaviour. This can result in changing the forwarding decision or adding some information or important data to the packet.

FIF (Forwarding Instruction Flags): A bitwise flag that influences the forwarding behaviour. This flag does not need any additional data to execute its FI.

FIOC (Forwarding Instruction Opcode): A Opcode value that refers to a specific Forwarding Instruction.

HBI (Hop-By-Hop Bottom of the Stack MPLS Extension Header Presence Indicator): This is the flag to indicate the presence of MPLS Extension Header after the bottom of the MPLS label stack that require Hop-By-Hop processing.

IS-FI (In-Stack Forwarding Instruction): This is the Forwarding Instruction that is encoded in the MPLS label stack.

IPI (In-Stack MPLS Extension Header Presence Indicator): This is the flag to indicate the presence of MPLS Extension Header in the MPLS label stack.

MEI (MPLS Extension Indicator): This is the Indicator MPLS Label which indicates the presence of MPLS Extension Header in the MPLS Label stack.

MEH (MPLS Extension Header): MPLS Extension Header encoding carried in the MPLS Label stack.

MPLS (Multiprotocol Label Switching): Multiprotocol Label Switching.

NPL (Network Programming Label): Network Programming Label provisioned by user.
SPI (Slice ID Presence Indicator): This is the flag to indicate the presence of Slice ID in the Entropy Label field.

SPL (Special Purpose Label): IANA Allocated Special Purpose Label in the range of 0-15. Extended Special Purpose Label (eSPL) uses label value 15.

TC (Traffic Class): Traffic Class.

TTL (Time-To-Live): Time To Live.

3. Overview

Extending existing MPLS Header needs two main parts.

* MPLS Extension Header Indicator (MEI) - This is a way to indicate the presence of MPLS Extension Header in the packet. This could be done using two different methods. Each method has its own advantages and disadvantages. This document describes both options of MEI. The encoding formats defined in this document are compatible with both options of MEI.

  Option 1. MEI by extending ELI/EL

  Option 2. MEI by using a New Special Purpose Label (SPL) allocated by IANA

  Option 3. MEI by using New Network Programming Label (NPL) provisioned by user

* MPLS Extension Header Format - The format in which the MPLS Extension Header could be carried in the MPLS packet. This includes both In-stack Extension Header and BOS Extension Header.

    0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
    ++++++++++++++++++++++++++++++++++++++++++++++++++
    | Label                    | TC  |S|      TTL      |
    ++++++++++++++++++++++++++++++++++++++++++++++++++

    Figure 1: MPLS Label Format

New In-Stack (IS) MPLS Extension Header format is defined in this document to carry the In-stack Forwarding Instruction and corresponding data in the MPLS label stack.

* It uses MPLS Label field to carry the Forwarding Instruction Opcode.
* It uses Traffic Class (TC) field to identify the Length of the MPLS In-Stack Extension Header size.

* It uses MPLS Label and Time-To-Live (TTL) fields to carry the In-Stack data (can be Flags or data).

A new Bottom Of Stack (BOS) MPLS Extension Header format is defined in this document to carry the BOS Forwarding Instruction and corresponding data after the MPLS Label stack.

The MPLS Extension Header encoding formats defined in this document are flexible and allow to stack multiple In-Stack and BOS MPLS Extension Headers in a desired order in the same MPLS packet.

3.1. Option 1 - ELC as MPLS Extension Header Indicator

As described in [I-D.decraene-mpls-slid-encoded-entropy-label-id], the EL’s 8-bit TTL field is re-purposed as Entropy Label Control (ELC) field. One bit from ELC is requested for the Slice ID Presence Indicator (SPI) and the 7 bits are available for use. From the ELC, 3 bits (for IPI, BPI and HBI) are allocated to indicate the presence of MPLS Extension Header.

TTL (in ELC) bit allocations are defined by user as follows:
### Table 1: Bit Fields

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD0</td>
<td>SPI - Slice ID Presence Indicator: Indicate the presence of Slice ID in the Entropy label as defined in [I-D.decraene-mpls-slid-encoded-entropy-label-id].</td>
</tr>
<tr>
<td>TBD1</td>
<td>IPI - In-Stack Extension Header Presence Indicator: Indicate the presence of In-Stack MPLS Extension Header after this label.</td>
</tr>
<tr>
<td>TBD2</td>
<td>BPI - Bottom Of Stack Extension Header Presence Indicator: Indicates the presence of MPLS Extension Header after the Bottom Of Stack (BOS).</td>
</tr>
<tr>
<td>TBD3</td>
<td>HBI - Hop-By-Hop Bottom Of Stack Extension Header Indicator: Indicates the MPLS Extension Header after the Bottom Of Stack requires Hop-By-Hop processing.</td>
</tr>
<tr>
<td>TBD4 - TBD7</td>
<td>Unassigned Bits.</td>
</tr>
</tbody>
</table>

IL - In-Stack Extension Header Length - The 3-bit TC field in the EL is used to indicate the length of the In-Stack MPLS Extension Header (excluding the ELI and EL labels) in terms of number of 32-bit labels. If more than 7 labels are needed in an MPLS extension header, the node can either use a BOS extension header to carry the data or use an additional In-stack MPLS Extension Header with MEI Label.

For backwards compatibility, an intermediate and decapsulating nodes MUST only read the length from the TC field when the IPI (In-Stack Extension Presence Indicator) is set to "1".

#### 3.1.1. Advantages with ELC

Faster deployment in an existing network that has EL already deployed with an incremental benefit (e.g., incremental signaling extension for ELI capability).

Single label for Entropy in the MPLS header which helps with keeping label stack size smaller.
When EL is already enabled in the network, the proposed scheme does not require hardware to support an additional SPL indicator.

Save a new Special Purpose Label and related protocol extensions to signal its capability in LDP, RSVP-TE, BGP, IS-IS, OSPF, BGP-LS, etc.

An intermediate node can compute ECMP hash with the EL field and avoid inconsistent load-balancing of traffic flow that can happen when MPLS Extension Header alters the label stack.

Reduce MPLS Label stack size when EL is enabled for ECMP hashing when MPLS Extension Header is also used. As there is only one field for EL in the MPLS Header, it simplifies the MPLS header processing.

3.2. Option 2 - New SPL as MPLS Extension Header Indicator

The MPLS Extension Header encoding formats defined in this document is equally applicable when using a new Special Purpose Label (SPL) (value TBA1) allocated by IANA.

```
+-----------------------------------------------+
|     MEI=SPL (TBA1)                    | IL  |S| IPI,BPI,HBI   |
+-----------------------------------------------+
```

Figure 3: New SPL as MPLS Extension Header Indicator

The TTL field in the SPL (value TBA1) is used to encode FI Flags including IPI, HBI and BPI flags defined in this document. The definition and meaning of these flags and IL field are exactly the same as those in ELC field.

3.3. Option 3 - NPL as MPLS Extension Header Indicator

The MPLS Extension Header encoding formats defined in this document is equally applicable when using a Network Programming Label (NPL) configured by an operator.

```
+-----------------------------------------------+
|     MEI=NPL                                   | IL  |S| IPI,BPI,HBI   |
+-----------------------------------------------+
```

Figure 4: NPL as MPLS Extension Header Indicator
The TTL field in the NPL is used to encode FI Flags including IPI, HBI and BPI flags defined in this document. The definition and meaning of these flags and IL field are exactly the same as those in ELC field.

4. In-Stack MPLS Extension Header Encoding

This section describes the encoding format of the MPLS Extension Header carried as part of the MPLS label stack. The encoding format defined is flexible (e.g., stackable opcodes in desired order), extensible (by defining new Opcodes) and ASIC friendly (by using Extension Header Length, Opcode+Data in the same field).

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL      | IL=1|S| IPI=1         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  IS-FI Opcode |    In-Stack Data      |R|D|E|S| In-Stack Data |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: In-Stack Extension Header Format

IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

Since In-Stack MPLS Extension Header is present as part of the MPLS Label stack, the 32-bit MPLS Label is redefined to encode the MPLS Extension Header as follows:

Label Field:

The first 8 bits are used to define the In-Stack Forwarding Instruction (IS-FI) Opcode. Next 12 bits in the Label field and the 8 bits from the TTL field are used to carry In-Stack data corresponding to the IS-FI opcode. This opcode ranges from 1 to 255. IS-FI Opcode value of 0 is marked as invalid to avoid the label value aliasing with the reserved SPLs.

* IS-FI Opcode Value:1 - IANA Allocated to carry the Forwarding Instruction Flags (FIF).

* IS-FI Opcode Value:2 - IANA Allocated to indicate the offset in terms of number of bytes for the start of the BOS data after the MPLS Label Bottom of the Stack. This can allow to carry Generic Control Word (0000b) [RFC4385] and G-ACh (0001b) [RFC5586] fields immediately after the BOS. Adding of this opcode is not required when the BOS data starts immediately after the Bottom of the Label Stack (i.e. when offset is 0).
* IS-FI Opcode Value: 3-254 - MUST be assigned by IANA.

* IS-FI Opcode Value: 255 - IANA Allocated for IS-FI Opcode range extension. This gives the extensibility for opcode range beyond 255.

IS-FI Opcode MUST define the following procedure before it can be used:

1. Define the Data format encoded in the MPLS extension header.

2. Define the Hop-By-Hop or Edge-To-Edge (only on the decapsulation node) processing scope.

3. The Hop-By-Hop IS-FI opcodes MUST be placed before the Edge-To-Edge IS-FI Opcodes in the MPLS Extension Header of the packet to optimize the Hop-By-Hop processing in hardware.

TC Field:

This field is used to indicate the MPLS Extension Header stacking and In-Stack Data stacking.

E (E2E-Bit): MPLS Extension Header In-Stack Data requires E2E processing. If this is set to "1", then this 4-byte MPLS Extension Header requires Edge-To-Edge processing. If this is set to 0, then this 4-byte MPLS Extension Header requires Hop-By-Hop processing. Note that E2E-Bit is not used with the Entropy Label TC field.

D (DS-Bit): Data Stacking Bit. This is used to encode more than 20 bits of data for this IS-FI Opcode. If this is set to "1", then this is the end of the data for the IS-FI Opcode.

R (Reserved Bit): MUST be set to "0" on transmit and ignored on receive.

TTL Field:

This 8-bit field is used to carry In-Stack data apart from the 12 bits in the Label field.

NOTE:
An intermediate node may use the full MPLS label stack for ECMP hash computation hence the In-Stack MPLS extension header MUST NOT change the Label Field part of the IS-FI data within the same traffic flow. But the TTL part of IS-FI data can change for the same traffic flow without affecting the ECMP hash. The In-Stack Extension Header encoding defined above ensures this.

5. Bottom Of Stack MPLS Extension Header Encoding

This section describes the encoding format of the MPLS Extension Header which is present after the bottom of the MPLS label stack.

BPI flag is set to "1" to indicate the presence of MPLS Extension Header after the bottom of MPLS label stack.

HBI flag is set to "1" to indicate the MPLS Extension Header after the Bottom Of Stack that requires Hop-By-Hop processing.

A new generic 4-byte header is defined to carry the information about the Forwarding Instruction and its corresponding data that is carried after the bottom of label stack. This generic header is added to each Forwarding Instruction that is encoded after the MPLS bottom of the stack. This generic header gives the flexibility to add multiple Forwarding Instruction after the BOS in any desired order.

The 4-byte BOS Extension Header is described below:
<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 3</td>
<td>This 4-bit nibble MUST be set to &quot;0010b&quot;. This is to avoid aliasing with an IPv4/IPv6 header.</td>
</tr>
<tr>
<td>4 - 7</td>
<td>This 4-bit nibble defines the version of the generic header format. The current version value is &quot;0&quot;.</td>
</tr>
<tr>
<td>8 - 15</td>
<td>This 8-bit field indicates the BOS FI Opcode value. This opcode values will be allocated by IANA.</td>
</tr>
<tr>
<td>16 - 23</td>
<td>This 8-bit field indicates the length of the data encoded in units of 4 bytes excluding the current header.</td>
</tr>
<tr>
<td>24 - 31</td>
<td>This 8-bit field carries the BOS-Flags. 0 - NH bit (Next-Header Presence Bit): Indicates the presence of next BOS extension header. 1 - H bit (Hop-By-Hop Bit): Hop-By-Hop processing is required for this Bottom Of Stack data. 7 - 2 bits: Unassigned bits.</td>
</tr>
</tbody>
</table>

Table 2: BOS MPLS Extension Header Format

BOS-FI Opcode value of 0 is marked as invalid.

BOS-FI Opcode Value: 0 - NH bit (Next-Header Presence Bit): Indicates the presence of next BOS extension header. 1 - H bit (Hop-By-Hop Bit): Hop-By-Hop processing is required for this Bottom Of Stack data. 7 - 2 bits: Unassigned bits.

BOS-FI Opcode Value: 1-254 - MUST be assigned by IANA.

BOS-FI Opcode Value: 255 - IANA Allocated for BOS-FI Opcode range extension. This gives the extensibility for opcode range beyond 255.

If an application requires to add its own data TLV, then the TLV can be added as part of BOS-Data.

BOS-FI Opcode MUST define the following procedure before it can be used:

1. Define the Data format encoded in the MPLS extension header.
2. Define the Hop-By-Hop or Edge-To-Edge (only on the decapsulation node) processing scope.
3. The Hop-By-Hop BOS-FI opcodes MUST be placed before the Edge-To-Edge BOS-FI Opcodes in the MPLS Extension Header of the packet to optimize the Hop-By-Hop processing in hardware.
6. MPLS Extension Header Encoding Example Use-case-1.a - Carrying FI without data in the MPLS label stack

The TTL field can support only up to 8-bit flags. This is the use-case to extend the TTL flags and carry additional Forwarding Instruction Flags (FIF) in the MPLS label stack. These forwarding instructions do not require any additional data to be carried with this FI.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Entropy Label or SPL or NPL | IL=1|0| IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IS-FI Opcode=1 | Flags                |R|1|E|1|   Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Example In-Stack Extension Header Carrying Forwarding Instruction Flags

- IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

**Label Field:**

- In this case the FI opcode value is set to "1". FI Opcode value "1" is reserved for extending the TTL flags. This indicates the presence of additional flags in the Label field and TTL fields

**TC Field:**

- DS-Bit - This bit is set to "1" to indicate that the flags are not extended further.

**TTL Field:**

- 8-bit field is used to encode the Forwarding Instruction Flags apart from 12 bits Label field.

The FIF bit position and its meaning MUST be defined by IANA.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Entropy Label or SPL or NPL | IL=2|0| IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|IS-FI Opcode=1 | Flags                |R|1|E|1|   Flags       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

More than 20 bits of data can be encoded as part of IS-FI opcode. In this specific case, the FI flags which are more than 20 bits are encoded in next 4 bytes of the MPLS header.

While encoding the additional data, the Most Significant bit of the Label Field MUST be set to "1" to prevent from aliasing with the reserved SPLs in the case of legacy devices.

7. MPLS Extension Header Encoding Example Use-case-1.b - Carrying FI with data in the MPLS label stack

This is the use-case where the MPLS Label stack to carry the Forwarding Instruction with a corresponding data.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Entropy Label or SPL or NPL      | IL=1|0|   IPI=1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode  |        Data           |R|1|E|1|   Data        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

Label Field:

First 8 bits encodes the In-Stack forwarding opcode. In this case the FI opcode value ranges from 1 to 254. This value is assigned by IANA. This opcode value defines data format carried in the Label field and the TTL field.

TC Field:

DS-Bit - This bit is set to "1" to indicate that the data is encoded in the 19-bit Label field and does not exceed 19 bits.

R-Bit - Reserved bit and MUST be set to "0" on transmit and ignored when received.

TTL Field:

8-bit field is used to encode the In-Stack data apart from 12-bit Label field.
More than 20 bits of data can be encoded as part of IS-FI opcode. In this specific case, the In-Stack data which are more than 20 bits are encoded in next 4 bytes of the MPLS header.

While encoding the additional data, the Most Significant bit of the Label Field MUST be set to "1" to prevent from aliasing with the reserved SPLs in the case of legacy devices.

8. MPLS Extension Header Encoding Example Use-case-2 - Carrying FI with data after the MPLS label stack

This is the use-case where the Forwarding Instruction with a corresponding data is carried after the MPLS bottom of label stack.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|      Entropy Label or SPL or NPL                   | TC  |1| BPI=1,HBI=1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|0 0 1 0|Reserve|BOS-FI Opcode=1| Length=1(word)|Flags NH=1,H=1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                        BOS-Data1                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|0 0 1 0|Reserve|BOS-FI Opcode=2| Length=2(word)|Flags NH=0,H=0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                        BOS-Data2                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                        BOS-Data2                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                        Payload                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 11: Example BOS Extension Header Carrying FI with data

BPI flag is set to "1" to indicate the presence of BOS MPLS Extension Header. Also, HBI flag is set to 1 to indicate the presence of BOS MPLS Extension Header that requires Hop-By-Hop processing.
In this case, the MPLS packet is encoding two different types of BOS FI (Opcode 1 and Opcode 2) after the bottom of MPLS label stack.

The first BOS MPLS Extension Header has the Length value as "1", this indicates that the data corresponding to this FI opcode "Type1" is 4 bytes following this header. Also the Next-Header (NH) flag in BOS-Flags is set to "0x1", this indicates the presence of next BOS MPLS Extension Header. The H flag is set to "0x1" that indicates the Hop-By-Hop processing is required.

The second BOS MPLS Extension Header has the Length value as "2", this indicates that the data corresponding to the FI opcode "Type2" is 8 bytes following this header. In this case the Next-Header flag in BOS-Flags is set to "0x0", this indicates that this is the last BOS MPLS Extension Header encoded. The H flag is set to "0x0" that indicates the Hop-By-Hop processing is not required.

9. MPLS Extension Header Encoding Example Use-case-3 - Carrying use-case 1.a, 1.b and 2 in MPLS packet

This is the use-case where the same MPLS packet carries the use-cases "1.a", "1.b" and "2".

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Entropy Label or SPL or NPL | IL=3|0| IPI=BPI=HBI=1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode=1| Flags |R|0|E|0| Flags |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode=2| 0          |R|1|E|0| Offset = 1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IS-FI Opcode=3| Data       |R|1|E|1| Data     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 0 1 0|Reserve|BOS-FI Opcode=1| Length=1(word)|Flags NH=1,H=1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BOS-Data1 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 0 1 0|Reserve|BOS-FI Opcode=2| Length=2(word)|Flags NH=0,H=0 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BOS-Data2 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| BOS-Data2 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
Figure 12: MPLS Packet Carrying 1.a, 1.b and 2 Use-cases

IPI and BPI flags are set to "1" to indicate the presence of both In-Stack and BOS MPLS Extension Header as mentioned in the above use-cases. IS-FI Opcode 2 is added to indicate the offset of 1 word after the MPLS header BOS and start of the BOS Extension Header.

10. Node Capability Signaling

The node capability for the MPLS Extension Header must be signaled before the MPLS Encapsulating node can add the necessary MPLS Extension Header in the MPLS label stack. The capability signaling will be added in LDP, RSVP-TE, BGP, IS-IS, OSPF, BGP-LS, etc. This is outside the scope of this document.

11. Security Considerations

The security considerations in [RFC3032] also apply to the extensions defined in this document. The MPLS Extension header MUST NOT be exposed to the node which does not support the new MPLS Extension Header.

12. Backward Compatibility

12.1. Backward Compatibility With ELC as MEH Indicator

As specified in [RFC6790], the TTL field of the EL MUST be "0". On the Node which is capable of processing the MPLS Extension Header when it finds that this TTL value is non-zero, then only it will start processing the MPLS Extension header.

In addition, the TC field will be interpreted as the In-Stack MPLS Header Extension Length only when the TTL field’s IPI Flag is set to "1".

For the legacy node that does not advertise the MPLS Extension Header capability, the Encapsulating node MUST make sure that the MPLS Extension header is not at the top of the MPLS label stack to avoid misforwarding the packets by misinterpreting In-Stack Extension Header as a label.

The MPLS Extension Header Encoding format is designed to make sure that it does not alias with any reserved SPL.

The MPLS extension does not affect the existing GAL / G-ACh [RFC5586] based encoding of data in the MPLS packet. This MPLS extension can co-exist with the existing GAL / G-ACh based encoding of data.
12.2. Backward Compatibility With SPL as MEH Indicator

For the legacy node that does not advertise the MPLS Extension Header capability, the Encapsulating node MUST make sure that the MPLS Extension header is not at the top of the MPLS label stack to avoid dropping the packets.

The MPLS Extension Header Encoding format is designed to make sure that it does not alias with any reserved SPL.

The MPLS extension does not affect the existing GAL / G-ACh [RFC5586] based encoding of data in the MPLS packet. This MPLS extension can co-exist with the existing GAL / G-ACh based encoding of data.

13. Processing In-Stack MPLS Extension Header

Encapsulating Node:
* MUST NOT add In-Stack MPLS Extension header if the decapsulation node is not capable of In-Stack MPLS Extension header.
* SHOULD NOT change the IS-FI Opcode and the first 12 bits of the In-Stack Data for the same packet flow avoid ECMP path change.
* MAY change In-Stack data part present only in the TTL field for the same packet flow.
* MUST ensure that the penultimate node does not remove the MPLS extension header.

Intermediate Node:
* MUST ignore the IS-FI Opcode that are not supported.
* MUST NOT add In-Stack MPLS Extension header if the decapsulation node is not capable of In-Stack MPLS Extension header.
* SHOULD NOT change the IS-FI Opcode and the first 12 bits of the In-Stack Data for the same packet flow.
* MAY change In-Stack data part present only in the TTL field for the same packet flow.
* MAY remove the IS-FI opcode and its corresponding data for all matching packet flow.

Decapsulating Node:
* MUST remove the In-Stack MPLS Extension header.

14. Processing BOS MPLS Extension Header

Encapsulating Node:

* MUST NOT add BOS MPLS Extension header if the decapsulation node is not capable of BOS MPLS Extension header.

* MUST ensure that the penultimate node does not remove the MPLS extension header.

Intermediate Node:

* MAY add additional data to the existing BOS-FI encoded.

* MAY add a new BOS-FI and its corresponding data if the decapsulation node supports BOS MPLS Extension header.

Decapsulating Node:

* MUST remove the BOS MPLS Extension header.

15. IANA Considerations

Below are the IANA actions which this document is requesting.

15.1. IANA Considerations for Forwarding Instruction Flags

IANA is requested to create a new registry to assign the bit position and the meaning to the Forwarding Instruction Flags based on the user request.

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-0</td>
<td>Unassigned</td>
<td>This document</td>
</tr>
</tbody>
</table>

Table 3: Forwarding Instruction Flags Registry
15.2. IANA Considerations for IS-FI Opcode

IANA is requested to create a new registry to assign IS-FIOC opcode values. All code-points in the range 1 through 175 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 176 through 239 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code-points are allocated according to Table 4:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 175</td>
<td>IETF Review</td>
<td>This document</td>
</tr>
<tr>
<td>176 - 239</td>
<td>First Come First Served</td>
<td>This document</td>
</tr>
<tr>
<td>240 - 251</td>
<td>Experimental Use</td>
<td>This document</td>
</tr>
<tr>
<td>252 - 254</td>
<td>Private Use</td>
<td>This document</td>
</tr>
</tbody>
</table>

Table 4: In-Stack Forwarding Instruction Opcode Registry

Following IS-FIOC Opcode values are assigned from this registry.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid value</td>
<td>This document</td>
</tr>
<tr>
<td>1</td>
<td>Forwarding Instruction Flags</td>
<td>This document</td>
</tr>
<tr>
<td>2</td>
<td>Offset of start of Bottom Of Stack Data after BOS Label</td>
<td>This document</td>
</tr>
<tr>
<td>255</td>
<td>Opcode Range Extension Beyond 255</td>
<td>This document</td>
</tr>
</tbody>
</table>

Table 5: In-Stack Forwarding Instruction Opcode Values

15.3. IANA Considerations for BOS-FI Opcode

IANA is requested to create a new registry to assign BOS-FIOC opcode values.
+-----------------------------------------------+------------------+
| Value | Description           | Reference |
+-----------------------------------------------+------------------+
| 1 - 175 | IETF Review         | This document |
| 176 - 239 | First Come First Served | This document |
| 240 - 251 | Experimental Use | This document |
| 252 - 254 | Private Use         | This document |
+-----------------------------------------------+------------------+

Table 6: Bottom-Of-Stack Forwarding Instruction Opcode Registry

Following BOS-FIOC Opcode values are assigned from this registry.

+-----------------------------------------------+------------------+
| Value | Description           | Reference |
+-----------------------------------------------+------------------+
| 0 | Invalid value         | This document |
| 255 | Opcode Range Extension Beyond 255 | This document |
+-----------------------------------------------+------------------+

Table 7: Bottom-Of-Stack Forwarding Instruction Opcode Values

The application that requires an Opcode for the Forwarding Instruction (IS-FIOC or BOS-FIOC) or a Flag must request the code-point and its meaning from IANA.

15.4. IANA Considerations for New Special Purpose Label

IANA is requested to allocate a value TBA1 for the MEI SPL label from the "Base Special-Purpose MPLS Label Values" registry to indicate the presence of MPLS Header Extension.

16. Appendix

16.1. Alternate approach for In-Stack Extension Header Encoding

In the above In-Stack Extension Header Encoding the Label field is used to encode the FI Opcode. So just for completeness, here is the alternate way of In-Stack Extension Header Encoding is provided.
Figure 13: Alternate In-Stack Extension Header Format

IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

Since In-Stack MPLS Extension Header is present as part of the MPLS Header, the MPLS Header is redefined to encode the MPLS Extension Header.

Label Field:

Most significant bit is always set to "1" to avoid aliasing with the reserved SPLs.

Rest of the 19 bits and the "R" bit from the TC bit can be used by the application. So total of 20 bits can be used to carry the data corresponding to IS-FI opcode.

TC Field:

This carries data stacking bits. They are as follows:

D (DS-Bit): Data Stacking Bit. This is used to encode more than 19 bits of extended data in the MPLS Label stack. If this is set to "1", then this is the end of extended data.

R (Reserved Bit): This is used to encode the IS-FI data.

TTL Field:

This carries In-Stack Forwarding Instruction opcode.

16.2.  MPLS Extension Header Example for Entropy Label using New SPL

The MPLS Extension Header encoding formats defined in this document is applicable when using a new Special Purpose Label (SPL) or using a Network Programming Label (NPL) configured by an operator.

The TTL field in the SPL (value TBA1) is used to encode FI Flags including IPI, HBI and BPI flags defined in this document.
16.3. MPLS BOS Extension Header Example with IOAM Data Fields

The Bottom Of Stack (BOS) Extension Header is used with BOS Opcode for IOAM.

Bottom Of Stack Presence Indicator (BPI) flag in TTL is set to "1" to indicate the presence of BOS Extension Header. HBI flag in TTL is set to "1" to indicate the BOS Extension Header requires Hop-By-Hop processing.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-------------------------------+-------------------------------+
| MEI=SPL (value TBA1)          | IL=1|S|   IPI=1       |
+-------------------------------+-------------------------------+
| IS-FI Opcode=3|  Entropy Label        |R|D|E|S|   SLID        |
+-------------------------------+-------------------------------+
```

Figure 14: MPLS Extension Header Encoding Example for Entropy Label using New SPL

The FI Opcode value 3 as an example indicates encoding of Entropy Label and Slice ID as shown in the above Figure.

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-------------------------------+-------------------------------+
| Entropy Label or SPL or NPL   | TC |1| BPI=1, HBI    |
+-------------------------------+-------------------------------+
| 0 0 1 0|Reserve|BOS Opcode=IOAM|Length (words) | Flags (NH, H) |
+-------------------------------+-------------------------------+
| IOAM-OPT-Type | IOAM HDR Len | Block Number | Reserved       |
+-------------------------------+-------------------------------+
```

Figure 15: Example MPLS Encapsulation for IOAM using BOS Extension Header
17. References

17.1. Normative References


17.2. Informative References


[I-D.bocci-mpls-miad-adi-requirements]


Acknowledgments

The authors of this document would like to thank the MPLS Working Group Design Team for the discussions and comments on this document.

Contributors

TBD

Authors’ Addresses

Jaganbabu Rajamanickam (editor)
Cisco Systems, Inc.
Canada
Email: jrajaman@cisco.com

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

Jisu Bhattacharya
Cisco Systems, Inc.
Email: jisu@cisco.com

Bruno Decraene
Orange
Email: bruno.decraene@orange.com
Roi Zigler
Broadcom
Email: royi.zigler@broadcom.com

Weiqiang Cheng
China Mobile
Email: chengweiqiang@chinamobile.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
Use Cases for SR Policy Group

draft-jiang-spring-sr-policy-group-use-cases-00

Abstract

Segment Routing is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node. An SR Policy is associated with one or more candidate paths, and each candidate path is either dynamic, explicit or composite. This document illustrates some use cases for SR policy group composite candidate path in MPLS and IPv6 environment.

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 September 7, 2022.

Copyright Notice

Copyright (c) 2022 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
1. Introduction

Segment routing (SR) [RFC8402] is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node. The ingress node steers packets into a specific path according to the Segment Routing policy (SR Policy) as defined in [I-D.ietf-spring-segment-routing-policy]. In order to distribute SR policies to the headend, [I-D.ietf-idr-segment-routing-te-policy] specifies a mechanism by using BGP.

An SR policy is associated with one or more candidate paths. A composite candidate path acts as a container for grouping SR policies. As described in section 2.2 in [I-D.ietf-spring-segment-routing-policy], the composite candidate path construct enables combination of SR policies, each with explicit candidate paths and/or dynamic candidate paths with potentially different optimization objectives and constraints, for load-balanced steering of packet flows over its constituent SR policies. For service-class based steering, and in the best-effort forwarding scenario when SR policy becomes unavailable, packets are also forwarded over the constituent SR policies of composite candidate path.
This document illustrates some use cases for SR policy group composite candidate path in MPLS and IPv6 environment.

2. Terminology

The definitions of the basic terms are identical to those found in Alternate Marking [RFC8402].

The important new terms that need to be explained are listed below:

SR policy group: An SR policy which contains a group of constituent SR policies. An SR policy group represents a composite candidate path.

ODN: On-demand Next-Hop.

ODN SR policy: Preconfigure an ODN template specified color. When the device receives a BGP route, if the color extended attribute value of the BGP route is the same as the color value of an ODN template, the device can automatically create an SR policy.

ODN SR policy group: An SR policy group dynamically created through ODN.

3. SR Policy Group

An SR policy group is specified as a group of its constituent SR policies. It is valid when it has at least one valid constituent SR policy.

As defined in [I-D.ietf-spring-segment-routing-policy], The endpoints of the constituent SR policies and the parent SR policy MUST be identical, and the colors of each of the constituent SR policies and the parent SR policy MUST be different. SR policy group and its constituent SR policies follow the same criteria:

- The endpoints of the constituent SR policies and its SR policy group MUST be identical.
- The colors of each of the constituent SR policies and its SR policy group MUST be different.
- The constituent SR policies MUST NOT contain SR policy groups.

As a special SR policy, SR policy group also has color attribute, which is identified by <color, endpoint> on the headend.
An SR policy can be generated by static configuration or a centralized controller distribution, also can be generated based on the on-demand SR policy optimization template dynamically.

4. Steering into An SR Policy Group

A headend can steer a packet flow into a valid SR policy group in various ways:

- **Per-flow Steering**: Specify the mapping relationship between color and flow characteristics (such as DSCP) for SR policy group, and create a policy group that binds a destination IP address to the SR policy group. Upon receiving a packet with the specified destination address, the device searches for the SR policy containing the color value mapped to the flow characteristics of the packet in the SR policy group. The device will use the matching SR policy to forward the packet.

  The device obtains an SR policy group for traffic steering as follows:

  * Matches the destination IP/IPv6 address in a packet with an SR policy group.
  
  * Searches for an SR policy group with color and endpoint address matching the color extended community attribute and next hop in a BGP route, and recurses the BGP route to the SR policy group.

  The Ingress node can match flow characteristics in its ingress interfaces (upon any field such as Ethernet destination/source/VLAN/TOS or IP destination/source/DSCP or transport ports or application attribute etc.) and color them with an internal per-packet forwarding-class variable. According to the forwarding-class variable the ingress node selects the matching SR policy in the SR policy group.

- **Policy-based Steering**: incoming packets match a routing policy that directs them on an SR policy group. Parse the flow characteristics (such as DSCP/802.1p value) from the packet header, find its corresponding color, and then match it to an SR policy in the SR policy group, forward the incoming packets through the matched SR policy.

  If an SR policy group has at least one valid constituent SR policy of specified color, flow load-balance steer over its valid constituent SR policies with the same color. When all constituent SR policies of specified color are invalid, packets are forwarded based on a default SR policy preconfigured.
5. On-demand SR Policy Group

SR policies are generally generated by manual static configuration or distributed by centralized controller. Manual configuration may be troublesome, especially when many SR policies need to be configured. The controller mode may also not be suitable for operators who need to make full use of distributed intelligence.

In scenarios that distinguish service forwarding paths based on DSCP value and 802.1p priority, SR policy groups can be automatically created through ODN to establish the dynamic mapping between service types and SR policy groups, which can greatly reduce the workload of configuration.

Create the ODN template of SR policy group in the headend. When the device receives a BGP route, if the color extended community attribute carried by the BGP route is the same as the color value of the ODN template, the next hop address of the BGP route is used as the destination endpoint address of the SR policy group, and the color value of the ODN template is used as the color attribute of the SR policy group to generate an SR policy group.

After the SR policy group is created by ODN, its constituent SR policy is usually generated by ODN. ODN SR policy dynamically generates candidate paths through affinity attributes, flex algo algorithm or PCE calculation.

6. SR Policy Group Use Cases

The use cases described in this section do not constitute an exhaustive list of all the possible scenarios: this section only includes some of the most common envisioned deployment models for SR policy group.

6.1. SR Policy Group in L3VPN over TE Scenarios

In Figure 1, CE1 and CE2 belong to the same L3VPN and access the public network through PE1 and PE2 respectively. There are many kinds of traffic between CE1 and CE2. When the ordinary traffic is too large, the forwarding of important traffic will be affected.

In order to ensure the forwarding quality of important services, the steering based on Forwarding class can be configured using SR policy group. After the steering based on forwarding class is configured, the traffic of different service levels will be carried by the specified SR policy tunnel, which can effectively ensure the forwarding quality of important services with high service levels.
It is assumed that in this network, the policy group contains three constituent policies: Policy-A, Policy-B and Policy-C. Services with different forwarding class will carry different DSCP values in the packet. Identify the customer’s service through DSCP on PE1. The voice traffic of VIP customers is forwarded according to the path of low-delay Policy-A, other traffic of VIP customers is forwarded according to the path of Policy-B, and all businesses of non VIP customers are carried by Policy-C.

6.2. SR Policy Group in Cloud Backbone Acceleration Scenarios

As shown in Figure 2, multiple cloud data centers are interconnected through cloud backbone networks. In the public cloud, there are different SLA requirements for different service types, such as voice service and cloud disk. Deploy a static SR policy group on the core of the cloud backbone network to prevent network congestion. There are multiple SR policies in the SR policy group.

In order to ensure the service quality of different types of services, the service types are distinguished by flow classification, then different services are mapped to different DSCP value, and finally the traffic of different DSCP is imported into different SR policies.
Through the SR policy group, different forwarding paths can be introduced based on the DSCP value in the IP/IPv6 packet header.

First, create an SR policy group and assign color identification to the SR policy group.

Then, configure multiple SR policies into one SR policy group in the headend, specify the mapping relationship between each SR policy and DSCP value in the SR policy group, and then bind the service type to the specified SR policy group.

In this way, when the headend receives traffic, it first matches to the SR policy group according to the next hop and color of the route, and then finds the mapped SR policy in the corresponding group according to the DSCP value carried in the IP/IPv6 packet header.

DSCP based steering is suitable for differentiating services at the source and specifying different DSCP value scenarios.

6.3. SR Policy Group in the L2VPN Network Scenarios

Similar to the DSCP-based steering scenario, in the layer 2 access network and L2VPN network, the service types are distinguished by the 802.1p priority in the packet header, and the 802.1p priority is mapped to color in the SR policy group. Different services can be forwarded into different paths.
As shown in Figure 3, CE1 and CE2 belong to the same VPLS and are connected to the MPLS backbone network through PE1 and PE2 respectively. Establish two MPLS-SR policy tunnels Policy-A and Policy-B between PE1 and PE2 to carry this VPLS service. Policy-A and Policy-B are the constituent policies of SR policy group. Two SR policy tunnels correspond to two different priorities. The VPLS access end classifies the traffic flow, trusts the priority of 802.1p, and introduces the services of VPLS leased line users and non-leased line users into different SR policy according to different priorities.

### 6.4. SR Policy Group in the Application-aware Scenarios

By carrying the application attribute (including APP ID and APP parameters) through data packets, i.e., the delivery of application-aware information and ensuring the security and reliability of application-aware information, the network senses the application groups’ requirements and provides high-quality differentiated services according to the demand of the applications. And when the network transmits the data packets, it matches the SR policy according to the application attribute in the data packets and selects the corresponding path of constituent SRv6 policy to transmit the data packets (e.g., low latency path) to meet the SLA requirements and service chain in order to improve the service quality.

As shown in Figure 4 below, the policy group contains three constituent SR policies: Policy-A, Policy-B and Policy-C. The data packets of APP1 are forwarded by Policy-A, the data packets of APP2 are forwarded by Policy-B, and the data packets of APP3 are forwarded by Policy-C.
6.5. Application of ODN SR Policy Group in Trusted Network Scenarios

Section 3 of [I-D.lin-opsec-trustroute-problem-statement] introduces the use case of trusted network. By dynamically creating SR policy through ODN, automatic steering traffic according to security level can be realized.

From the perspective of security and trustworthiness, the security levels for users with different security requirements and the trustworthiness levels of the network transmission devices can be determined according to their performance and reliability. Different forwarding paths are provided for packets with different security levels.
As shown in Figure 5, the trustworthiness level is configured on each network transmission device.

Device-E colors the advertised BGP routes through the color extended community attribute, and different services correspond to different colors.

When Device-A receives a BGP route with color C1 and endpoint E, device a will automatically generate an SR policy group (C1, E) according to the ODN template of color C1.
The composition SR policy of SR policy group is also generated according to ODN template. DSCP1 is mapped to color C2. After creating an SR policy group (C1, E), Device-A generates an ODN SR policy (C2, E) according to the mapping relationship between DSCP and color (DSCP1->C2).

Services with different security levels use different DSCPs. When the user generates a service packet, it carries the corresponding DSCP value (DSCP1) on the IPv6 packet header, and sends it to the Device-A. After receiving the service packet, the service packet is steered according to SR policy (C2, E).

6.6. Best-effort Forwarding Scenarios when SR Policy Becomes Unavailable

When all the constituent SR policies in the SR policy group are not valid, or all the selected paths of the SR policy are unavailable, the service traffic will not be forwarded according to the specified path. At this time, the best-effort forwarding path can be configured for the SR policy group, and the endpoints through which traffic forwarding must pass can be designed in the best-effort forwarding path.

During network deployment, The best-effort forwarding path can be a default SR policy or an SR BE forwarding path. Specify an best-effort forwarding path in the SR policy group. When all specified candidate paths are invalid, or the mapping relationship corresponding to their service type is not matched in the SR policy group, select the default best-effort path forwarding.

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

This document presents use cases to be considered by the deployment of SR Policy. It does not introduce any security considerations.

9. References

[I-D.ietf-idr-segment-routing-te-policy]
[I-D.ietf-spring-segment-routing-policy]

[I-D.lin-opsec-trustroute-problem-statement]


Authors’ Addresses

Wenying Jiang
China Mobile

Email: jiangwenying@chinamobile.com

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Changwang Lin
New H3C Technologies

Email: linchangwang.04414@h3c.com

Yuanxiang Qiu
New H3C Technologies

Email: qiuyuanxiang@h3c.com
Abstract

Network slicing can be used to meet the connectivity and performance requirement of different services or customers in a shared network. An IETF network slice can be realized as enhanced VPNs (VPN+), which is delivered by integrating the overlay VPN service with a Virtual Transport Network (VTN) as the underlay. An end-to-end IETF network slice may span multiple network domains. Within each domain, traffic of the end-to-end network slice service is mapped to a domain VTN. In the context of IETF network slicing, a VTN can be instantiated as a Network Resource Partition (NRP).

When segment routing (SR) is used to build a multi-domain IETF network slices, information of the local network slices in each domain can be specified using special SR binding segments called NRP binding segments (NRP BSID). The multi-domain IETF network slice can be specified using a list of NRP BSIDs in the packet, each of which can be used by the corresponding domain edge nodes to steer the traffic of end-to-end IETF network slice into the specific NRP in the local domain.

This document describes the functionality of NRP binding segment and its instantiation in SR-MPLS and SRv6.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 25 September 2022.

Copyright Notice

Copyright (c) 2022 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
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 Revised BSD License text as
described in Section 4.1 of the Trust Legal Provisions and are
provided without warranty as described in the Revised BSD License.

Table of Contents

1.  Introduction ............................................. 3
2.  Segment Routing for IETF E2E Network Slicing ............ 4
3.  SRv6 NRP Binding Functions ............................. 5
   3.1.  End.B6NRP.Encaps ................................. 5
   3.2.  End.NRP.Encaps .................................. 6
   3.3.  End.BNRP.Encaps ................................ 7
4.  SR-MPLS NRP BSIDs ...................................... 8
5.  IANA Considerations .................................... 9
6.  Security Considerations ................................. 9
7.  Acknowledgements ....................................... 9
8.  References ............................................. 10
   8.1.  Normative References ............................. 10
   8.2.  Informative References ......................... 10
Authors’ Addresses ......................................... 11
1. Introduction

[I-D.ietf-teas-ietf-network-slices] introduces the concept and the characteristics of IETF network slice, and describes a general framework for IETF network slice management and operation. It also introduces the concept Network Resource Partition (NRP), which is a collection of resources identified in the underlay network.

[I-D.ietf-teas-enhanced-vpn] describes the framework and the candidate component technologies for providing enhanced VPN (VPN+) services based on existing VPN and Traffic Engineering (TE) technologies with enhanced characteristics that specific services require above traditional VPNs. It also introduces the concept of Virtual Transport Network (VTN). A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the underlay, so as to provide the network characteristics required by the customers. Enhanced VPN (VPN+) and VTN can be used for the realization of IETF network slices. In the context of IETF network slicing, a VTN can be instantiated as an NRP. VTN and NRP are considered interchangable terms in this document.

[I-D.dong-teas-nrp-scalability] describes the scalability considerations in the control plane and data plane to enable NRPs and provide the suggestions to improve the scalability of NRP. In the control plane, it proposes the approach of decoupling the topology and resource attributes of NRP, so that multiple NRPs may share the same topology and the result of topology based path computation. In the data plane, it proposes to carry a dedicated NRP-ID of a network domain in the data packet to determine the set of resources reserved for the corresponding NRP.

An IETF network slice may span multiple network domains. Within each domain, traffic of the end-to-end network slice is mapped to a local network slice. The NRP ID which identifies the NRP in the local domain for the end-to-end network slice needs to be determined on the domain edge node.
When segment routing (SR) is used to build a multi-domain IETF network slice, information of the local network slices in each domain can be specified using special SR binding segments called NRP binding segments (NRP BSID). The multi-domain IETF network slice can be specified using a list of NRP BSIDs in the packet, each of which can be used by the corresponding domain edge nodes to steer the traffic of end-to-end IETF network slice using the specific resource-aware segments or NRP-ID of the local domain.

This document describes the functionality of the network slice binding segment and its instantiation in SR-MPLS and SRv6.

2. Segment Routing for IETF E2E Network Slicing

[I-D.dong-teas-nrp-scalability] describes the scalability considerations in the control plane and data plane to create NRPs. In data plane, it proposes to carry a dedicated NRP-ID in data packet to determine the set of resources reserved for the corresponding NRP in a network domain.

[I-D.li-teas-e2e-ietf-network-slicing] describes the framework of carrying network slice related identifiers in the data plane, each of the network slice IDs may have a different network scope. It provides an approach of mapping the global NRP-ID to domain NRP-IDs at the network domain border nodes.

With Segment Routing, there are several optional approaches to realize the mapping between the end-to-end network slice and the network slice constructs in the local domain.

The first type of approaches are to use one type of NRP BSID to steer traffic to an SR Policy associated with a local NRP. This is called the NRP-TE BSID. There are some variants in terms of the detailed behavior:

* The first variant is to use one type of NRP BSID to specify the mapping of traffic to a SR policy which consists of list of resource-aware segments [I-D.ietf-spring-resource-aware-segments] associated with a local NRP.

* The second variant is to use one type of NRP BSID to specify the mapping of traffic to a SR policy which is bound to a local NRP-ID.

The second type of approaches is to use one type of NRP BSID to steer traffic to follow the shortest path within a local domain NRP. This is called the NRP-BE BSID. There are some variants in terms of the detailed behavior:
The first variant is to use one type of NRP BSID to determine a local NRP-ID, and instruct the encapsulation of the local NRP-ID into the packet at the domain edge node.

The second variant is to use one type of NRP BSID to specify the mapping of traffic to a local NRP, the local NRP-ID is specified in the associated fields by the ingress node, and is encapsulated into the packet at the domain edge node.

The behavior of the first type of NRP BSID is similar to the function of the existing SR BSID, the difference is it is associated with a particular NRP. The second type of the NRP BSID is different from the existing binding segment. The instantiation of the NRP BSIDs in SR-MPLS and SRv6 are described in the following sections.

3. SRv6 NRP Binding Functions

[RFC8986] defines the SRv6 Network Programming concept and specifies the base set of SRv6 behaviors. The SRv6 End.B6.Encaps function is defined to instantiate the Binding SID in SRv6, which can be reused as one type of NRP-TE BSID to specify the mapping of traffic to a list of resource-aware SRv6 segments of a domain NRP.

[I-D.ietf-6man-enhanced-vpn-vtn-id] describes the mechanism of carrying the VTN-ID of a network domain in the IPv6 Hop-by-Hop (HBH) extension header. For the type 2, 3, 4 of NRP binding segments described in section 2, three new SRv6 Binding functions are defined in the following sections.

3.1. End.B6NRP.Encaps

A new SRv6 function called End.B6NRP.Encaps: Endpoint bound to a SRv6 Policy in a NRP with IPv6 encapsulation is defined in this section. This is a variation of the End behavior. It instructs the endpoint node to determine an SRv6 Policy in a specific NRP of the local domain, and encapsulate the SID list of the SR Policy and the NRP-ID in a new IPv6 header.

Any SID instance of this behavior is associated with an SR Policy B, a NRP-ID V and a source address A.

When node N receives a packet whose IPv6 DA is S, and S is a local End.B6NRP.Encaps SID, N does the following:
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
            with 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
            with Code 0 (Erroneous header field encountered)
            and 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, and
        the VTN-ID in VTN option set to V in the HBH Ext header
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 for
        transmission to the new destination
S20. }

3.2. End.NRP.Encaps

A new SRv6 function called End.NRP.Encaps is defined. This is a
variation of the End behavior. It instructs the endpoint node to
determine the corresponding NRP-ID of the local domain based on the
mapping relationship between the End.NRP.Encaps SID and the NRPs
maintained on the endpoint. The NRP-ID is encapsulated in the VTN
option in the IPv6 HBH extension header.

Any SID instance of this behavior is associated with one NRP-ID V and
a source address A.

When node N receives a packet whose IPv6 DA is S, and S is a local
End.NRP.Encaps SID, N does the following:
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
         with 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
         with Code 0 (Erroneous header field encountered)
         and 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.   Set the VTN-ID in VTN option to V in the HBH Ext header
S16.   Submit the packet to the egress IPv6 FIB lookup for
         transmission to the new destination
S17. }

3.3. End.BNRP.Encaps

A new SRv6 function called End.BNRP.Encaps: Endpoint bound to a NRP
with IPv6 encapsulation is defined. This is a variation of the End
behavior. For the End.BNRP SID, its corresponding NRP-ID should be
specified and encapsulated by the ingress node of SRv6 Path. It
instructs the endpoint node to obtain the corresponding NRP-ID from
the SRH, and encapsulate it in the VTN option in the IPv6 HBH
extension header. Through the End.BNRP.Encaps, the ingress node can
flexibly specify the local NRP the packet traverses in the network.

Any SID instance of this behavior is associated with one NRP-ID V and
a source address A.

There can be several options to carry the local NRP-ID corresponding
to the End.BNRP.Encaps function:

1. The NRP-ID is carried in the argument field of the
   End.BNRP.Encaps SID.

2. The NRP-ID is carried in the SRH TLV field.
3. The NRP-ID is carried in the next SID following the End.BNRP.Encaps SID in the SID list.

Editor’s note: In the current version of this document, option 1 is preferred, in which the local NRP-ID is carried in the argument field of the SRv6 SID.

When an ingress node of an SR path encapsulates the End.BNRP.Encaps SID into the packet, it SHOULD put the NRP-ID which the packet is expected to be mapped to into the argument part of the SID.

When node N receives a packet whose IPv6 DA is S, and S is a local End.BNRP.Encaps SID, N does the following:

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 with 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 with Code 0 (Erroneous header field encountered) and Pointer set to the Segments Left field, interrupt packet processing, and discard the packet.
S11.   }
S12.   Obtain the NRP-ID V from the argument part of the IPv6 DA
S13.   Decrement IPv6 Hop Limit by 1
S14.   Decrement Segments Left by 1
S15.   Update IPv6 DA with Segment List [Segments Left]
S16.   Set the VTN-ID in VTN option to V in the HBH Ext header
S17.   Submit the packet to the egress IPv6 FIB lookup for transmission to the new destination
S18. }

4. SR-MPLS NRP BSIDs

[I-D.li-mpls-enhanced-vpn-vtn-id] describes the mechanism of carrying the VTN-ID of a network domain in the MPLS extension header.
With SR-MPLS data plane, NRP BSIDs can be allocated by a domain edge node for the three types of NRP binding behaviors described in section 2.

For the first type of NRP BSID, a BSID can be bound to a list of resource-aware segments of a local NRP. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding SID list which consists of the resource-aware segments of a local NRP, and encapsulates the SID list to the MPLS label stack.

For another variant of the first type NRP BSID, a NRP BSID is bound to a SR Policy and a local NRP-ID. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding SID list and the local NRP-ID, and encaps the packet with the SID list and an MPLS VTN extension header which carries the local NRP-ID. Note this requires to assign a NRP BSID for each SR policy in each NRP the node participates in.

For the second type of NRP BSID, a NRP BSID is bound to the shortest path in an NRP of the local network domain. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding local NRP-ID based on the mapping relationship between the NRP BSID and the NRP-ID, and encapsulates the packet with an MPLS VTN extension header which carries the local NRP-ID. Note this requires to assign a NRP BSID for each local NRP.

For a variant of the second type NRP BSID, a NRP BSID is bound to the shortest path in an NRP of the local network domain, the NRP-ID is specified and encapsulated by the ingress node in the MPLS VTN extension header. When a node receives a packet with a locally assigned NRP BSID, it obtains the corresponding local NRP-ID from the NRP-ID list in the VTN extension header, and update the local NRP-ID in the VTN extension header with the obtained NRP-ID.

5. IANA Considerations

TBD

6. Security Considerations

TBD

7. Acknowledgements

The authors would like to thank Zhibo Hu for his review and valuable comments.
8. References

8.1. Normative References

[I-D.ietf-teas-enhanced-vpn]

[I-D.ietf-teas-ietf-network-slices]

[I-D.li-teas-e2e-ietf-network-slicing]


8.2. Informative References

[I-D.dong-teas-nrp-scalability]

[I-D.ietf-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra,
"Carrying Virtual Transport Network (VTN) Identifier in
IPv6 Extension Header", Work in Progress, Internet-Draft,
draft-ietf-6man-enhanced-vpn-vtn-id-00, 5 March 2022,

[I-D.ietf-spring-resource-aware-segments]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad,
"Introducing Resource Awareness to SR Segments", Work in Progress, Internet-Draft,
draft-ietf-spring-resource-aware-segments-04, 5 March 2022,

[I-D.ietf-spring-sr-for-enhanced-vpn]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad,
"Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", Work in Progress,
Internet-Draft, draft-ietf-spring-sr-for-enhanced-vpn-02, 5 March 2022,

[I-D.li-mpls-enhanced-vpn-vtn-id]
Li, Z. and J. Dong,
"Carrying Virtual Transport Network Identifier in MPLS Packet", Work in Progress, Internet-Draft,
draft-li-mpls-enhanced-vpn-vtn-id-02, 7 March 2022,

Authors’ Addresses

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: lizhenbin@huawei.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: jie.dong@huawei.com

Ran Pang  
China Unicom  
Email: pangran@chinaunicom.cn

Yongqing Zhu  
China Telecom  
Email: zhuyq8@chinatelecom.cn
Abstract

Bidirectional Forwarding Detection (BFD) can be used to monitor paths between nodes. Seamless BFD (S-BFD) provides a simplified mechanism which is suitable for monitoring of paths that are setup dynamically and on a large scale network. In SRv6, when a headend use S-BFD to monitor the segment list/CPath of SRv6 Policy, the forward path of control packet is indicated by segment list, the reverse path of response control packet is via the shortest path from the reflector back to the initiator (headend) as determined by routing. The forward path and reverse path of control packet are likely inconsistent going through different intermediate nodes or links. This document describes a method to keep the forward path and reverse path of S-BFD consistent when detecting SRv6 Policy.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html
1. Introduction

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Per-path states of Intermediate nodes are eliminated thanks to source routing. The headend node steers a flow into an SR Policy. The packets steered into an SR Policy carry an ordered list of segments associated with that SR Policy.

S-BFD is used to monitor different kinds of paths between nodes. In SRv6, when a headend use S-BFD to monitor the segment list/CPath of SRv6 Policy, the forward and reverse path of S-BFD packet are inconsistent with high probability because the reverse path is via
IPv6 forwarding and forward path is via SRv6 segment list (loose path or explicit path).

The inconsistency impacts the detecting result. If the forward path is up and reverse path is down, then the S-BFD session will be down. If there are multiple path (segment list) in a SRv6 Policy between a headend (initiator) router and a tailend(reflector) router, multiple S-BFD session will be created for each path. Each S-BFD session uses corresponding path to send control packet, but the reverse path is identical for all S-BFD sessions. If the reverse path is down, all sessions will be down. Then the SRv6 Policy is down.

The consistency of forward and reverse path of the same S-BFD session should be guaranteed. This document describes a method to keep the forward path and reverse path of S-BFD consistent using path segment when detecting SRv6 Policy.

1.1. 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.

2. Requirement for S-BFD in SRv6

Monitor SRv6 Policy using S-BFD is usually based on segment list S-BFD creates session for each segment list and associates the session with segment list.

When S-BFD initiator detects the continuity of an S-BFD session, it will use the associated segment list to encapsulate IPv6 header and SRH of the control packet.

After the reflector receives the S-BFD control packet, the response control packet should be able to return along the path to avoid the false detection of the session caused by the inconsistency of the forward and reverse paths.

Referring to the following topology, there are two paths between Node A and D, and All nodes allocate end.x Segments. Node A and D are headend and tailend nodes of each other, and SRv6 policy is created on A and D respectively.
Assuming that the deployed SRv6 policy has one candidate path and each path has two segment lists. For ease of description, segment lists with the same number on Node A and D are forward and reverse paths to each other.

Node A:                       Node D:  
SRv6 Policy A-D               SRv6 Policy D-A  
  Candidate Path1             Candidate Path1  
    Segment list1             Segment list1  
       SID-A1, SID-B2, SID-C2    SID-D1, SID-C1, SID-B1  
    Segment list2             Segment list2  
       SID-A2, SID-E2           SID-D2, SID-E1  

When node A is the S-BFD initiator, S-BFD sessions for segment list1 and segment list2 could be created respectively.

The control packet of S-BFD session associated with the segment list1 is forwarded to node D according to the segment list1 of node A. The response control packet of node D needs to be returned to node A according to the segment list1 of node D. Thus the forward and reverse paths of S-BFD packets are ensured to be consistent.

3. Correlate bidirectional path using Path Segment

A Path Segment is defined to identify an SR path in [draft-ietf-spring-srv6-path-segment]. SRv6 Path segments can be used to correlate the two unidirectional SRv6 paths at both ends of the paths.

[draft-ietf-idr-sr-policy-path-segment] proposes an extension to BGP SR Policy distribute SR policies carrying Path Segment and bidirectional path information.
Through this extension, when distributing SRv6 policy to the headend, reverse path information and path segment of segment list can be carried together.

Node A                       Node D

SRv6 Policy A-D                SRv6 Policy D-A
  Candidate Path1               Candidate Path1
    Segment list1
      SID-A1, SID-B2, SID-C2  SID-D1, SID-C1, SID-B1
    Path Segment: SID-Path-1  Path Segment: SID-Path-2
    Reverse Path Segment:    Reverse Path Segment:
      SID-Path-2              SID-Path-1
    Segment list2
      SID-A2, SID-E2           SID-D2, SID-E1
    Path Segment: SID-Path-3  Path Segment: SID-Path-4
    Reverse Path Segment:    Reverse Path Segment:
      SID-Path-4              SID-Path-3

In this way, on the headend in both directions of the forward and reverse paths, the path segment of the paths in both directions can be obtained, and the paths in both directions use the same intermediate link.

The headend can use path segment in two directions to establish a mapping table. Using this mapping table, the headend can index the reverse path through the path segment of the forward path.

The mapping table of Node A and Node D is shown below:

Node A:

```
+-----------------+          +--------------------+
|  Path Segment   |          |Reverse Path Segment|
+-----------------+          +--------------------+
|  SID-Path-1     |-+        | SID-Path-2         |--+
+-----------------+ |        +--------------------+  |
|  SID-Path-3     | |        | SID-Path-4         |--|-+ |
+-----------------+ |        +--------------------+  | | |
|            |                                | |
|            |  +-----------------------+     | |
|            |  | segment List          |     | |
|            |  +-----------------------+     | |
|            +->|SID-A1, SID-B2, SID-C2 |<----+ |
|               +-----------------------+       |
+-------------->|SID-A2, SID-E2         |<------+
+-----------------------+
```

4. S-BFD Procedure with Path segment

This document proposes to forward S-BFD control packets and response control packets through the consistent path by path segment.

4.1. S-BFD Initiator procedure

For instance, the S-BFD initiator is Node A in Figure 1, and the S-BFD session is bounded with Segment List 1 of Policy A-D. The encapsulation format of S-BFD control packet is as follows:

Figure 2: mapping table
NodeA Encapsulates the path segment of segment list1 in SRH, and set
SRH.P-Flag.

The S-BFD control packet is as follows:
4.2. S-BFD Reflector procedure

S-BFD control packet is forwarded along the path A→B→C→D. While packet arrives at Node D, RH.SL is 0 and the destination address is IPv6 address of Node D. Packet is delivered up to the S-BFD module in control plane.

S-BFD module detects SRH.P-flag is set, extracts the path segment of the forward path from SRH, gets the path segment of the reverse path through the mapping table. When responding to S-BFD control packet, S-BFD module uses the segment list associated with path segment of the reverse path to encapsulate SRH.

The encapsulation format of S-BFD response control packet is as follows:
The Example of S-BFD response control packet is as follows:

Figure 5: Encapsulation format of S-BFD response control packet
5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

The security requirements and mechanisms described in [RFC8402] and [RFC8754] also apply to this document.

This document does not introduce any new security consideration.

7. References

7.1. Normative References


Lin, et al. Expires October, 2022


Contributors

Yisong Liu contributed to the content of this document.
Authors’ Addresses

Changwang Lin
New H3C Technologies
Beijing
China

Email: linchangwang.04414@h3c.com

Weiqiang Cheng
China Mobile
Beijing
CN

Email: chengweiqiang@chinamobile.com

Wenying Jiang
China Mobile
Beijing
CN

Email: jiangwenying@chinamobile.com
SRv6 inter-domain mapping SIDs
draft-salih-spring-srv6-inter-domain-sids-02

Abstract

This document describes three new SRv6 end-point behaviors, called END.REPLACE, END.REPLACEB6 and END.DB6. These behaviors are used in distributed inter-domain solutions and are normally executed on border routers. They also can be used to provide multiple intent-based paths across these domains.

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 3 September 2022.

Copyright Notice

Copyright (c) 2022 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.
1. Overview

Segment Routing (SR) [RFC8402] allows source nodes to steer packets through SR paths. It can be implemented over IPv6 [RFC8200] or MPLS [RFC3031]. When SR is implemented over IPv6, it is called SRv6 [RFC8986].

This document describes three new SRv6 end-point behaviors, called END.REPLACE, END.REPLACEB6 and END.DB6. These behaviors are used to build paths across SRv6 domains. They also facilitate end-to-end SRv6 intent-based path stitching.

2. 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. Usecases

3.1. usecase 1

This use-case is mentioned in Section 4.1.1 of [I-D.hegde-spring-mpls-seamless-sr].

Figure 1: Multiple ASes connected with E-BGP

Figure 1 depicts three ASes (AS1, AS2 and AS3). All the three domains deploy SRv6. Inter-provider Option C[RFC4364] connectivity is maintained from PE1 to PE2.

3.2. usecase 2

Figure 2: Single AS with different IGP domains
The above diagram Figure 2 shows two different SRv6 IGP domains. Services are running between PE1 and PE2 in option B [RFC4364] style. The requirement here is to avoid service route lookup on ABR1 and ABR2 to provide option B style end to end connectivity

4. SRv6 SID Behaviors

4.1. END.REPLACE

The END.REPLACE behavior is applicable in the Multiple ASes Connected With E-BGP (Section 3.1) use-case.

The End.REPLACE SID cannot be the last segment in SRH or SR Policy.

Any SID instance of this behavior is associated with a set, J, of one or more L3 adjacencies of immediate BGP neighbors

When Node N receives a packet destined to S and S is a locally instantiated End.REPLACE SID, Node N executes the following procedure:

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. Procedure is as per Section 4.1.1 of [RFC8986].
S04.   }}
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address with Code 0 (Hop limit exceeded in transit), interrupt packet processing, and discard packet
S07.   }
S08.   Decrement IPv6 Hop Limit by 1
S09.   Update IPv6 DA with new destination address(SID) mapped with END.REPLACE SID.
S10.   Submit the packet to the IPv6 module for transmission to the new destination via a member of J.
S11. }

4.2. END.REPLACEB6

The END.REPLACEB6 behavior is applicable in the Multiple ASes Connected With E-BGP (Section 3.1) use-case.

The End.REPLACEB6 SID cannot be the last segment in a SRH or SR Policy.
Node N is configured with an IPv6 address T (e.g., assigned to its loopback).

When Node N receives a packet destined to S and S is a locally instantiated End.REPLACEB6 SID, Node N executes the following procedure:

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. Procedure is as per Section 4.1.1 of [RFC8986].
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address with Code 0 (Hop limit exceeded in transit), interrupt packet processing, and discard packet
S07.   }
S08.   Decrement IPv6 Hop Limit by 1
S09.   Update IPv6 DA with new destination address(SID) mapped with END.REPLACEB6.
S10.   Push an IPv6 header with an SRH.
S11.   Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segment list.
S13.   Set the outer Next Header value
S14.   Submit the packet to the IPv6 module for transmission to the First SID.
S15. }

Note:
S10 - S13. Implementation may choose to avoid outer encapsulation for flex-algo and best effort based SRv6 transport tunnels.
S12. The Payload Length, Traffic Class, Hop Limit, and Next Header fields are set as per [RFC2473]. The Flow Label is computed as per [RFC6437].

4.3. END.DB6

For the use-case mentioned under Section 3.2 END.DB6 SID is applicable.

The End.DB6 SID MUST be the last segment in SRH or SR Policy.

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

When Node N receives a packet destined to S and S is a locally instantiated End.DB6 SID, Node N executes the following procedure:
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.   If (Upper-Layer header type == 4(IPv4) OR Upper-Layer header type == 4
1(IPv6) OR
       Upper-Layer header type == 143(Ethernet)) {
S06.     Remove the outer IPv6 header with all its extension headers.
S07.     Push the new IPv6 header with the SRv6 SIDs associated with the END.D
B6 sid in an SRH.
S08.     Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segme
nt list.
S09.     Set outer Payload Length, Traffic Class, Hop Limit, and Flow Label fi
elds
S10.     Set the outer Next Header value
S11.     Submit the packet to the IPv6 module for transmission to First SID.
S12.   } else {
S13.     Process as per Section 4.1.1 of [RFC8986].
S14.   }
S15. }

Note :
S09. The Payload Length, Traffic Class, Hop Limit, and Next Header fields
are set as per [RFC2473]. The Flow Label is
computed as per [RFC6437].

5. Interworking Procedures

Here we will describe the control plane and data plane procedures by
taking examples.

Node n has a classic IPv6 loopback address An::1/128. One of the SID
at node n with locator block B and function F is represented by
B:n:F::sid_num.

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.

5.1. Option C Transport Interworking

Here we will discuss the use-case mentioned under Section 3.1
Node [1] acts as ingress PE and Node [16] acts as egress PE.

Nodes [2], [3], [8], [9], [14] and [15] are P routers.

Nodes [4], [5], [6], [7], [10], [11], [12] and [13] are ASBR routers.

A VPN route is advertised via service RRs between an egress PE(node 16) and an ingress PE (node 1). The example below shows IBGP-CT connection between border routers in each domain and single hop EBGP-CT for inter-domain connections. However the forwarding procedure for the sids remains the same irrespective of the the various inter-domain protocol extensions used to advertise the sids. AS1, AS2 and AS3 has SRTE policy for the required intent paths.
Control plane example:

For simplicity only one path is tracked.

For a route if the next hop is one hop away then while advertising use END.REPLACE SID. For a route if the next hop is multi hop away then while advertising use END.REPLACEB6 SID. For single hop neighbor case, no encap required as it is just replace and forward on specific link while in multihop case one encap will be required.

Routing Protocol(RP) @16:
* In ISIS advertise locator B:16::/48 and an END SID B:16::END::1.
* BGP AFI=1,SAFI=128 originates a VPN route RD:V/v via A:16::1 and Prefix-SID attribute B:16:DT4::1.
  This route is advertised to service RR with color extended community red.
* BGP originates prefix A:16::1 with color red to ASBR [12] with SRv6 SID B:16:END::1 since its the egress node.

RP @12:
* BGP receives the route A:16::1 over the ibgp session and readvertises with nexthop self to ASBR [10].
  it advertises the SRv6 SID B:12:REPLACEB6::1 in the protocol extensions. As the advertisement was received on a multihop i-bgp session this node allocates a REPLACEB6 sid.

RP @10:
* BGP receives the route A:16::1 over the ebgp session and readvertises with nexthop self to ASBR [6].
  it advertises the SRv6 SID B:10:REPLACE::1 in the protocol extensions. As the advertisement was received on a single hop e-bgp session this node allocates a REPLACE sid.

RP @6:
* BGP receives the route A:16::1 over the ibgp session and readvertises with nexthop self to ASBR [4].
  it advertises the SRv6 SID B:6:REPLACEB6::1 in the protocol extensions. As the advertisement was received on a multihop i-bgp session this node allocates a REPLACEB6 sid.

RP @4:
* BGP receives the route A:16::1 over the ebgp session and readvertises with nexthop self to PE [1].
  it advertises the SRv6 SID B:4:REPLACE::1 in the protocol extensions. As the advertisement was received on a single hop e-bgp session this node allocates a REPLACE sid.

RP @1:
* BGP receives the route A:16::1 with color red over the ibgp session.
* BGP AFI=1, SAFI=128 learn service prefix RD:V/v, next hop A:16::1 and PrefixSID attribute TLV type 5 with SRv6 SID B:16:DT4
FIB State:

@1: IPv4 VRF V/v => H.Encaps.red <B:2:END::1, B:4:REPLACE::1, B:16:DT4::1 > with SRH, SRH NextHeader=IPv4 where the first sid B:2:END::1 belongs to the SR-policy in AS1.

@2: IPv6 Table: B:2:END::1 => Update DA with B:4:REPLACE::1, decrement SL and forward towards the ASBR [4].

@4: IPv6 Table: B:4:REPLACE::1 => Update DA with B:6:REPLACEB6::1 and forward on the interface/interfaces identified by the ebgp neighbor; the SL remains at 1.

@6: IPv6 Table: B:6:REPLACEB6::1 => Update DA with B:8:END::1, B:10:END::1> with SRH where the new SRH SIDs belong to SR policy in AS2.

@8: IPv6 Table: B:8:END::1 => Update outer IPv6 packet DA with B:10:END::1 AND do a fresh H.Encaps.red <B:8:END::1, B:10:END::1> and forward towards ASBR [10]

@10: IPv6 table: B:10:END::1 => Decap Outer IPv6 header and lookup next I Pv6 DA B:10:REPLACE::1 => Update DA to B:12:REPLACEB6:1 and forward on the interface/interfaces identified by the ebgp neighbor. SL remains at 1.

@12: IPv6 Table B:12:REPLACEB6::1 => Update DA with B:16:END::1 and do a fresh H.Encaps.red <B:15:END::1, B:16:END::1> with SRH where the new SIDs belong to the SR policy in AS3.

@15: IPv6 Table B:15:END::1 => Update outer IPv6 packet DA with B:16:END::1 and forward towards [16].

@16: IPv6 Table B:16:END::1 => Decap the outer header and lookup the inner DA which results in B:16:DT4:1 lookup. DT4 lookup results in Decap and inner IPv4 packet DA lookup in the corresponding VRF.

Note: At [16] its possible to optimize the lookups required with minor control plane extensions.

5.2. Option B service interworking

Here we will discuss the use-case mentioned under Section 3.2

---MP-IBGP/---- ---MP-IBGP/--
| EBGP | EBGP |

-----[2]------[5]-----


-----[3]------[6]-----

---SRv6--- | ---SRv6---

Figure 4: Option B style Service Interworking

Control Plane example:

Routing Protocol (RP) @7:
* BGP AFI=1, SAFI=128 originates a VPN route RD:V/v via A:7::1 and Prefix-SID attribute B:7:DT4::1. This route is advertised to service RR [4].

RP @4:
* BGP receives the route over MP-IBGP/MP-EBGP session and readvertises with nexthop self to PE [1].
  it advertises the SRv6 SID B:4:DB6::1 in the Prefix-SID attribute TLV along with it. For all prefixes having SRv6 service SID B:7:DT4::1; the same DB6 SID B:4:DB6::1 will be reused. If a different service sid B:7:DT4::2 comes then a different DB6 SID B:4:DB6::2 will be allocated. This ensures that if the egress allocates per CE sid; the translation at border also ensures per CE sid.

RP @1:
* BGP AFI=1, SAFI=128 learns service prefix RD:V/v, next hop A:4::1 and PrefixSID attribute TLV type 5 with SRv6 SID B:4:DB6::1

FIB State:

@1: IPv4 VRF V/v => H.Encaps.red <B:4:DB6::1> with SRH, SRH NextHeader=IPv4 where the first sid belongs to the SR-policy in AS1
@4: IPv6 Table: B:4:DB6::1 => Decapsulate the incoming IPv6 header and H.Encaps <B:7:DT4::1>
@7: IPv6 Table: B:7:DT4::1 => Decapsulate the header and lookup the inner IPv4 packet DA in the VRF

6. IANA Considerations

This document requires no IANA action.

The authors will request an early allocation from the "SRv6 Endpoint Behaviors" sub-registry of the "Segment Routing Parameters" registry.

7. Security Considerations

Because SR inter-working requires co-operation between inter-working domains, this document introduces no security consideration beyond those addressed in [RFC8402], [RFC8754] and [RFC8986].

8. Contributors

Jie Dong
Huawei Technologies
Email: jie.dong@huawei.com

Swamy SRK
Juniper Networks
Email: swamys@juniper.net

G. Sri Karthik Goud
Juniper Networks
Email: gkarthik@juniper.net
9. Acknowledgements

Thanks to Ram Santhanakrishnan, Srihari Sangli, Rajendra Prasad Bollam and Kiran Kushalad for their valuable comments.

10. References

10.1. Normative References


10.2. Informative References

[I-D.hegde-spring-mpls-seamless-sr]


Authors’ Addresses

Salih K A
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: salih@juniper.net

Shraddha Hegde
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: shraddha@juniper.net

Rajesh
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: mrajesh@juniper.net

Ron Bonica
Juniper Networks
Herndon, Virginia 20171
United States of America
Email: rbonica@juniper.net
Haibo Wang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: rainsword.wang@huawei.com

Peng Shaofu
ZTE Corporation
China
Email: peng.shaofu@zte.com.cn
Circuit Style Segment Routing Policies
draft-schmutzer-pce-cs-sr-policy-02

Abstract

This document describes how Segment Routing (SR) policies can be used to satisfy the requirements for strict bandwidth guarantees, end-to-end recovery and persistent paths within a segment routing network. SR policies satisfying these requirements are called "circuit-style" SR policies (CS-SR policies).

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 6 November 2022.

Copyright Notice

Copyright (c) 2022 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
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 Revised BSD License text as
described in Section 4.e of the Trust Legal Provisions and are
provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ................................. 3
2. Terminology .................................. 3
3. Reference Model ............................... 4
4. CS-SR Policy Characteristics .................. 5
5. CS-SR Policy Creation .......................... 6
  5.1. Maximum Segment Depth ................. 7
6. Recovery Schemes ............................. 8
  6.1. Unprotected ............................ 8
  6.2. 1:1 Protection ......................... 9
  6.3. Restoration ................................ 10
    6.3.1. 1+R Restoration .................... 10
    6.3.2. 1:1+R Restoration ................... 10
7. Operations, Administration, and Maintenance (OAM) ............... 11
  7.1. Connectivity Verification ............... 11
  7.2. Performance Measurement .................. 11
  7.3. Candidate Path Validity Verification .... 12
8. External Commands ................................ 12
  8.1. Candidate Path Switchover ............... 12
  8.2. Candidate Path Recomputation ............. 12
9. Security Considerations ........................ 12
10. IANA Considerations .......................... 13
11. Acknowledgements ............................ 13
12. Contributors ................................ 13
13. References .................................. 13
  13.1. Normative References ................... 13
  13.2. Informative References .................. 13
Authors’ Addresses ................................. 18
1. Introduction

Segment routing does allow for a single network to carry both typical IP (connection-less) services and connection-oriented transport services commonly referred to as "private lines". IP services typically require ECMP and TI-LFA, while transport services that normally are delivered via dedicated circuit-switched SONET/SDH or OTN networks do require:

* Persistent end-to-end traffic engineered paths that provide predictable and identical latency in both directions
* Strict bandwidth commitment per path to ensure no impact on the Service Level Agreement (SLA) due to changing network load from other services
* End-to-end protection (<50msec protection switching) and restoration mechanisms
* Monitoring and maintenance of path integrity
* Data plane remaining up while control plane is down

Such a "transport centric" behavior is referred to as "circuit-style" in this document.

This document describes how SR policies [I-D.ietf-spring-segment-routing-policy] and the use of adjacency-SIDs defined in the SR architecture [RFC8402] together with a stateful Path Computation Element (PCE) [RFC8231] can be used to satisfy those requirements. It includes how end-to-end recovery and path integrity monitoring can be implemented.

SR policies that satisfy those requirements are called "circuit-style" SR policies (CS-SR policies).

2. Terminology

* BSID : Binding Segment Identifier
* CS-SR : Circuit-Style Segment Routing
* ID : Identifier
* LSP : Label Switched Path
* LSPA : LSP attributes
3. Reference Model

The reference model for CS-SR policies is following the Segment Routing Architecture [RFC8402] and SR Policy Architecture [I-D.ietf-spring-segment-routing-policy] and is depicted in Figure 1.

By nature of CS-SR policies, paths will be computed and maintained by a stateful PCE defined in [RFC8231]. The stateful PCE provides a consistent simple mechanism for initializing the co-routed bidirectional end to end paths, performing bandwidth allocation control, as well as monitoring facilities to ensure SLA compliance for the live of the CS-SR Policy. When using a MPLS data plane
In order to satisfy the requirements of CS-SR policies, each link in the topology MUST have:

* An adjacency-SID which is:
  - Manually allocated or persistent: to ensure that its value does not change after a node reload
  - Non-protected: to avoid any local TI-LFA protection to happen upon interface/link failures

* The bandwidth available for CS-SR policies specified

* A per-hop behavior ([RFC3246] or [RFC2597]) that ensures that the specified bandwidth is available to CS-SR policies at all times independent of any other traffic

When using a MPLS data plane [RFC8660] existing IGP extensions defined in [RFC8667] and [RFC8665] and BGP-LS defined in [RFC9085] can be used to distribute the topology information including those persistent and unprotected adjacency-SIDs.


4. CS-SR Policy Characteristics

A CS-SR policy has the following characteristics:

* Requested bandwidth: bandwidth to be reserved for the CS-SR policy

* Bidirectional co-routed: a CS-SR policy between A and Z is an association of an SR-Policy from A to Z and an SR-Policy from Z to A following the same path(s)

* Deterministic and persistent paths: segment lists with strict hops using unprotected adjacency-SIDs

* Not automatically recomputed or reoptimized: the SID list of a candidate path must not change automatically to a SID list representing a different path (for example upon topology change)
* Multiple candidate paths in case of protection/restoration:
  - Following the SR policy architecture, the highest preference valid path is carrying traffic
  - Depending on the protection/restoration scheme (Section 6), lower priority candidate paths
    o may be pre-computed
    o may be pre-programmed
    o may have to be disjoint

* Connectivity verification and performance measurement is activated on each candidate path (Section 7)

5. CS-SR Policy Creation

A CS-SR policy between A and Z is configured both on A (with Z as endpoint) and Z (with A as endpoint) as shown in Figure 1.

Both nodes A and Z act as PCC and delegate path computation to the PCE using the extensions defined in [RFC8664]. The PCRpt message sent from the headends to the PCE contains the following parameters:

* BANDWIDTH object (Section 7.7 of [RFC5440]) : to indicate the requested bandwidth

* LSPA object (section 7.11 of [RFC5440]) : to indicate that no local protection requirements
  - L flag set to 0 : no local protection
  - E flag set to 1 : protection enforcement (section 5 of [I-D.ietf-pce-local-protection-enforcement])

* ASSOCIATION object ([RFC8697]) : 
  - Type : Double-sided Bidirectional with Reverse LSP Association ([I-D.ietf-pce-sr-bidir-path])
  - Bidirectional Association Group TLV ([RFC9059]) :
    o R flag is always set to 0 (forward path)
    o C flag is always set to 1 (co-routed)
If the SR-policies are configured with more than one candidate path, a PCEP request is sent per candidate path. Each PCEP request does include the "SR Policy Association" object (type 6) as defined in [I-D.ietf-pce-segment-routing-policy-cp] to make the PCE aware of the candidate path belonging to the same policy.

The signaling extensions described in [I-D.sidor-pce-circuit-style-pcep-extensions] are used to ensure that

* Path determinism is achieved by the PCE only using segment lists representing a strict hop by hop path using unprotected adjacency-SIDs.

* Path persistency across node reloads in the network is achieved by the PCE only including manually configured adjacency-SIDs in its path computation response.

* Persistency across network changes is achieved by the PCE not performing periodic nor network event triggered re-optimization.

Bandwidth adjustment can be requested after initial creation by signaling both requested and operational bandwidth in the BANDWIDTH object but the PCE is not allowed to respond with a changed path.

As discussed in section 3.2 of [I-D.ietf-pce-multipath] it may be necessary to use load-balancing across multiple paths to satisfy the bandwidth requirement of a candidate path. In such a case the PCE will notify the PCC to install multiple segment lists using the signaling procedures described in section 5.3 of [I-D.ietf-pce-multipath].

5.1. Maximum Segment Depth

A Segment Routed path defined by a segment list is constrained by maximum segment depth (MSD), which is the maximum number of segments a router can impose onto a packet. [RFC8491], [RFC8476], [RFC8814] and [RFC8664] provide the necessary capabilities for a PCE to determine the MSD capability of a router. The MSD constraint is typically resolved by leveraging a label stack reduction technique, such as using Node SIDs and/or BSIDs (SR architecture [RFC8402]) in a segment list, which represents one or many hops in a given path.

As described in Section 4, adjacency-SIDs without local protection are to be used for CS-SR policies to ensure no ECMP, no rerouting due to topological changes nor localized protection is being invoked on the traffic, as the alternate path may not be providing the desired SLA.
If a CS-SR Policy path requires SID List reduction, a Node SID cannot be utilized as it is eligible for traffic rerouting following IGP re-convergence. However, a BSID can be programmed to a transit node, if the following requirements are met:

* The BSID is unprotected, hence only has one candidate path
* The BSID follows the rerouting and optimization characteristics defined in Section 4 which implies the SID list of the candidate path MUST only use unprotected adjacency-SIDs.

This ensures that any CS-SR policies in which the BSID provides transit for do not get rerouted due to topological changes or protected due to failures. A BSID may be pre-programmed in the network or automatically injected in the network by a PCE.

6. Recovery Schemes

Various protection and restoration schemes can be implemented. The terms "protection" and "restoration" are used with the same subtle distinctions outlined in section 1 of [RFC4872], [RFC4427] and [RFC3386] respectively.

* Protection : another candidate path is computed and fully established in the data plane and ready to carry traffic
* Restoration : a candidate path may be computed and may be partially established but is not ready to carry traffic

The term "failure" is used to represent both "hard failures" such as complete loss of connectivity detected by Section 7.1 or degradation, a packet loss ratio, beyond a configured acceptable threshold.

6.1. Unprotected

In the most basic scenario no protection nor restoration is required. The CS-SR policy has only one candidate path configured. This candidate path is established, activated (O field in LSP object is set to 2) and is carrying traffic.

In case of a failure the CS-SR policy will go down and traffic will not be recovered.

Typically two CS-SR policies are deployed either within the same network with disjoint paths or in two completely separate networks and the overlay service is responsible for traffic recovery.
6.2. 1:1 Protection

For fast recovery against failures the CS-SR policy is configured with two candidate paths. Both paths are established but only the candidate with higher preference is activated (O field in LSP object is set to 2) and is carrying traffic. The candidate path with lower preference has its O field in LSP object set to 1.

Appropriate routing of the protect path diverse from the working path can be requested from the PCE by using the "Disjointness Association" object (type 2) defined in [RFC8800] in the PCRpt messages. The disjoint requirements are communicated in the "DISJOINTNESS-CONFIGURATION TLV"

* L bit set to 1 for link diversity
* N bit set to 1 for node diversity
* S bit set to 1 for SRLG diversity
* T bit set to enforce strict diversity

The P bit may be set for first candidate path to allow for finding the best working path that does satisfy all constraints without considering diversity to the protect path.

The "Objective Function (OF) TLV" as defined in section 5.3 of [RFC8800] may also be added to minimize the common shared resources.

Upon a failure impacting the candidate path with higher preference carrying traffic, the candidate path with lower preference is activated immediately and traffic is now sent across it.

Protection switching is bidirectional. As described in Section 7.1, both headends will generate and receive their own loopback mode test packets, hence even a unidirectional failure will always be detected by both headends without protection switch coordination required.

Two cases are to be considered when the failure impacting the candidate path with higher preference is cleared:

* Revertive switching: re-activate the candidate path, change O field from 0 to 2 and start sending traffic over it
* Non-revertive switching: do not activate the candidate path, change O field from 0 to 1, keep the second candidate path active with O field set to 2 and continue sending traffic over it
6.3. Restoration

6.3.1. 1+R Restoration

Compared to 1:1 protection described in Section 6.2, this restoration scheme avoids pre-allocating protection bandwidth in steady state, while still being able to recover traffic flow in case of a network failure in a deterministic way (maintain required bandwidth commitment).

The CS-SR policy is configured with two candidate paths. The candidate path with higher preference is established, activated (O field in LSP object is set to 2) and is carrying traffic.

The second candidate path with lower preference is only established and activated (O field in LSP object is set to 2) upon a failure impacting the first candidate path in order to send traffic over an alternate path through the network around the failure with potentially relaxed constraints but still satisfying the bandwidth commitment.

The second candidate path is generally only requested from the PCE and activated after a failure, but may also be requested and pre-established during CS-SR policy creation with the downside of bandwidth being set aside ahead of time.

As soon as failure(s) that brought the first candidate path down are cleared, the second candidate path is getting deactivated (O field in LSP object is set to 1) or torn down. The first candidate path is activated (O field in LSP object is set to 2) and traffic sent across it.

Restoration and reversion behavior is bidirectional. As described in Section 7.1, both headends use connectivity verification in loopback mode and therefore even in case of unidirectional failures both headends will detect the failure or clearance of the failure and switch traffic away from the failed or to the recovered candidate path.

6.3.2. 1:1+R Restoration

For further resiliency in case of multiple concurrent failures that could affect both candidate paths of 1:1 protection described in Section 6.2, a third candidate path with a preference lower than the other two candidate paths is added to the CS-SR policy.
The third candidate path enables restoration and will generally only be established, activated (O field in LSP object is set to 2) and carry traffic after failure(s) have impacted both the candidate path with highest and second highest preference.

The third candidate path may also be requested and pre-computed already whenever either the first or second candidate path went down due to a failure with the downside of bandwidth being set aside ahead of time.

As soon as failure(s) that brought either the first or second candidate path down are cleared the third candidate path is getting deactivated (O field in LSP object is set to 1), the candidate path that recovered is activated (O field in LSP object is set to 2) and traffic sent across it.

Again restoration and reversion behavior is bidirectional. As described in Section 7.1, both headends use connectivity verification in loopback mode and therefore even in case of unidirectional failures both headends will detect the failure or clearance of the failure and switch traffic away from the failed or to the recovered candidate path.

7. Operations, Administration, and Maintenance (OAM)

7.1. Connectivity Verification

The proper operation of each segment list is validated by both headends using STAMP in loopback measurement mode as described in section 4.2.3 of [I-D.ietf-spring-stamp-srpm].

As the STAMP test packets are including both the segment list of the forward and reverse path, standard segment routing data plane operations will make those packets get switched along the forward path to the tailend and along the reverse path back to the headend.

The headend forms the bidirectional SR Policy association using the procedure described in [I-D.ietf-pce-sr-bidir-path] and receives the information about the reverse segment list from the PCE as described in section 4.5 of [I-D.ietf-pce-multipath]

7.2. Performance Measurement

The same STAMP session is used to estimate round-trip loss as described in section 5 of [I-D.ietf-spring-stamp-srpm].
The same STAMP session used for connectivity verification can be used to measure delay. As loopback mode is used only round-trip delay is measured and one-way has to be derived by dividing the round-trip delay by two.

7.3. Candidate Path Validity Verification

A stateful PCE is in sync with the network topology and the CS-SR Policies provisioned on the headend routers. As described in Section 4 a path must not be automatically recomputed after or optimized for topology changes. However there may be a requirement for a PCE to tear down a path if the path no longer satisfies the original requirements, detected by PCE, such as insufficient bandwidth, diversity constraint no longer met or latency constraint exceeded.

The PCC may measure the actual bandwidth utilization of a CS-SR policy and report it to the PCE in order for the PCE to take an appropriate action if necessary.

For a CS-SR policy configured with multiple candidate paths, a PCC may switch to another candidate path if the PCE decided to tear down the active candidate path.

8. External Commands

8.1. Candidate Path Switchover

It is very common to allow operators to trigger a switch between candidate paths even if no failure is present. I.e. to proactively drain a resource for maintenance purposes. Operator triggered switching between candidate paths is unidirectional and has to be requested on both headends.

8.2. Candidate Path Recomputation

While no automatic re-optimization or pre-computation of CS-SR policy candidate paths is allowed as specified in Section 4, network operators trying to optimize network utilization may explicitly request a candidate path to be re-computed at a certain point in time.

9. Security Considerations

TO BE ADDED
10. IANA Considerations

This document has no IANA actions.

11. Acknowledgements

The author’s want to thank Samuel Sidor, Mike Koldychev, Rakesh Gandhi and Tarek Saad for providing their review comments.

12. Contributors

Contributors’ Addresses

Brent Foster
Cisco Systems, Inc.
Email: brfoster@cisco.com

Bertrand Duvivier
Cisco System, Inc.
Email: bduvivie@cisco.com

Stephane Litkowski
Cisco Systems, Inc.
Email: slitkows@cisco.com

Jie Dong
Huawei Technologies
Email: jie.dong@huawei.com

13. References

13.1. Normative References


13.2. Informative References

[I-D.ietf-lsr-isis-srv6-extensions]

[I-D.ietf-lsr-ospfv3-srv6-extensions]

[I-D.ietf-pce-local-protection-enforcement]

[I-D.ietf-pce-multipath]

[I-D.ietf-pce-segment-routing-ipv6]

[I-D.ietf-pce-segment-routing-policy-cp]
[I-D.ietf-pce-sr-bidir-path]

[I-D.ietf-spring-segment-routing-policy]

[I-D.ietf-spring-stamp-srpm]

[I-D.sidor-pce-circuit-style-pcep-extensions]


Authors’ Addresses

Christian Schmutzer (editor)
Cisco Systems, Inc.
Email: cschmutz@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Email: cfilsfil@cisco.com

Zafar Ali (editor)
Cisco Systems, Inc.
Email: zali@cisco.com

Francois Clad
Cisco Systems, Inc.
Email: fclad@cisco.com

Praveen Maheshwari
Airtel India
Email: Praveen.Maheshwari@airtel.com

Reza Rokui
Ciena
Email: rrokui@ciena.com

Andrew Stone
Nokia
Email: andrew.stone@nokia.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
SRv6 In-situ Active Measurement with IOAM
draft-song-spring-siam-03

Abstract

This draft describes an active measurement method for SRv6 which can support hop-by-hop and end-to-end measurement on any SRv6 path using existing protocols such as IOAM. A packet containing an SRH uses a flag bit to indicate the packet is an active probing packet. The measurement information, such as the IOAM header and data, is encapsulated in UDP payload, indicated by a dedicated port number. The probing packet originates from a segment source node, traverses an arbitrary segment path, and terminates at a segment endpoint node, as configured by the segment list in SRH. Each segment node on the path, when detecting the flag, shall parse the UDP header and the payload. In the case of IOAM, the node shall process the IOAM option conforming to the standard procedures defined in the IOAM documents. The method is compatible with some other SRv6 active measurement proposals and support multiple applications.

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.

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/.
1. Introduction

SRv6 network OAM needs various means to collect data, detect issues, and measure its performance. [I-D.ietf-6man-spring-srv6-oam] provides some mechanisms for SRv6 OAM. Some other general methods for performance measurement such as [RFC8762] can also be applied for SRv6. However, these methods have limited data coverage and measurement capability. More mechanisms should be provided to enrich the OAM coverage.
IOAM [I-D.ietf-ippm-ioam-data] can support extensible hop-by-hop data collection for user traffic. It is beneficial for SRv6 network monitor and measurement. Since it is designed for user-packet measurement, [I-D.ali-spring-ioam-srv6] proposes to encapsulate IOAM in SRH TLV options.

However, with its well-defined structure and functions, IOAM can also be used for active measurement (i.e., in dedicated probing packets without user payload) to fulfill many measurement tasks that are inconvenient or infeasible to be applied on user traffic. For active measurement, we can directly encapsulate the IOAM header and data in the UDP-based probing packet payload. The similar method has been proposed in [I-D.ietf-spring-stamp-srpm] to support STAMP for SRv6 measurement. IOAM is complement to STAMP by providing hop-by-hop measurement capability. The high-level method can be generalized and extended to support other performance measurement protocols under the same framework.

Fully built on exiting protocol components, the SR-based active measurement method using IOAM can support some useful applications. For example, it can be used to support network-wide telemetry coverage by using pre-planned paths [I-D.tian-bupt-inwt-mechanism-policy]; it can be used to actively measure the backup paths for SRv6 traffic engineering; and by setting the path end as the path head in SRH, it can naturally support two-way or round-trip measurement.

2. An Active Measurement Framework for SRv6

As specified by [RFC8754], the Segment Routing Header (SRH) contains an 8-bit "Flags" field. This document defines the following flag bit 'T' to designate the packet as a dedicated probing packet for active measurement.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-
| |T|           |
+-+-+-+-+-+-+-+-+-
```

Figure 1: A Hierarchical Edge Network

The O-bit defined in [I-D.ietf-6man-spring-srv6-oam] servers for user traffic OAM, so the T-bit and O-bit are mutual exclusive. When T-bit is set, O-bit must be cleared, and vice versa.

The Next Header of SRH is set to UDP. A destination UDP port is reserved to encode the type of the payload. For example, a port number has been proposed to be reserved for STAMP in [I-D.ietf-spring-stamp-srpm]. Similarly, another port number should be reserved for IOAM trace option. If the destination port number matches the reserved values, the UDP payload would encapsulate the corresponding protocol header. The source UDP port can be used or ignored depending on each use case. The UDP checksum processing procedure conforms to [RFC6936].

3. SRv6 In-Situ Active Measurement with IOAM

We focus on a specific use case of the framework: using IOAM trace option for hop-by-hop measurement. The IOAM header and data format are specified in [I-D.ietf-ippm-ioam-data]. The complete active probing packet format for IOAM is shown in Figure 2. The source UDP port can be used as sequence number to track the probing packets on a specific SR path.
4. Network Operation

To initiate an IOAM active measurement on a path, the probing packets are generated at the SR source node. The source address is the address of the SR source node and the destination address is the address of first SR segment endpoint node. The SRH lists all the SR segment endpoint nodes for which IOAM data will be collected.
Each SR node on the path including the source node, when detecting the T-flag, in addition to normal SRH processing, will further parse the UDP header and IOAM header, and as directed by the IOAM header, add data to the IOAM node data list.

The last SR segment endpoint node will terminate the probing packet. The collected data can be exported according to the specifications for IOAM data export.

If an SR segment endpoint node on the path is incapable of processing the probing packet, it should ignore the T-flag and continue forwarding the packet. The last SR segment endpoint node MUST be able to process and terminate the probing packets.

5. Applications

This section summarizes a list of applications of the SRv6 In-situ Active Measurement (SIAM) approach.

* The method can be used as an alternative way for applying IOAM on user traffic in SRv6, because the forwarding behavior in SRv6 networks is determined by the SRH. As long as a probing packet has the same SRH as the user packet, the data collected can faithfully reflect the user packet’s forwarding experience along the same path. In this case, in order to collect the on-path data for a specific flow, all we need is to copy the SRH from the flow packet and construct the probing packets. The probing packet rate can match the original flow or arbitrarily configured. The edge of the SR domain must terminate the probing packets to avoid leakage.

* To support SRv6 traffic engineering, some alternative paths may be pre-computed. It is desirable to constantly measure the performance of these paths so the best path can be picked when a flow is swapped. Since each path can be represented by an SRH, we can construct the probing packets with these SRHs to actively measure their status and performance.

* In an SRv6 network, it is easy to conduct round trip measurement by setting the starting node and the end node of a path to the same segment source node, and setting the destination node as an intermediate node on the path.
In order to detect or prevent gray network failures for SLA guarantee, it is necessary to collect network-wide telemetry data to gain full visibility within a SRv6 domain. We can apply the algorithm described in [I-D.tian-bupt-inwt-mechanism-policy] to calculate the minimum number of optimal SR paths to achieve the full coverage, and construct probing packets on these paths.

6. Probing Packet Type Extension

The same framework can support other OAM protocols. In addition to STAMP [I-D.ietf-spring-stamp-srpm], the active probing packets can carry IOAM E2E option header and data [I-D.ietf-ippm-ioam-data], IOAM DEX option header [I-D.ietf-ippm-ioam-direct-export], and other OAM options. It is easy to use different reserved UDP port numbers to differentiate the payload types.

7. Security Considerations

TBD

8. IANA Considerations

An SRH Flag bit ‘T’. The bit position TBD

Optional UDP destination port numbers indicating different IOAM options (TBD)

9. Acknowledgments

We acknowledge the comments and suggestions from Greg Mirsky and Tianran Zhou which help to improve this document.

10. References

10.1. Normative References


10.2. Informative References

[I-D.ali-spring-ioam-srv6]

[I-D.ietf-6man-spring-srv6-oam]

[I-D.ietf-ippm-ioam-data]

[I-D.ietf-ippm-ioam-direct-export]
[I-D.ietf-spring-stamp-srpm]

[I-D.tian-bupt-inwt-mechanism-policy]


Authors’ Addresses

Haoyu Song
Futurewei Technologies
Santa Clara,
United States of America
Email: haoyu.song@futurewei.com

Gyan Mishra
Verizon Inc.
Email: gyan.s.mishra@verizon.com

Tian Pan
BUPT
Email: pan@bupt.edu.cn
Export of Segment Routing IPv6 Information in IP Flow Information Export (IPFIX)
draft-tgraf-opsawg-ipfix-srv6-srh-03

Abstract

This document introduces new IP Flow Information Export (IPFIX) information elements to identify the SRv6 Segment Routing Header dimensions and SRv6 Control Plane Protocol that traffic is being forwarded with.

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 21 September 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.
1. Introduction

A new type of Routing Extension Header called Segment Routing Header (SRH) is defined by [RFC8754] which is used for applying Segment Routing (SR) on the IPv6 data plane.

Three routing protocol extensions, OSPFv3 Extensions [I-D.li-lsr-ospfv3-srv6-extensions], IS-IS Extensions [I-D.ietf-lsr-isis-srv6-extensions], BGP Prefix Segment Identifiers (Prefix-SIDs) [I-D.ietf-bess-srv6-services] and one Path Computation Element Communication Protocol (PCEP) Extension [I-D.ietf-pce-segment-routing-ipv6] have been defined to propagate Segment Identifiers (SIDs) for the IPv6 data plane.

This document defines eight new IPFIX Information Elements (IEs) and two new subregistries within the "IPFIX Information Elements" registry [RFC7012], respectively for the new SRH dimensions and routing protocol and PCEP extensions.

2. IPFIX Information Elements

This section defines and describes the new IPFIX IEs.

ipv6SRHFlags
8-bit flags defined in the SRH.

ipv6SRHTag
16-bit tag field defined in the SRH that marks a packet as part of a class or group of packets sharing the same set of properties.

ipv6SRHSegment
128-bit IPv6 address that represents an SRv6 segment.
ipv6SRHSegmentBasicList

Ordered basicList [RFC6313] of zero or more 128-bit IPv6 addresses in the SRH that represents the SRv6 segment list. The Segment List is encoded starting from the active segment of the SR Policy.

ipv6SRHSegmentListSection

Exposes the SRH Segment List as defined in section 2 of [RFC8754] as series of n octets.

ipv6SRHSegmentsLeft

8-bit unsigned integer defining the number of route segments remaining to reach the end of the segment list.

ipv6SRHSection

Exposes the SRH and its TLV’s as defined in section 2 of [RFC8754] as series of n octets.

ipv6SRHSegmentType

Name of the routing protocol or PCEP extension from where the active SRv6 segment has been learned from.

3. Use Cases

By using ipv6SRHSegmentBasicList(TBD4), ipv6SRHSegmentsLeft (TBD6), ipv6SRHSegmentType(TBD8) and forwardingStatus(89) it is possible to identify

* how many packets are forwarded or dropped
* if dropped, for which reasons,
* identify the control plane protocol which defined the active segment,
* the SRv6 segment list and
* how many SRv6 segments are left.

4. IANA Considerations

This document requests IANA to create new IEs (see table1) and two new subregistries called "IPFIX IPv6 SRH Flags" (table 2) and "IPFIX IPv6 SRH Segment type" (table 3) under the "IPFIX Information Elements" registry [RFC7012] available at [IANA-IPFIX] and assign the following code initial points.
Note to the RFC-Editor:
* Please replace TBD1 - TBD12 with the values allocated by IANA
* Please replace the [RFC-to-be] with the RFC number assigned to this document

4.1.  ipv6SRHFlags

Name: ipv6SRHFlags  ElementID: TBD1  Description: This Information Element identifies the 8-bit flags defined in the SRH. Values for this Information Element are listed in the "IPFIX IPv6 SRH Flags" registry, see [IANA-IPFIX]. Initial values in the registry are defined by the table below. New assignments of values will be administered by IANA and are subject to Expert Review. Abstract Data Type: unsigned8  Data Type Semantics: flags  Reference: [RFC-to-be], RFC8754[RFC8126]. Experts need to check definitions of new values for completeness, accuracy, and redundancy.
Table 2: "IPFIX IPv6 SRH Flags" registry

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-7</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

4.2. ipv6SRHTag

Name: ipv6SRHTag ElementID: TBD2 Description: This Information Element identifies the 16-bit tag field defined in the SRH that marks a packet as part of a class or group of packets sharing the same set of properties. Abstract Data Type: unsigned16 Data Type Semantics: identifier Reference: [RFC-to-be], RFC8754

4.3. ipv6SRHSegment

Name: ipv6SRHSegment ElementID: TBD3 Description: This Information Element identifies the 128-bit IPv6 address that represents an SRv6 segment. Abstract Data Type: ipv6address Data Type Semantics: default Reference: [RFC-to-be], RFC8754

4.4. ipv6SRHSegmentBasicList

Name: ipv6SRHSegmentBasicList ElementID: TBD4 Description: This Information Element identifies the Ordered basicList [RFC6313] of zero or more 128-bit IPv6 addresses in the SRH that represents the SRv6 segment list. The Segment List is encoded starting from the active segment of the SR Policy. Abstract Data Type: basicList Data Type Semantics: list Reference: [RFC-to-be], RFC8754

4.5. ipv6SRHSegmentListSection

Name: ipv6SRHSegmentListSection ElementID: TBD5 Description: Exposes the SRH Segment List as defined in section 2 of Abstract Data Type: octetArray Data Type Semantics: default Reference: [RFC-to-be], RFC8754[RFC8754] as series of n octets.

4.6. ipv6SRHSegmentsLeft

Name: ipv6SRHSegmentsLeft ElementID: TBD6 Description: This Information Element identifies the 8-bit unsigned integer defining the number of route segments remaining to reach the end of the segment list. Abstract Data Type: unsigned8 Data Type Semantics: quantity Reference: [RFC-to-be], RFC8754
4.7. ipv6SRHSection

Name: ipv6SRHSection ElementID: TBD7 Description: This Information Element exposes the SRH and its TLV's as defined in section 2 of Abstract Data Type: octetArray Data Type Semantics: default Reference: [RFC-to-be], RFC8754 as series of n octets.

4.8. ipv6SRHSegmentType

Name: ipv6SRHSegmentType ElementID: TBD8 Description: This Information Element identifies the name of the routing protocol or PCEP extension from where the active SRv6 segment has been learned from. Values for this Information Element are listed in the "IPFIX IPv6 SRH Segment type" registry, see [IANA-IPFIX]. Initial values in the registry are defined by the table below. New assignments of values will be administered by IANA and are subject to Expert Review Abstract Data Type: unsigned8 Data Type Semantics: identifier Reference: [RFC-to-be][RFC8126]. Experts need to check definitions of new values for completeness, accuracy, and redundancy.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD9</td>
<td>Unknown</td>
<td>[RFC-to-be]</td>
</tr>
<tr>
<td>TBD10</td>
<td>Path Computation Element</td>
<td>[RFC-to-be], draft-ietf-pce-segment-routing-ipv6</td>
</tr>
<tr>
<td>TB11</td>
<td>OSPFv3 Segment Routing</td>
<td>[RFC-to-be], draft-li-ospf-ospfv3-srv6-extensions</td>
</tr>
<tr>
<td>TBD12</td>
<td>IS-IS Segment Routing</td>
<td>[RFC-to-be], draft-ietf-lsr-isis-srv6-extensions</td>
</tr>
<tr>
<td>TBD13</td>
<td>BGP Segment Routing Prefix-SID</td>
<td>[RFC-to-be], draft-ietf-bess-srv6-services</td>
</tr>
</tbody>
</table>

Table 3: "IPFIX IPv6 SRH Segment type" subregistry

5. Operational Considerations

The zero or more 128-bit IPv6 addresses in the SRH [RFC8754] can be exported in two different ways, with two different IPFIX IEs:

* ipv6SRHSegmentBasicList
The ipv6SRHSegmentBasicList encodes the SID list of IPv6 addresses with a basicList, specified in the IPFIX Structured Data [RFC6313]. This encoding offers the advantage to the data collection that the different IPv6 addresses are already structured as a list, without the need of post processing. However, this method requires some extra processing on the exporter, to realize the BasicList data mapping.

The ipv6SRHSegmentListSection, on the other hand, encodes the list of IPv6 addresses as an octetArray. This doesn’t impose any data flow manipulation on the exporter, facilitating the immediate export. However, the data collection must be able to decode the IPv6 addresses according the SR specifications.

It is not expected that an exporter would support both ipv6SRHSegmentBasicList and ipv6SRHSegmentListSection at the same time.

6. Security Considerations

There exists no significant extra security considerations regarding the allocation of these new IPFIX IEs compared to [RFC7012].

7. Acknowledgements

I would like to thank Pierre Francois, Yao Liu and Paolo Lucente for their review and valuable comments.

8. References

8.1. Normative References


8.2. Informative References

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R.,
Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based
Overlay Services", Work in Progress, Internet-Draft,
draft-ietf-bess-srv6-services-13, 19 March 2022,
<https://www.ietf.org/archive/id/draft-ietf-bess-srv6-
services-13.txt>.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and
Z. Hu, "IS-IS Extensions to Support Segment Routing over
IPv6 Dataplane", Work in Progress, Internet-Draft, draft-
ietf-lsr-isis-srv6-extensions-18, 20 October 2021,
<https://www.ietf.org/archive/id/draft-ietf-lsr-isis-srv6-
extensions-18.txt>.

[I-D.ietf-pce-segment-routing-ipv6]
Li, C., Negi, M., Sivabalan, S., Koldychev, M.,
Kaladharan, P., and Y. Zhu, "PCEP Extensions for Segment
Routing leveraging the IPv6 data plane", Work in Progress,
Internet-Draft, draft-ietf-pce-segment-routing-ipv6-12, 6
March 2022, <https://www.ietf.org/internet-drafts/draft-
ietf-pce-segment-routing-ipv6-12.txt>.

[I-D.li-lsr-ospfv3-srv6-extensions]
Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak,
"OSPFv3 Extensions for SRv6", Work in Progress, Internet-
Draft, draft-li-lsr-ospfv3-srv6-extensions-00, 15 January
2020, <https://www.ietf.org/archive/id/draft-li-lsr-
ospfv3-srv6-extensions-00.txt>.

[IANA-IPFIX]
"IANA, "IP Flow Information Export (IPFIX) Entities"",
<https://www.iana.org/assignments/ipfix/ipfix.xhtml>.

Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header
(SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020,

Appendix A. IPFIX Encoding Example

In this section an example is provided to show the encoding format
for the newly introduced IEs.
Table 4: 3 observed SRH headers and their routing protocol

A.1. Template Record

<table>
<thead>
<tr>
<th>SRH Nr</th>
<th>SRH Flags</th>
<th>SRH Tag</th>
<th>Segment Type</th>
<th>Segment List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>123</td>
<td>IS-IS</td>
<td>2001:db8::1, 2001:db8::2, 2001:db8::3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>456</td>
<td>IS-IS</td>
<td>2001:db8::4, 2001:db8::5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>789</td>
<td>IS-IS</td>
<td>2001:db8::6</td>
</tr>
</tbody>
</table>

Table 4: Template Record Encoding Format

In this example, the Template ID is 256, which will be used in the Data Record. The field length for ipv6SRHSegmentBasicList is 0xFFFF, which means the length of this IE is variable, and the actual length of this IE is indicated by the List Length field in the basicList format as per [RFC6313].

A.2. Data Set

The data set is represented as follows:
Table 5: Data Set Encoding Format

Authors’ Addresses