SPRING
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

S. Agrawal, Ed.
Z. Ali
C. Filsfils
Cisco Systems
D. Voyer
Bell Canada
G. Dawra
LinkedIn
Z. Li
Huawei Technologies
February 22, 2021

SRv6 and MPLS interworking
draft-agrawal-spring-srv6-mpls-interworking-05

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 August 26, 2021.

Table of Contents

1. Introduction

Many of the deployments require SRv6 insertion in the brownfield networks. The incremental deployment of SRv6 into existing networks require SRv6 to interwork and co-exist with SR-MPLS/MPLS. This document discusses solutions for the various interworking scenarios.

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.

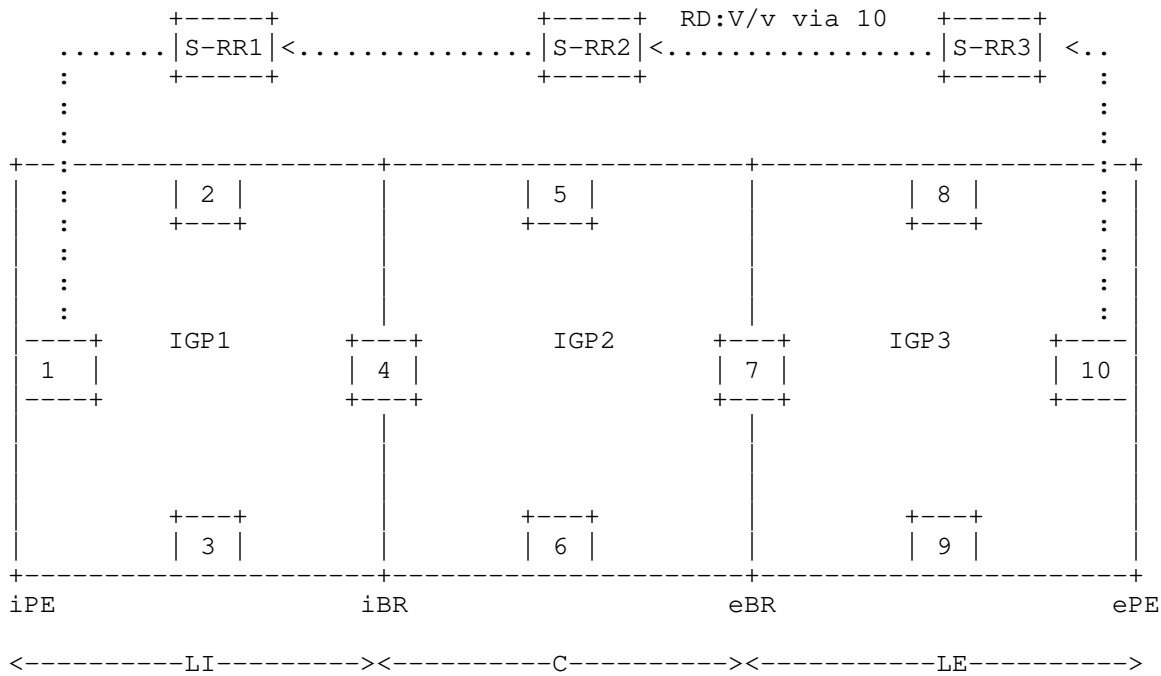


Figure 1: Reference multi-domain network topology

Document assumes SR-MPLS-IPv4 for MPLS data plane. Note: Procedures in the document equally work for SR-MPLS-IPv6, LDP-IPv4/IPv6 and RSVP-TE-MPLS.

2.1. IW scenarios

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.

2.1.1. Transport IW

L3/L2 service continuity over a different intermediate transport.

- SRv6 over SR-MPLS-IPv4 (6oM)
  - LI and LE domains are SRv6 data plane, C is SR-MPLS-IPv4 data plane

* 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 destination address(DA) is the SRv6 Service SID.

* Tunnel traffic destined to egress PE SRv6 locator over SR-MPLS-IPv4 C domain.

o SR-MPLS-IPv4 over SRv6 (Mo6)

* LI and LE domains are SR-MPLS-IPv4 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 next hop.

* Tunnel MPLS LSP to egress PE next hop over SRv6 C domain.

2.1.2. Service IW

Service discontinuity over a different intermediate transport i.e. L2/L3 BGP SRv6 PE interworking with L2/L3 BGP MPLS PE for service connectivity.

o SRv6 to SR-MPLS-IPv4 (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 SR-MPLS-IPv4 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 and S3 is the last SID to traverse. (S3, S2, S1; SL) represents the same SID list but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID. When referring to an SR policy in a high-level use-case, it is simpler to use the <S1, S2, S3> notation. When referring to an illustration of the detailed packet behavior, the (S3, S2, S1; SL) notation is more convenient.

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. }
```

Note: IANA has allocated the Internet Protocol number 137 [RFC4023] for MPLS-in-IP.

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].

5.2. H.Encaps.M.Red: H.Encaps.Red applied to MPLS label stack

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. 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 need to support single data plane type both for 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

6.1. Transport IW

As described in Section 2.1.1, transport IW requires:

- Tunnel traffic destined to SRv6 Service SID bound to SRv6 locator of egress PE over SR-MPLS-IPv4 C domain.
- Tunnel MPLS LSP bound to IPv4 loopback address of egress PE over SRv6 C domain.

This draft enhances two well-known solutions to achieve above tunneling: a controller(SR-PCE) and BGP inter domain routing based approach. The SR-PCE based solution is applicable to both best effort as well as deployments where intents are required (e.g., On-Demand Next-hop like deployments scenarios) by L3/L2 services. The BGP signaling covers the best effort case.

Specifically, the draft proposes the following two ways:

- 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. These procedures can be used when overlay prefixes are signaled with a color extended community [I-D.ietf-idr-tunnel-encaps].

- o BGP Inter-Domain routing procedures advertising PE locator/IPv4 Loopback address for best effort end to end connectivity. These procedures can be used when overlay prefixes don't have color extended community.

6.1.1. 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. A Color is a 32-bit numerical value that associates an SR Policy with an intent [I-D.ietf-spring-segment-routing-policy]. In this case, based on the intent, the PCE computes and programs end to end path using SR-Policy(C,PE). The PCE is also aware of interworking requirement at border nodes, as each domain feeds topological information to the PCE through BGP LS feeds. Intermediate domain of different data plane type is represented by Binding SID (BSID) [RFC8402] of ingress domain type in SID list. In summary, an intermediate domain of different data plane is replaced by a BSID of the data plane nature of headend.

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

6.1.1.1. 6oM

Refer Section 2.1.1 for 6oM scenario. Service prefix (e.g. VPN or EVPN) is received on head-end(node 1) with color extended community(C1) from egress PE(node 10) and SRv6 service SID. Head-end does not know how to compute the traffic engineered path through the multi-domain network to node 10. Node 1 requests SR-PCE to compute a path to node 10 providing intent (e.g. low latency). The PCE computes low latency path via node 2, 5 and 8. The PCE identifies the end-to-end path is not consistent data plane and kicks in interworking procedures at the border router(node 4). It programs a SR policy with MPLS segment list at 4 along required SLA path(node 5 and 7) bounded to an End.BM BSID [I-D.ietf-spring-srv6-network-programming]. 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 rout4er) 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.

6.1.1.2. 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). Head-end does not know how to compute the traffic engineered path through the multi-domain network to node 10. Node 1 requests SR-PCE to compute a path to node 10 providing intent(eg: low latency). The PCE computes low latency path via node 2, 5 and 8. The PCE identifies the end-to-end path is not consistent data plane and kicks in interworking procedures at the border router(node 4). It programs a SRv6 policy bound to MPLS BSID at node 4 with SRv6 SID segment list along required SLA path with last segment of behavior End.DTM. End.DTM behavior decapsulates the IPv6 header and looks up top MPLS label in MPLS table. SR-PCE responds back to node 1 with MPLS segment list along required SLA path 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)

4. Node 7 receive IPv6 packet with DA=B:7:DTM::. It performs DTM behavior to remove IPv6 header and perform 16008 lookup in MPLS table.
   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.

6.1.2. BGP inter domain routing procedures

BGP 3107 [I-D.ietf-mpls-seamless-mpls] like procedures to advertise PE locators and IPv4 loopbacks transport reachability in multi-domain network with next hop self on border routers.

Below sections describe 6oM and Mo6 IW with BGP procedures

6.1.2.1. 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.

- 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.

- 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.

- 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

- 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 i n the VRF

6.1.2.2. 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.

BGP LU [RFC8277] advertise IPv4 PE loopbacks. Next hop self-performed on border routers.

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

6.1.2.2.1. Tunnel BGP LU LSP across SRv6 C domain

Intuitive solution for an MPLS-minded operator

- Existing BGP-LU label cross-connect on border routers for each PE IPv4 loopback address.
- The lookups at the ingress border router are based on BGP3107 label as usual
- 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.
- Ingress border router forwarding perform 3107 label swap and H.Encaps.M with DA = SRv6 SID associated with DTM behavior
- 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

6.1.2.2.1.1. 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.

- 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 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.

- "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

- 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 implementation to keep operating(legacy ingress border routers).

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 LI domain.

* BGP learns node10 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

6.1.2.2.2. Label and SRv6 SID cross connect for BGP LU route

- Allocate SRv6 SID associated with behavior that is decap variant of End.BM in [I-D.ietf-spring-srv6-network-programming] 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 performs pop and H.Encaps.M.Red with DA=per PE SRv6 SID on receiving packet with local label

BGP protocol extension will be detailed in future version.

6.2. 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.

6.2.1. 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) [I-D.ietf-spring-srv6-network-programming]. 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.

6.2.2. 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.

7. 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

8. Availability

- Failure within domain are taken care by existing FRR mechanisms [I-D.ietf-rtgwg-segment-routing-ti-lfa].
- Procedures listed in [I-D.ietf-spring-segment-routing-policy] provides protection in SR-PCE multi-domain On Demand Nexthop (ODN) SR policy based approach.
- 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.

9. IANA Considerations

9.1. BGP Prefix-SID TLV Types registry

This document introduce 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

Value	Type	Reference
TBD	SRv6 label route tunnel TLV	<this document>

9.2. 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.

Value	Hex	Endpoint behavior	Reference
TBD	TBD	End.DTM	<this document>

10. Security Considerations

11. Acknowledgements

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

12. References

12.1. Normative 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", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <https://www.rfc-editor.org/info/rfc4023>.

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

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

12.2. Informative References

[I-D.ietf-idr-tunnel-encaps] Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-21 (work in progress), January 2021.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

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

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2021

Z. Ali
C. Filsfils
P. Camarillo
Cisco Systems
D. Voyer
Bell Canada
S. Matsushima
Softbank

February 21, 2021

Building blocks for Slicing in Segment Routing Network
draft-ali-spring-network-slicing-building-blocks-04.txt

Abstract

This document describes how to build network slice using the Segment Routing based technology. It explains how the building blocks specified for the Segment Routing can be used for this purpose.

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 August 21, 2021.

Table of Contents

1 Introduction

As more and more Service Providers and Enterprises operate a single network infrastructure to support an ever-increasing number of services, the ability to custom fit transport to application needs is critically important. This includes creating network slices with different characteristics can coexist on top of the shared network infrastructure.

Network Slicing is meant to create (end-to-end) partitioned network infrastructure that can be used to provide differentiated connectivity behaviors to fulfill the requirements of a diverse set of services. Services belonging to different Network slices can be wholly disjoint or can share different parts of the network infrastructure. Network Slicing is one of the requirements in 5G [3GPP 23501].

Segment Routing enables Service Providers to support Network Slicing without any additional protocol, other than the SR IGP extensions. The network as a whole, in a distributed and entirely automated manner, can share a single infrastructure resource along multiple virtual services (slices). For example, one slice is optimized continuously for low-cost transport; a second Slice is optimized continuously for low-latency transport; a third Slice is orchestrated to support disjoint

services, etc. The optimization objective of each of these slices is programmable by the operator.

The Segment Routing specification already contains the various building blocks required to create network slices. This includes the following.

- . SR Policy with or without Flexible Algorithm.
- . TI-LFA with O(50 msec) protection in the slice underlay.
- . SR VPN.
- . SR Service Programming (NFV, SFC).
- . Operation, Administration and Management (OAM) and Performance Management (PM).
- . QoS using DiffServ.
- . Stateless Network Slice Identification
- . Orchestration at the Controller.

Each of these building blocks works independently of each other. Their functionality can be combined to satisfy service provider's requirement for the Network Slicing. An external controller plays an important role to orchestrate these building blocks into a Slicing service.

This document elaborates on the attributes of each of these building blocks for Network Slicing. The document also highlights how services in each Slice can benefit from traffic engineering, network function virtualization/ service chaining (service programming), OAM, performance management, SDN readiness, O (50 msec) TI-LFA protection, etc. features of SR while respecting resource partitioning employed over the common networking infrastructure.

The document equally applicable to the MPLS and SRv6 instantiations of segment routing.

The following subsection elaborates on each of these build blocks.

2 Segment Routing Policy

Segment Routing (SR) allows a headend node to steer a packet flow along any path without creating intermediate per-flow states [I-D.ietf-spring-segment-routing-policy]. The headend node steers a flow into a Segment Routing Policy (SR Policy). I.e., the SR Policy can be used to steer traffic along any arbitrary path in the network. This allows operators to enforce low-latency and / or disjoint paths, regardless of the normal forwarding paths.

The SR policy is able to support various optimization objectives [I-D.draft-filsfils-spring-sr-policy-considerations]. The optimization objectives can be instantiated for the IGP metric ([RFC1195] [RFC2328] [RFC5340]) xor the TE metric ([RFC5305], [RFC3630]) xor the latency extended TE metric ([RFC7810] [RFC7471]). In addition, an SR policy is able to various constraints, including inclusion and/or exclusion of TE affinity, inclusion and/or exclusion of IP address, inclusion and/or exclusion of SRLG, inclusion and/or exclusion of admin-tag, maximum accumulated metric (IGP, TE, and latency), maximum number of SIDs in the solution SID-List, maximum number of weighted SID-Lists in the solution set, diversity to another service instance (e.g., link, node, or SRLG disjoint paths originating from different head-ends), etc. [I-D.draft-filsfils-spring-sr-policy-considerations]. The supports for various optimization objectives and constraints enables SR policy to create Slices in the network.

SR policy can be instantiated with or without IGP Flexible Algorithm feature. The following subsection describes the SR Flexible Algorithm feature and how SR policy can utilize this feature.

2.1 Flex-Algorithm

Flexible Algorithm enriches the SR Policy solution by adding additional segments having different properties than the IGP Prefix segments. Flex Algo adds flexible, user-defined segments to the SRTE toolbox. Specifically, it allows for association of the "intent" to Prefix SIDs. [I-D.ietf-lsr-flex-algo] defines the IGP based Flex-Algorithm
solution which allows IGPs themselves to compute paths constraint by the "intent" represented by the Flex-Algorithm.

The Flex-Algorithm has the following attributes:

- Algorithm associate to the SID a specific TE intent expressed as an optimization objective (an algorithm) [I-D.ietf-lsr-flex-algo].
- Flexibility includes the ability of network operators to define the intent of each algorithm they implement.
- By design the mapping between the Flex-Algorithm and its meaning is flexible and is defined by the user.
- Flexibility also includes ability for operators to make the decision to exclude some specific links from the shortest path computation, e.g.,

- o operator 1 may define Algo 128 to compute the shortest path for TE metric and exclude red affinity links.
- o operator 2 may define Algo 128 to compute the shortest path for latency metric and exclude blue affinity links.

A Network Slice can be created by associating a Flexible-Algorithm value with the Slice via provisioning.

Flex Alg leverages SR on-demand next hop (ODN) and Automated Steering for intent-based instantiation of traffic engineered paths described in the following sub-sections. Specifically, as specified in [RFC8402] the IGP Flex Algo
Prefix SIDs can also be used as segments within SR Policies thereby leveraging the underlying IGP Flex Algo solution.

2.2 On-demand SR policy

Segment Routing On-Demand Next-hop (ODN) functionality enables on-demand creation of SR Policies for service traffic. Using a Path Computation Element (PCE), end-to-end SR Policy paths can be computed to provide end-to-end Segment Routing connectivity, even in multi-domain networks running with or without IGP Flexible-Algorithm [I-D.draft-ietf-spring-segment-routing-policy].

The On-Demand Next-hop functionality provides optimized service paths to meet customer and application SLAs (such as latency, disjointness) without any pre-configured TE tunnel and with the automatic steering of the service traffic on the SR Policy without a static route, autoroute-announce, or policy-based routing.

With this functionality, a Network Service Orchestrator can deploy the service based on their requirements. The service head-end router requests the PCE to compute the path for the service and then instantiates an SR Policy with the computed path and steers the service traffic into that SR Policy. If the topology changes, the stateful PCE updates the SR Policy path. This happens seamlessly, while TI-LFA protects the traffic in case the topology change happened due to a failure.

2.3 Automatic Steering

Automatically steering traffic into a Network Slice is one of the fundamental requirement for Slicing. That is made possible by the "Automated Steering" functionality of SR. Specifically, SR policy can be used for traffic engineer paths within a slice, "automatically steer" traffic to the right slice and connect IGP Flex-Algorithm domains sharing the same "intent".

A headend can steer a packet flow into a valid SR Policy within a slice in various ways [I-D.draft-ietf-spring-segment-routing-policy]:

- Incoming packets have an active SID matching a local Binding SID (BSID) at the headend.
- Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.
- Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

2.4 Inter-domain Considerations

The network slicing needs to be extended across multiple domains such that each domain can satisfy the intent consistently. SR has native inter-domain mechanisms, e.g., SR policies are designed to span multiple domains using a PCE based solution [I-D.ietf-spring-segment-routing], [I-D.ietf-spring-segment-routing-central-epe]. An edge router upon service configuration automatically requests to the Segment Routing PCE an inter-domain path to the remote service endpoint. The path can either be for simple best-effort inter-domain reachability or for reachability with an SLA contract and can be restricted to a Network Slice.

The SR native mechanisms for inter-domain are easily extendable to include the case when different IGP Flex-Algorithm values are used to represent the same intent. E.g., in domain1 Service Provider 1 (SP1) may use flex-algo 128 to indicate low latency Slice and in domain2 Service Provider 2 (SP2) may use flex-algo 129 to indicate low latency Slice. When an automation system at a PE1 in SP1 network configures a service with next hop (PE2) in SP2 network, SP1 contacts a Path Computation Element (PCE) to find a route to PE2. In the request, the PE1 also indicates the intent (i.e., the Flex-Algo 128) in the PCEP message. As the PCE has a complete understanding of both Domains, it can understand the path computation in Domain1 needs to be performed for Algorithm 128 and path computation in Domain2 needs to be

performed for Algorithm 129 (i.e., in the Low Latency Network Slice in both domains).

3 TI-LFA and Microloop Avoidance

The Segment Routing-based fast-reroute solution, TI-LFA, can provide per-destination sub-50msec protection upon any single link, node or SRLG failure regardless of the topology. The traffic is rerouted straight to the post-convergence path, hence avoiding any intermediate flap via an intermediate path. The primary and backup path computation is completely automatic by the IGP.

[I-D.draft-bashandy-rtgwg-segment-routing-ti-lfa] proposes a Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at protecting node and adjacency segments within O(50 msec) in the Segment Routing networks. Furthermore, [I-D. draft-bashandy-rtgwg-segment-routing-uloop] provides a mechanism leveraging Segment Routing to ensure loop-freeness during the IGP reconvergence process following a link-state change event.

As mentioned earlier, Network Slicing in Segment Routing works seamlessly with all the other components of the Segment Routing. This, of course, includes TI-LFA and microloop avoidance within a Slice, with the added benefit that backup path only uses resources available to the Slice. For example, when Flexible Algorithm is used, the TI-LFA backup path computation is performed such that it is optimized per Flexible-Algorithm. The backup path shares the same properties are the primary path. The backup path does not use a resource outside the Slice of the primary path it is protecting.

4 SR VPN

Virtual Private Networks (VPNs) provides a mean for creating a logically separated network to a different set of users access to a common network. Segment Routing is equipped with the rich multi-service virtual private network (VPN) capabilities, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN). The ability of Segment Routing to support different VPN technologies is one of the fundamental building blocks for creating slicing an SR network.

5 Stateless Service Programming

An important part of the Network Slicing is the orchestration of virtualized service containers. [I-D.draft-xuclad-spring-sr-service-chaining] describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. It introduces the notion of service segments. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of the service, running either on a physical appliance or in a virtual environment, is associated with a segment identifier (SID) for the service. These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined with topological SIDs to achieve service programming while steering the traffic through a specific topological path in the network. In this fashion, SR provides a fully integrated solution for overlay, underlay and service programming building blocks needed to satisfy network slicing requirements.

6 Operations, Administration, and Maintenance (OAM)

There are various OAM elements that are critical to satisfy Network Slicing requirements. These includes but not limited to the following:

- Measuring per-link TE Matric.
- Flooding per-link TE Matric.
- Taking TE Matric into account during path calculation.
- Taking TE Matric bound into account during path calculation.
- SLA Monitoring: Service Provider can monitor each SR Policy in a Slice to Monitor SLA offered by the Policy using technique described in [I-D.draft-gandhi-spring-udp-pm]. This includes monitoring end-to-end delays on all ECMP paths of the Policy as well as monitoring traffic loss on a Policy. Remedial mechanisms can be used to ensure that the SR policy conforms to the SLA contract.

7 QoS

Segment Routing relies on MPLS and IP Differentiated Services. Differentiated services enhancements are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. [RFC2475] defines

an architecture for implementing scalable service differentiation in the Internet. This architecture is composed of many functional elements implemented in network nodes, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions including metering, marking, shaping, and policing.

The DiffServ architecture achieves scalability by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviors to aggregates of traffic depending on the traffic marker. Specifically, the node at the ingress of the DiffServ domain conditions, classifies and marks the traffic into a limited number of traffic classes. The function is used to ensure that the slice's traffic conforms to the contract associated with the slice.

Per-hop behaviors are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams. Specifically, per class scheduling and queuing control mechanisms are applied at each IP hop to the traffic classes depending on packet's marking. Techniques such as queue management and a variety of scheduling mechanisms are used to get the required packet behavior to meet the slice's SLA.

8 Stateless Network Slice Identification

Some use-cases require a slice identifier (SLID) in the packet to provide differentiated treatment of the packets belonging to different network slices.

The network slice instantiation using the SLID in the packet is required to work with the building blocks described in the previous sections. For example, the QoS/ DiffServ needs to be observed on a per slice basis. The slice identification needs to be topologically independent and stateless.

8.1 Stateless Slice Identification in SRv6

[I-D.draft-filsfils-spring-srv6-stateless-slice-id] describes a stateless encoding of slice identification in the outer IPv6 the header of an SRv6 domain. As defined in RFC8754 [RFC8754], when an ingress PE receives a packet that traverses the SR domain, it encapsulates the packet in an outer IPv6 header and an optional SRH. Based on a local policy of the SR domain, the Flow Label field of the outer IPv6 header carries the SLID. Specifically, the SLID is added in the 8 most significant bits of the Flow Label field of the outer IPv6 header. The remaining 12 bits of the Flow Label field are set as described in section 5.5 of [RFC8754] for inter-domain packets. Based on the local policy of the SR domain, the draft also uses one of the bits in the Traffic Class field of the outer IPv6 header to indicate that the entropy label contains the SLID.

The network slicing mechanism described in [I-D.draft-filsfils-spring-srv6-stateless-slice-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SRv6 overlays.

8.2 Stateless Slice Identification in SR-MPLS

[I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] describes a similar stateless encoding of slice identification in the SR-MPLS domain. Specifically, the document extends the use of the Entropy Label to carry the SLID. The number of bits to be used for encoding the SLID in the Entropy Label is governed by a local policy of the SR domain. Based on the local policy of the SR domain, the draft uses one of the bits in the TTL field of the Entropy Label to indicate that the Entropy Label contains the SLID.

The network slicing mechanism described in [I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SR-MPLS overlays.

8 Orchestration at the Controller

A controller plays a vital role in orchestrating the SR building blocks discussed above to create Network Slices. The controller also performs admission control and traffic placement for slice management at the transport layer. The SDN friendliness of the SR technology becomes handy to realize the orchestration. The controller may use PCEP or Netconf to interact with the routers. The router implements Yang model for SR-based network slicing.

Specification of the controller technology for orchestrating Network Slices, services and admission control for the services is outside the scope of this draft.

9 Illustration

To be added in a later revision.

10 Security Considerations

This document does not impose any additional security challenges.

11 IANA Considerations

This document does not define any new protocol or any extension to an existing protocol.

12 References

12.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

7.2. Informative References

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, et al, "Segment Routing Policy For Traffic Engineering", draft-ietf-spring-segment-routing-policy (work in progress).

[I-D.ietf-lsr-flex-algo] P. Psenak, et al, draft-ietf-lsr-flex-algo, work in progress.

[RFC8402] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC8402.

[I-D.draft-filsfils-spring-sr-policy-considerations] Filsfils, C., et al. draft-filsfils-spring-sr-policy-considerations (work in progress)

[RFC8754] Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.

[I-D.draft-filsfils-spring-srv6-stateless-slice-id] Filsfils, C., et al. draft-filsfils-spring-srv6-stateless-slice-id, work in progress.

[I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] Decraene B., Filsfils, C., Henderickx W., Saad T., Beeram V., work in progress.

13 Acknowledgments

14 Contributors

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

Authors' Addresses

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

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

Pablo Camarillo Garvia
Cisco Systems, Inc.
Email: pcamaril@cisco.com

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

spring
Internet-Draft
Intended status: Standards Track
Expires: August 22, 2021

Z. Ali
C. Filsfils
N. Nainar
C. Pignataro
F. Clad
Cisco Systems, Inc.
F. Iqbal
Arista Networks
X. Xu
Alibaba

February 22, 2021

OAM for Service Programming with Segment Routing
draft-ali-spring-sr-service-programming-oam-03

Abstract

This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

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 August 22, 2021.

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-spring-sr-service-programming] defines data plane functionality required to implement service segments and achieve service programming in SR-enabled MPLS and IP networks, as described in the Segment Routing architecture. This document defines the Operations, Administrations and Maintenance (OAM) for service programming in SR-enabled MPLS and IP networks.

2. Requirements notation

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].

3. Terminology

This document uses the terminologies defined in [RFC8402], [I-D.ietf-spring-srv6-network-programming] [I-D.ietf-spring-sr-service-programming] and so the readers are expected to be familiar with the same.

4. Document Scope

The initial focus of this document to define and document the machinery required to apply OAM mechanisms on SRv6 based service programming.

Future version of this document will include the required details to apply OAM mechanism on other data planes.

5. OAM for Service Programming

Section 4 of [I-D.ietf-spring-sr-service-programming] introduces Service segments and the procedure of service programming when the services are SR-aware and SR-unaware. By integrating the OAM functionality in the services, versatile OAM tool kits can be used to execute programmable OAM for service programming with Segment Routing.

This section describes the procedure to perform basic OAM mechanisms such as ping and path tracing to Service Programming environment in Segment Routing network.

5.1. Service Programming OAM Packet Processing

Any services upon receiving OAM packet may apply the service treatment if it cannot differentiate the OAM packet from normal data packet. Depending on the service type, service treatment on OAM packet may result in dropping the OAM probe packet that may cause uncertainty in OAM mechanism.

The pseudo code for the service function SIDs in [I-D.ietf-spring-sr-service-programming] has been defined to avoid such uncertainty, as explained in the following subsections.

5.2. Service Programming OAM in SRv6 Data Plane

When the service programming is applied in an SRv6 network, the Upper-layer header type is typically set to ICMPv6 or UDP to differentiate the OAM packet from the data packets.

5.2.1. OAM with SR-aware services

As defined in section 4.1 of [I-D.ietf-spring-sr-service-programming], an SR-aware service can process the SR information in the packet header such as performing lookup or executing the next segment, processing the upper layer header, etc.

An SR-aware service SHOULD skip applying the service on the OAM. As defined in section 9, a local policy may be used to control any malicious use of OAM marker.

An SR-aware service follows the procedure defined in the [I-D.draft-ietf-6man-spring-srv6-oam] to implement ping and trace-route to a SR-aware SID and additional OAM mechanisms including the support for the OAM flag (O-flag).

5.2.2. OAM with SR-unaware services

As defined in section 4.2 of [I-D.ietf-spring-sr-service-programming], an SR-unaware service may be a legacy service that is not able to process the SR information in the packet header. SR Proxy, an entity that is external to the service is used to handle the SR information processing on behalf of the service. SR Proxy will remove the SR header before forwarding the packet to SR-unaware services to avoid any erroneous decision due to the presence of SR header that the service cannot recognize.

The SRv6 pseudocode for SR Proxy defined in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming] handles the OAM packets as explained in the following.

- Case 1: The service service programming segment is a transit segment. In this case, if the Upper-layer header does not match Ethernet, IPv4 or IPv6, the service function is skipped and packet is resubmitted to the IPv6 module for transmission to the new destination in the header (towards the next SRv6 segment).

  Please refer to the following lines of SRv6 pseudocode for SR Proxy defined in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming], respectively.

  In case of Static Proxy for Inner Type Ethernet:
  S15. If (Upper-layer header type != 143 (Ethernet)) {
  S16. Resubmit the packet to the IPv6 module for transmission to the new destination.
  S17. }

  In case of Static Proxy for Inner Type IPv4:
  S15. If (Upper-layer header type != 4 (IPv4)) {
  S16. Resubmit the packet to the IPv6 module for transmission to the new destination.
  S17. }

  In case of Static Proxy for Inner Type IPv6:
  S15. If (Upper-layer header type != 41 (IPv6)) {
  S16. Resubmit the packet to the IPv6 module for transmission to the new destination.
  S17. }

- Case 2: The service service programming segment is the ultimate segment. This is the case of OAM operations are targetted to a service programming SID (e.g., Ping and Trace-route to a service programming SID). In this case, as part of the Upper-layer header processing, the SR proxy processes to OAM payload, skips applying the service on the OAM packet and responds to the OAM message, accordingly.

  Please refer to the following lines of SRv6 pseudocode for SR Proxy defined in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming], respectively.

  In case of Static Proxy for Inner Type Ethernet:
  When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for Ethernet traffic, the following pseudocode is executed.

  S01. If (Upper-layer header type != 143 (Ethernet)) {
  S02. Process as per [I-D.ietf-spring-srv6-network-programming] Section 4.1.1
  S03. }

  In case of Static Proxy for Inner Type IPv4:
  When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv4 traffic, the following pseudocode is executed.
  S01. If (Upper-layer header type != 4 (IPv4)) {

```
S02.   Process as per [I-D.ietf-spring-srv6-network-programming]
       Section 4.1.1
S03. }
```

In case of Static Proxy for Inner Type IPv6:
When processing the Upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 static proxy SID for IPv6 traffic, the following pseudocode is executed.

```
S01. If (Upper-layer header type != 41 (IPv6)) {
S02.   Process as per [I-D.ietf-spring-srv6-network-programming]
       Section 4.1.1
S03. }
```

5.3. Service Programming OAM in SR-MPLS Data Plane

This section will be updated later.

5.4. Controlling OAM packet processing in Services

As mentioned in the above sections, SR-aware service or the SR proxy can use the Upper-layer header to differentiate the OAM packet from data packet to skip the service treatment. To avoid any intentional or unintentional use of OAM, a local policy SHOULD be used in the SR-aware service or SR Proxy to rate limit the incoming OAM packets.

6. Illustration

This section illustrates how the existing OAM tools can be used to perform the connectivity check or path tracing of SR Service Policies.

6.1. SRv6 Dataplane

This section illustrates how ICMPv6 can be used to ping or trace SR service policies in an SRv6 network using the below example topology.

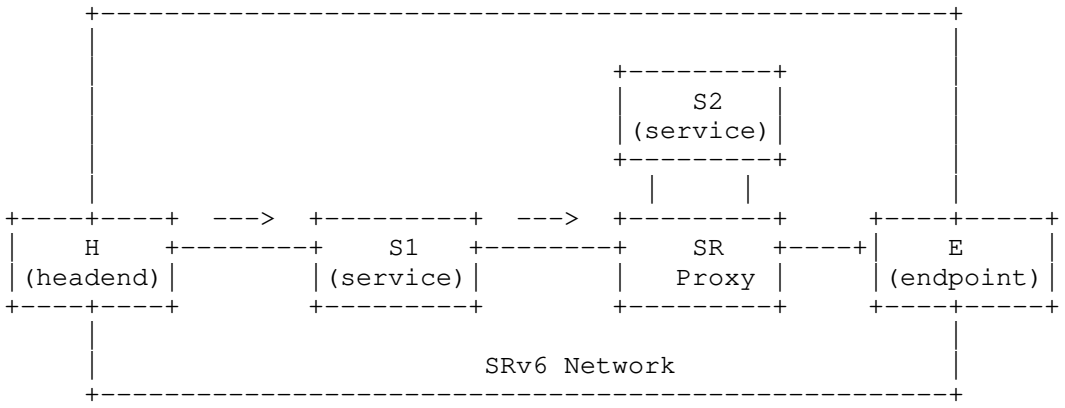


Figure 2. SR Service Policies in SRv6 Network

6.1.1. Pinging SR Service Policy

The user interested to ping the SR service policy shown in Figure 2 will trigger the ICMPv6 echo request from the headend H with IP6(H,S1)(SRH) and the upper layer header set to ICMPv6. The probe will be processed along the path as below:

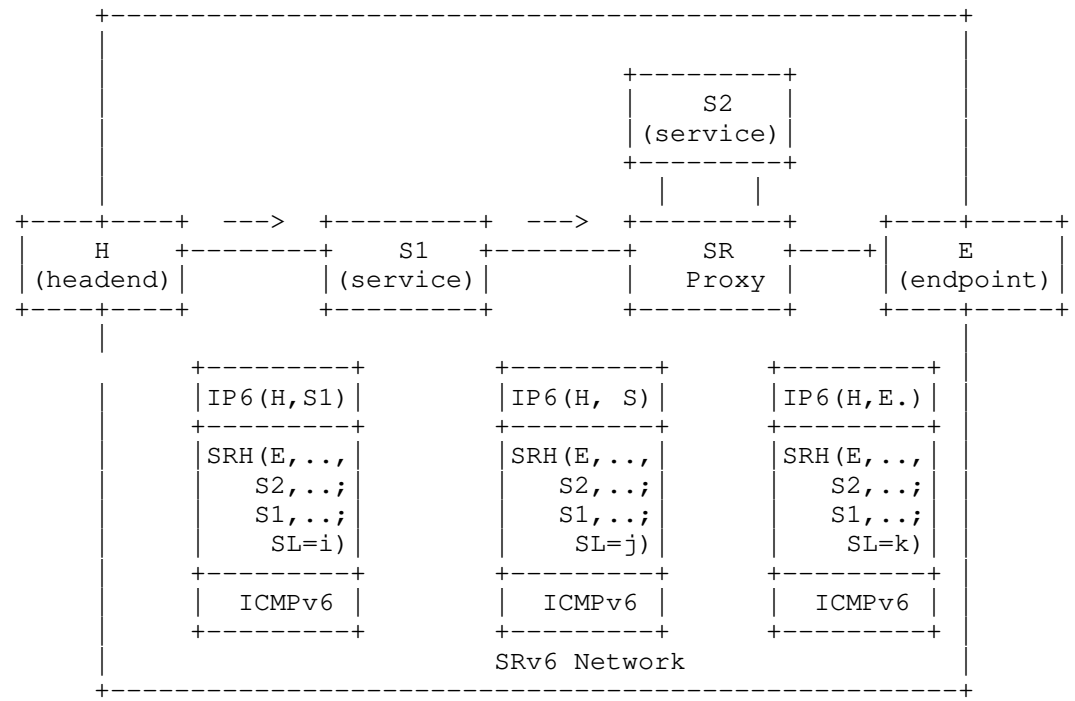


Figure 3. Ping to SR Service Policies in SRv6 Network

S1 (SR-aware service) will apply END function and follow the steps defined in [I-D.draft-ietf-6man-spring-srv6-oam].
The Upper-layer header matches ICMPv6 but
the Segment Left is not 0 and so the packet will be forwarded to the next destination S2. Service function is skipped due to ICMPv6 payload.

SR Proxy upon receiving the packet will match the local proxy SID and follow the steps defined in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming].
The Upper-layer header
does not match Ethernet, IPv4 or IPv6 and so resubmit the packet to the IPv6 module for transmission to the next destination E and service function is skipped.

The endpoint E will process the upper-layer header and reply back to the initiator node H.

6.1.2. Pinging a Service SID

The user interested to ping a specific service SID SR service policy shown in Figure 4 will trigger the ICMPv6 echo request from the headend H with IP6(H,S1) and the upper layer header set to ICMPv6. The probe will be processed along the path as below:

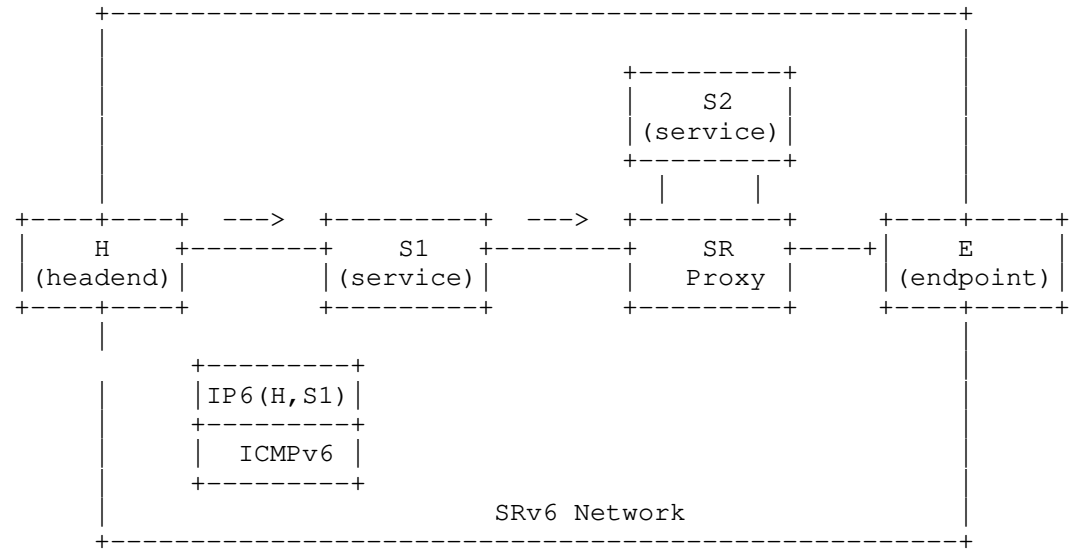


Figure 4. Ping to specific Service SID in SRv6 Network

S1 (SR-aware service) will follow the steps defined in [I-D.draft-ietf-6man-spring-srv6-oam]. Specifically, the service processes the ICMPv6 message and respond to the source, accordingly.

S2 (SR-Unaware Service): The SR Proxy upon receiving the packet will match the local proxy SID and follow the steps defined in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming]. When processing the Upper-layer header of a packet matching a FIB entry locally instantiated SID, the proxy process the ICMPv6 payload and respond to it, accordingly.

6.1.3. Tracing a SR Service Policy

The user interested to trace the SR service policy shown in Figure 2 will trigger the ICMPv6 echo request from the headend H with IPv6(H,S1)(SRH), set the upper layer header set to ICMPv6 and the TTL to 1 and increment the same in the subsequent packets. The probe will be processed along the path as below:

The first probe sent from H will reach S1 (SR-aware service) with Hop Limit of 1. S1 will process TTL expiry as described in [I-D.draft-ietf-6man-spring-srv6-oam] and sends an ICMP Time Exceeded message to H with Code 0.

The second probe sent from H will reach S2 (SR Proxy) with Hop Limit of 1. SR Proxy will process as defined in the step S05 in Sections 6.1.2.1, 6.1.2.2 and 6.1.2.3 of [I-D.ietf-spring-sr-service-programming] and sends an ICMP Time Exceeded message to H with Code 0.

The third probe sent from H will reach E with Hop Limit of 1. E processes TTL expiry as described in [I-D.draft-ietf-6man-spring-srv6-oam] and send an ICMP Time Exceeded message to H with Code 0.

6.2. SR-MPLS Dataplane

To be Updated.

7. IANA Considerations

None.

8. Security Considerations

A local policy may be used to control any malicious use of OAM marker. More details are to be added in a future revision of the document.

9. Acknowledgement

Authors would like to thank Bruno Decraene for review and useful comments.

10. Normative References

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

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Belgium

Email: cfilsfils@cisco.com

Nagendra Kumar Nainar
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: naikumar@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200 Kit Creek Road
Research Triangle Park, NC 27709-4987
US

Email: cpignata@cisco.com

Francois Clad
Cisco Systems, Inc.
France

Email: fclad@cisco.com

Faisal Iqbal
Arista Networks

Email: faisal.ietf@gmail.com

Xiaohu Xu
Alibaba

Email: xiaohu.xxh@alibaba-inc.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

T. Saad
V. Beeram
Juniper Networks
R. Chen
S. Peng
ZTE Corporation
B. Wen
Comcast
D. Ceccarelli
Ericsson

February 22, 2021

Scalable Network Slicing over SR Networks
draft-bestbar-spring-scalable-ns-01

Abstract

Multiple network slices can be realized on top of a single shared network. A router that requires forwarding of a packet that belongs to a slice aggregate may have to decide on the forwarding action to take based on selected next-hop(s), and the forwarding treatment (e.g., scheduling and drop policy) to enforce based on the slice aggregate per-hop behavior. Segment Routing is a technology that enables the steering of packets in a network by encoding pre-established segments within the network into the packet header. This document introduces mechanisms to enable forwarding of packets over a specific slice aggregate along a Segment Routing (SR) path.

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 August 26, 2021.

Table of Contents

1. Introduction

Network slicing allows a service provider, or a network operator to create independent and isolated logical networks on top of a common or shared physical network infrastructure.

When logical network slices are realized on top of a shared physical network, it is important to forward traffic using only the specific resource(s) allocated to the network slice.

The definition of a network slice for use within the IETF and the characteristics of IETF network slice are specified in [I-D.nsdt-teas-ietf-network-slice-definition]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF network slices is discussed in [I-D.nsdt-teas-ns-framework].

[I-D.bestbar-teas-ns-packet] introduces the notion of a Slice Aggregate as the construct that comprises of one of more IETF network slice traffic streams. A slice policy can be used to realize a slice aggregate by instantiating specific control and data plane resources on select topological elements in an IP/MPLS network. The packets belonging to a specific slice aggregate MAY require to be identified so that a specific forwarding treatment (e.g., scheduling and drop policy) is enforced.

Segment Routing (SR) [RFC8402] specifies a mechanism to steer packets through a network by carrying an ordered list of segments. A segment is referred to by its Segment Identifier (SID).

This document introduces two approaches applicable to SR networks that enable forwarding of packet(s) that belong to a slice aggregate over a SR Path.

The first approach extends the SR paradigm by defining a new SID type (slice SID) that, in addition to defining the forwarding action (next-hop selection), associates a SID to slice aggregate and allows enforcing the associated forwarding treatment. The extensions to IGPs for slice aggregate SIDs are defined in [I-D.bestbar-lsr-spring-sa].

The second approach relies on a separate field that is carried in the packet (e.g., MPLS label) to identify the slice aggregate and uses another field (e.g., existing SR segments) for the path selection for the packet.

1.1. Terminology

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. Forwarding over SR Network Slices

A router that receives a packet that belongs to a slice aggregate has to decide on the set of eligible next-hop(s) to forward the packet on (path selection), and on the forwarding treatment (scheduling and drop policy) that needs to be enforced for a specific slice aggregate (slice aggregate selection).

2.1. Path Selection

The segment routing architecture [RFC8402] defines a number of topological segments that may be advertised in routing protocols to allow for a flexible definition of end-to-end paths. For example, an SR-capable IGP router may advertise SIDs for its attached prefixes and adjacencies.

The IGP-Adjacency segment represents the strict path over a specific adjacency between two routers, while the IGP-Prefix segment represents the path to a prefix that is computed over a specific topology and algorithm.

For an IGP-Prefix segment, the IGP uses the topology and algorithm to compute the primary, and optionally alternate (backup) next-hop(s) for a destination prefix. SR allows the use of multiple routing algorithms (e.g., Flexible Algorithms) that enable IGPs on a router to compute paths for Prefix-SIDs whose topology may be constrained and whose paths optimized for additional metric types other than the default IGP cost (e.g., delay metric).

Multiple slice aggregates may overlap over the same topology and require paths for prefixes to be optimized for the same Algorithm. In such case, the IGP selected path for the slice aggregate Prefix-SIDs can share the same IGP computed path (including the primary and backup next-hop(s)). This enables the IGP to optimize the path computation and path programming for such SA Prefix-SIDs.

2.2. Network Slice Selection

The routers in network that forward traffic over links that are shared by multiple slice aggregates need to identify the slice aggregate that the packet belongs to in order to enforce the associated forwarding treatment on it.

[I-D.bestbar-teas-ns-packet] introduces the slice policy as a means to realize a slice aggregate by instantiating specific control and/or data plane resources on select topological elements in the network. In order to enforce a forwarding treatment associated with a slice aggregate, the packets traversing a router MUST be identified as part of a slice aggregate (for example, by using field(s) carried in the packet).

2.2.1. Segment Range as Slice Selector

It is possible to derive the forwarding action (next-hop selection) and the per-hop forwarding treatment from a single field (e.g. SR segment) carried inside a packet that is traversing the SR network.

For example, one way to achieve this is leverage the SR Flexible-Algorithm [I-D.ietf-lsr-flex-algo] to assign SR SID per slice aggregate. A router can can assign and advertise SR Prefix-SIDs per Flex-Algorithm for a prefix to enable reachability over multiple slice aggregates.

For a specific Flexible Algorithm, the range of Prefix-SIDs of all prefixes in the network can be used as a slice selector mapping to a specific slice aggregate. This approach does not require protocol extensions to be realized; however, it poses serious IGP scalability concerns when realizing a large number of slice aggregates.

Alternatively, this document proposes to extend the IGP SR Prefix-SID and Adjacency-SID sub-TLVs defined in [RFC8667] and [RFC8665] to carry an additional distinguisher (Slice Aggregate identifier) to allow multiple SIDs to be assigned (and advertised) for the same topological element for the same Flexible-Algorithm and topology. In such a case, a transit router can use the SR slice aggregate SID carried in the packet to identify the slice aggregate, as well as to determine the forwarding next-hop.

Multiple Slice Aggregate Prefix-SIDs (SA Prefix-SIDs) can be assigned to the same prefix, while they share the same topology and Algorithm. The SA Prefix-SIDs can also share the same IGP computed paths (primary and backup). Similarly, multiple Slice Aggregate Adjacency-SIDs (SA Adjacency-SIDs) can be allocated for the same adjacency between the two routers to distinguish forwarding over the same adjacency for each slice aggregate. The extensions for IGPs to advertise SA Prefix-SIDs and SA Adjacency-SIDs are defined in [I-D.bestbar-lsr-spring-sa].

The same forwarding treatment MUST be enforced on all packets belonging to a slice aggregate but destined to different topological elements in the network. In this case, a range of SA (Prefix and Adjacency) SIDs is used to select the slice aggregate, and hence enforce the same forwarding treatment on them.

Note that this approach requires maintaining per slice aggregate state for each topological element on every router in the network in both the control and data plane. For example, a network composed of 'N' routers, where each router has up to 'K' adjacencies to its neighbors, a router would have to assign and advertise 'M' SA Prefix-SIDs and 'M' SA Slice Adjacency-SID(s) to each of it 'K' adjacencies. Consequently, a router will have to maintain up to (N+K)*M SIDs in the control plane, and an equal number of labeled routes in its forwarding plane.

Consider a network shown in Figure 1 that is enabled for SR. The Segment Routing Global Block (SRGB) on all routers is assumed to start from 16000. We assume the links in the network are partitioned amongst two network slice aggregates: SA1, and SA2.

- Node R5 assigns two Algorithm 0 SA Prefix-SIDs, index=105, and index=205 to represent the shortest IGP to R5 for slice aggregates SA1 and SA2 respectively.

- A Flexible Algorithm Definition (FAD) for Algorithm 128 is defined by the user such that the FAD Metric-Type is 1 (Min Unidirectional Link Delay).

- Node R5 assigns two Algorithm 128 SA Prefix-SIDs, index=815, and index=825 to represent the least delay path to R5 for slice aggregates SA1 and SA2 respectively.

- All routers in the network participate and advertise their capability to compute FAD 128 Prefix-SID paths.

Using the approach described in this section, R1 is able to forward packets that traverse slices aggregates SA1 and SA2 along the least delay path by imposing the MPLS SR SID 16815, and 16825 respectively.

In addition, R1 is able to forward packets that traverse slice aggregate SA1 and SA2 along the IGP shortest path to R5 by imposing the MPLS SR SID 16015, and 16025 respectively.

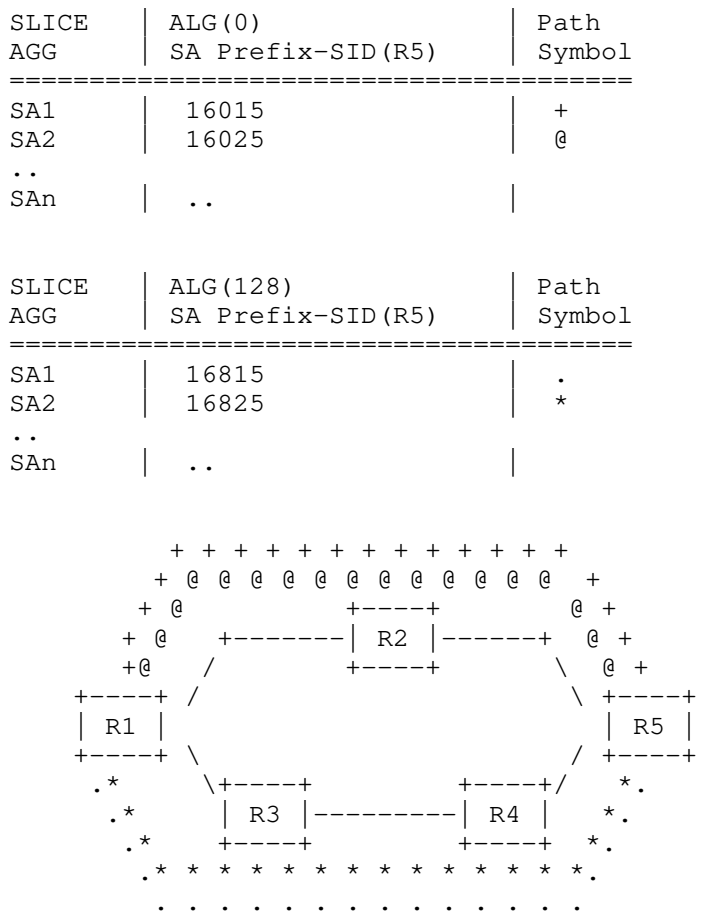


Figure 1: Example of forwarding over slice aggregates using SR Paths.

2.2.2. Global Identifier as Slice Selector

It is possible that the forwarding action and the per-hop behavior treatment is derived from different fields carried in a packet. For example, a packet can carry a global slice selector field that can be used to define the forwarding treatment while the forwarding next-hop relies on the SR topological SIDs. This makes the slice aggregate identification independent of the topology or the destination of the packet, and thus, allows for scalable slice aggregates.

The Slice aggregate Selector (SS) is carried in each packet destined to any topological element and that is to be steered over the slice aggregate. For example, the slice aggregate SS can be carried in an MPLS label that is present in an MPLS packet's label stack. It is

possible, also, to have a range of MPLS labels to represent the SS associated with slice aggregate.

When the slice aggregate is realized over an IPv6 dataplane, the SS can be encoded in the IP header. For example, the SS can be encoded in a portion of the IPv6 Flow Label field as described in [I-D.filsfils-spring-srv6-stateless-slice-id].

Routers within the network use the topological SR segment SIDs to determine the forwarding action (next-hop selection), and use the slice aggregate selector to enforce the dataplane policy (e.g., as defined by the slice policy in [I-D.bestbar-teas-ns-packet]).

The SS label may be embedded at different positions in the MPLS label stack. For example, the SS label MAY be located at the top of the MPLS packet label stack and maintained, by each hop, while the packet is forwarded along the SR path. However, since assigning a global MPLS label on all nodes for the SS may not be always feasible, an alternative is to assign a global Index for a Slice Aggregate Selector (SA Selector Index). In this case, the SA Selector Index is used to determine the actual MPLS label value (e.g., from the router Global Label Block) on a given router.

The SS label can also reside at the bottom of the label stack. For example, a range of VPN service labels may also serve as a SS to map traffic from multiple VPNs to the same slice aggregate.

Another option is to encode the SS as part of a well-known label such as Entropy Label (EL) as suggested in [I-D.decraene-mpls-slid-encoded-entropy-label-id]. This optimizes the number of the MPLS labels needed in the stack and provides an ease incremental deployment.

Lastly, a new Special Purpose Label- e.g., Slice Selector Indicator (SSI)- from the MPLS the Base Special-Purpose MPLS Label, or Extended Special-Purpose MPLS Label spaces can be used to indicate that a SS label immediately follows the SSI. In this case, the ingress router of slice aggregate boundary will impose at least two additional MPLS labels (SSI + SS) to identify the packets that belong to the slice aggregate.

This approach reduces the amount of state required to be stored on a router to allow forwarding over slice aggregates since it does not require a Prefix-SID state per slice aggregate in the control plane, nor in the forwarding plane.

To illustrate forwarding over slice aggregates using a SS label, we consider the same network described earlier in Figure 1, but with some changes in the configuration:

- Node R5 assigns an Algorithm 0 Prefix-SID of index=5 to represent the shortest IGP path from any router to R5.
- Node R5 assigns Algorithm 128 Prefix-SID of index=805 to represent the least delay path from any router to R5.
- All routers in the network participate and advertise their capability to compute FAD 128 Prefix-SID paths.
- The SS labels 1001 and 1002 are used for packets that require to be forwarded over slice aggregates SA1 and SA2 respectively.

Using the approach described in this section, R1 is able to forward packets that traverse slice aggregate SA1 and SA2 along the least delay path by imposing the following labels {16805, SSI, 1001} and {16805, SSI, 1002} respectively.

Similarly, R1 is able to forward packets that traverse over slice aggregates SA1 and SA2 along the IGP shortest path to R5 by imposing the following labels {16005, SSI, 1001} and {16005, SSI, 1002} respectively. The path that the packets traverse in each of the above case remains as described in Figure 1.

3. IANA Considerations

This document has no IANA actions.

4. Security Considerations

The main goal of network slicing is to allow for some level of isolation for traffic from multiple different network slices that are utilizing a common network infrastructure and to allow for different levels of services to be provided for traffic traversing a single slice aggregate resource(s).

A variety of techniques may be used to achieve this, but the end result will be that some packets may be mapped to specific resource(s) and may receive different (e.g., better) service treatment than others. The mapping of network traffic to a specific slice is indicated primarily by the SS, and hence an adversary may be able to utilize resource(s) allocated to a specific network slice by injecting packets carrying the same SS field in their packets.

Such theft-of-service may become a denial-of-service attack when the modified or injected traffic depletes the resources available to forward legitimate traffic belonging to a specific slice aggregate.

The defense against this type of theft and denial-of-service attacks consists of the combination of traffic conditioning at network slicing domain boundaries with security and integrity of the network infrastructure within a network slicing domain.

5. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, and Prabhu Raj Villadathu Karunakaran for their review of this document, and for providing valuable feedback on it.

6. Contributors

The following individuals contributed to this document:

Colby Barth
Juniper Networks
Email: cbarth@juniper.net

Srihari R. Sangli
Juniper Networks
Email: ssangli@juniper.net

Chandra Ramachandran
Juniper Networks
Email: csekar@juniper.net

7. References

7.1. Normative References

[I-D.bestbar-lsr-spring-sa]
Saad, T., Beeram, V., Chen, R., Peng, S., Wen, B., and D. Ceccarelli, "IGP Extensions for SR Slice Aggregate SIDs", February 2021.

[I-D.bestbar-teas-ns-packet]
Saad, T., Beeram, V., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., and X. Liu, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-01 (work in progress), December 2020.

[I-D.decraene-mpls-slid-encoded-entropy-label-id]
Decraene, B., Filsfils, C., Henderickx, W., Saad, T., and V. Beeram, "Using Entropy Label for Network Slice Identification in MPLS networks.", draft-decraene-mpls-slid-encoded-entropy-label-id-00 (work in progress), December 2020.

[I-D.filsfils-spring-srv6-stateless-slice-id]
Filsfils, C., Clad, F., Camarillo, P., and K. Raza, "Stateless and Scalable Network Slice Identification for SRv6", draft-filsfils-spring-srv6-stateless-slice-id-02 (work in progress), January 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

7.2. Informative References

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-02 (work in progress), December 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Ran Chen
ZTE Corporation

Email: chen.ran@zte.com.cn

Shaofu Peng
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Daniele Ceccarelli
Ericsson

Email: daniele.ceccarelli@ericsson.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

S. Hegde
R. Bonica
Juniper Networks
P. Shaofu
G. Mirsky
Z. Zhang
ZTE Corporation
B. Decraene
Orange

February 22, 2021

The SRv6 END.DTM Endpoint Behavior
draft-bonica-spring-srv6-end-dtm-04

Abstract

This document describes a new SRv6 endpoint behavior, called END.DTM. END.DTM supports inter-working between SRv6 and SR-MPLS. Like any endpoint behavior, END.DTM contains a function and arguments. The function causes the processing node to decapsulate a packet, impose an SR-MPLS label stack and forward the packet. The arguments determine SR-MPLS label stack contents.

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 August 26, 2021.

Table of Contents

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 [I-D.ietf-spring-srv6-network-programming]. When SR is implemented over MPLS, it is called SR-MPLS [RFC8660].

This document describes a new SRv6 endpoint behavior, called END.DTM. END.DTM supports inter-working between SRv6 and SR-MPLS. Like any endpoint behavior, END.DTM contains a function and arguments. The function causes the processing node to:

- Decapsulate a packet (i.e., remove an IPv6 header and its extensions).
- Impose an SR-MPLS label stack.
- Forward the packet.

The arguments determine MPLS-label stack contents and anything that might be encoded in the MPLS-label stack (e.g., transport class [I-D.hegde-spring-mpls-seamless-sr])

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. Use-case

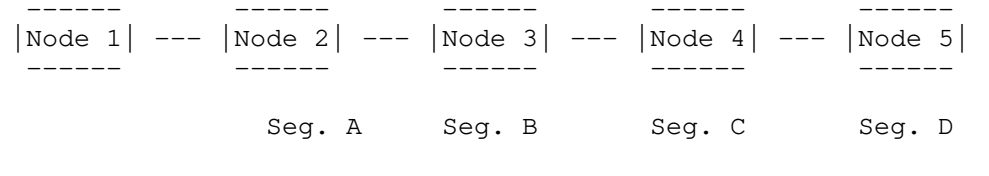


Figure 1: END.DTM Use-case

Figure 1 depicts an inter-working SR path. The SR path originates on Node 1 and terminates on Node 5. It contains:

- An SRv6 part
- An SR-MPLS part

The SRv6 part includes Nodes 1, 2 and 3. Nodes 1 and 2 MUST be SRv6-capable but are NOT REQUIRED to be SR-MPLS capable. An END.DTM segment is instantiated on Node 3. Therefore, Node 3 MUST be SRv6-capable and SR-MPLS capable.

The SRv6 part also includes:

- Segment A - An END segment that is instantiated on Node 2.
- Segment B - An END.DTM segment that is instantiated on Node 3.

The SR-MPLS part includes Nodes 4 and 5. These nodes MUST be SR-MPLS-capable but are NOT REQUIRED to be SRv6 capable.

The SR-MPLS part also includes:

- Segment C - A prefix segment that is instantiated on Node 4.
- Segment D - A prefix segment that is instantiated on Node 5.

The following paragraphs describe how a packet traverses this inter-working SR path:

Node 1 encapsulates the packet in an SRv6 header. The SRv6 header contains the following Segment Identifiers (SID):

- A SID representing Segment A, encoded in the Destination Address field of the IPv6 header.
- A SID representing Segment B, encoded in a Segment Routing Header (SRH) [RFC8754].

Node 1 sends the packet to Node 2. When the packet arrives at Node 2, The Destination Address field in the IPv6 header represents a locally instantiated END SID. Node 2 processes the packet as follows:

- Decrement the Segments Left field in the SRH
- Copy the next SID from the SRH to the Destination Address field of the IPv6 header.
- Forward the packet to Node 3.

When the packet arrives at Node 3, The Destination Address field in the IPv6 header represents a locally instantiated END.DTM SID. Node 3 processes the packet as follows:

- Decapsulate the packet (i.e., remove the IPv6 header and its extensions, including the SRH)
- Push two SR-MPLS label stack entries, representing Segments D and C. Set the MPLS Traffic Class and TTL values to reflect the Traffic Class and Hop count values received in the IPv6 header.
- Forward the packet to Node 4.

When the packet arrives at Node 4, it is encapsulated in an SR-MPLS label stack. Node 4 processes the packet as described in SR-MPLS [RFC8660].

4. Processing

The End.DTM SID MUST be the last segment in a SR Policy. Its arguments are associated with an SR-MPLS label stack.

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

```
S01. When an IPv6 Routing Header 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 executes the following procedure:

S01. Decapsulate the packet (i.e., remove the outer IPv6 Header and all its extension headers)
S02. Push the SR-MPLS label stack that is associated with the END.DTM arguments. Set the MPLS Traffic Class and TTL values to reflect the Traffic Class and Hop count values received in the IPv6 header.
S03. Submit the packet to the MPLS FIB lookup for transmission to the new destination

5. 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.

6. 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 [I-D.ietf-spring-srv6-network-programming].

7. Acknowledgements

Thanks to Melchior Aelmans, Bruno Decraene, Takuya Miyasaka and Jeff Tantsura for their comments.

8. References

8.1. Normative References

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

8.2. Informative References

[I-D.hegde-spring-mpls-seamless-sr]
Hegde, S., Bowers, C., Xu, X., Gulko, A., Bogdanov, A., Uttaro, J., Jalil, L., Khaddam, M., and A. Alston, "Seamless Segment Routing", draft-hegde-spring-mpls-seamless-sr-04 (work in progress), January 2021.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

Authors' Addresses

Shraddha Hegde
Juniper Networks
Embassy Business Park
Bangalore, KA 560093
India

Email: shraddha@juniper.net

Ron Bonica
Juniper Networks
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Peng Shaofu
ZTE Corporation
Peoples Republic of China

Email: peng.shaofu@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Zheng Zhang
ZTE Corporation
Peoples Republic of China

Email: zhang.zheng@zte.com.cn

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Inter-Domain Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 15, 2021

K. Deevi
K. Raza
Cisco
K. Majumdar
Commscope
B. Decraene
Orange
Z. Jiang
Tencent

January 11, 2021

YANG data model for BGP Segment Routing TE Extensions
draft-deevi-idr-bgp-srte-yang-01

Abstract

This document defines a YANG data model that can be used to configure and manage Segment Routing TE extensions in BGP.

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 July 15, 2021.

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) [RFC8040] and encodings other than XML (e.g. JSON) [RFC7951] are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines the YANG model for Segment Routing TE specific extensions in BGP.

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. BGP Segment Routing Traffic Engineering Yang model

2.1. Overview

Segment Routing (SR), as defined in [RFC8402], leverages the source routing paradigm where a node steers a packet through an ordered list of instructions, called segments. SR, thus, allows enforcing a flow through any topological path and/or service chain while maintaining per-flow state only at the ingress nodes to the SR domain.

When applied to ipv6 data-plane (i.e. SRv6), the ordered set of instructions are realized via SRv6 SIDs. The various functions and behaviors corresponding to network programming using SRv6 are specified in [I-D.ietf-spring-srv6-network-programming].

This document defines Yang model for the Segment Routing TE extensions applicable for BGP as following:

- BGP signaled SR Policy as described in [I-D.ietf-idr-segment-routing-te-policy].

- Automatic Steering as described in [I-D.ietf-spring-segment-routing-policy] and [I-D.ietf-idr-segment-routing-te-policy].

The Yang extensions proposed in this model augment the base BGP model defined in [I-D.ietf-idr-bgp-model].

Note: Base BGP model does not have a common structure for BGP RIB. The placeholder containers defined in this model can be removed once base BGP model has the BGP RIB structure.

The modeling in this document complies with the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all

"config true" (rw) and "config false" (ro) nodes defined in the schema.

2.2. SR Policy

Architecture for SR Policies is described in [I-D.ietf-spring-segment-routing-policy]. BGP Signaled SR Policies are described in the [I-D.ietf-idr-segment-routing-te-policy]. Following Yang extensions for SR Policy configuration and state data are applicable:

- Addition of identies extending the BGP-AFI-SAFI base identity. This is to add two new address families namely IPv4 SR-policy and IPv6 SR-policy, as described in [I-D.ietf-idr-segment-routing-te-policy].

- BGP Signaled SR Policy candidate paths. These refer to the explicit candidate paths signaled via BGP as SAFI NLRIs, state of which is applicable in the context of BGP speaker process. This is modeled by adding SR Policy address family specific container under generic BGP afi-safi list entry defined in the base BGP model [I-D.ietf-idr-bgp-model].

- On Demand SR Policy candidate paths. These refer to the dynamic candidate paths as described in [I-D.ietf-spring-segment-routing-policy]. There are two parts to this in the context of BGP. A set of authorized SR Policy colors for on demand policy triggers, and the actual instantiated candidate paths per BGP next-hop. New containers and lists are added under BGP global mode to model this information.

- SR Policy state in the context of BGP speaker. This represents the state SR Policies (regardless of method of instantiation per candidate path). The SR Policy state is maintained in the context of BGP speaker process to realize the Automatic Steering of overlay routes. Automatic Steering extensions are described in the next section.

Note: The common parameters and datatypes for the SR Policy, currently defined in this model, should be imported from SR Policy Manager model, once available.

2.3. Automatic Steering

Automatic Steering (AS) refers to the ability to forward traffic over a SR Policy on the head-end, as described in [I-D.ietf-spring-segment-routing-policy]. When a BGP route is received with the color extended community and if the color value

matches the color of an authorized SR Policy installed on the head-end, the route is programmed to resolve over SR Policy in forwarding. Automatic Steering information associated with the BGP routes is modeled as state information per route.

TBD: The configuration parameters for Automatic Steering are yet to be added as an augmentation to the BGP route policy model. Such as, extensions for opaque color extended community in BGP policy model, and the Color Only (CO) flags controlling the Automatic Steering behavior as described in [I-D.ietf-idr-segment-routing-te-policy].

3. Yang Tree

3.1. SR Policy

On Demand Nexthop (ODN) policies triggered by BGP

```
augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global:
   +--rw segment-routing
      +--rw on-demand-policies
      |  +--ro authorized-colors
      |  |  +--ro colors* [color]
      |  |     +--ro color    uint32
      |  +--ro installed-policies
      |     +--ro sr-policy* [color end-point]
      |        +--ro color        uint32
      |        +--ro end-point    inet:ip-address
      +--ro policy-state
         +--ro sr-policy* [color end-point]
            +--ro color                uint32
            +--ro end-point            inet:ip-address
            +--ro policy-state?        enumeration
            +--ro binding-sid?         sid-type
            +--ro steering-disabled?   empty
            +--ro ref-count?           uint32
```

BGP Signaled Explicit SR Policies under ipv4 and ipv6 SR-Policy SAFI

```
augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi:
   +--rw ipv4-srpolicy
      +--ro explicit-policies
         +--ro sr-policy* [distinguisher color end-point]
            +--ro distinguisher           uint32
            +--ro color                   uint32
            +--ro end-point               inet:ip-address
            +--ro preference?             uint32
            +--ro explicit-binding-sid
            |  +--ro binding-sid?       sid-type
```

```
            |  +--ro strict?            boolean
            |  +--ro drop-on-invalid?   boolean
            +--ro usable?                 boolean
            +--ro registered?             boolean
            +--ro segment-lists
               +--ro segment-list* [weight]
                  +--ro weight      uint32
                  +--ro segments
                     +--ro segment* [index]
                        +--ro index            uint32
                        +--ro type?            segment-type
                        +--ro segment-types
                           +--ro segment-type-1
                           |  +--ro sid-value?   rt-types:mpls-label
                           +--ro segment-type-2
                           |  +--ro sid-value?   srv6-types:srv6-sid
                           +--ro segment-type-3
                           |  +--ro ipv4-address?   inet:ipv4-address
                           |  +--ro algorithm?      uint8
                           +--ro segment-type-4
                           |  +--ro ipv6-address?   inet:ipv6-address
                           |  +--ro algorithm?      uint8
                           +--ro segment-type-5
                           |  +--ro ipv4-address?           inet:ipv4-address
                           |  +--ro interface-identifier?   uint32
                           +--ro segment-type-6
                           |  +--ro local-ipv4-address?    inet:ipv4-address
                           |  +--ro remote-ipv4-address?   inet:ipv4-address
                           +--ro segment-type-7
                           |  +--ro local-ipv6-address?            inet:ipv6-addr
ess
                           |  +--ro local-interface-identifier?    uint32
                           |  +--ro remote-ipv6-address?           inet:ipv6-addr
ess
                           |  +--ro remote-interface-identifier?   uint32
                           +--ro segment-type-8
                           |  +--ro local-ipv6-address?    inet:ipv6-address
                           |  +--ro remote-ipv6-address?   inet:ipv6-address
                           +--ro segment-type-9
                           |  +--ro ipv6-address?   inet:ipv6-address
                           |  +--ro algorithm?      uint8
                           +--ro segment-type-10
                           |  +--ro local-ipv6-address?            inet:ipv6-addr
ess
                           |  +--ro local-interface-identifier?    uint32
                           |  +--ro remote-ipv6-address?           inet:ipv6-addr
ess
                           |  +--ro remote-interface-identifier?   uint32
                           +--ro segment-type-11
                              +--ro local-ipv6-address?    inet:ipv6-address
                              +--ro remote-ipv6-address?   inet:ipv6-address
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi:
```

```
   +--rw ipv6-srpolicy
      +--ro explicit-policies
         +--ro sr-policy* [distinguisher color end-point]
            +--ro distinguisher           uint32
            +--ro color                   uint32
            +--ro end-point               inet:ip-address
            +--ro preference?             uint32
            +--ro explicit-binding-sid
            |  +--ro binding-sid?       sid-type
            |  +--ro strict?            boolean
            |  +--ro drop-on-invalid?   boolean
            +--ro usable?                 boolean
            +--ro registered?             boolean
            +--ro segment-lists
               +--ro segment-list* [weight]
                  +--ro weight      uint32
                  +--ro segments
                     +--ro segment* [index]
                        +--ro index            uint32
                        +--ro type?            segment-type
                        +--ro segment-types
                           +--ro segment-type-1
                           |  +--ro sid-value?   rt-types:mpls-label
                           +--ro segment-type-2
                           |  +--ro sid-value?   srv6-types:srv6-sid
                           +--ro segment-type-3
                           |  +--ro ipv4-address?   inet:ipv4-address
                           |  +--ro algorithm?      uint8
                           +--ro segment-type-4
                           |  +--ro ipv6-address?   inet:ipv6-address
                           |  +--ro algorithm?      uint8
                           +--ro segment-type-5
                           |  +--ro ipv4-address?           inet:ipv4-address
                           |  +--ro interface-identifier?   uint32
                           +--ro segment-type-6
                           |  +--ro local-ipv4-address?    inet:ipv4-address
                           |  +--ro remote-ipv4-address?   inet:ipv4-address
                           +--ro segment-type-7
                           |  +--ro local-ipv6-address?            inet:ipv6-addr
ess
                           |  +--ro local-interface-identifier?    uint32
                           |  +--ro remote-ipv6-address?           inet:ipv6-addr
ess
                           |  +--ro remote-interface-identifier?   uint32
                           +--ro segment-type-8
                           |  +--ro local-ipv6-address?    inet:ipv6-address
                           |  +--ro remote-ipv6-address?   inet:ipv6-address
                           +--ro segment-type-9
                           |  +--ro ipv6-address?   inet:ipv6-address
                           |  +--ro algorithm?      uint8
```

```
                           +--ro segment-type-10
                           |  +--ro local-ipv6-address?            inet:ipv6-addr
ess
                           |  +--ro local-interface-identifier?    uint32
                           |  +--ro remote-ipv6-address?           inet:ipv6-addr
ess
                           |  +--ro remote-interface-identifier?   uint32
                           +--ro segment-type-11
                              +--ro local-ipv6-address?    inet:ipv6-address
                              +--ro remote-ipv6-address?   inet:ipv6-address
```

3.2. Automatic Steering

Yang Tree for Automatic Steering with example of ipv4-unicast SAFI

```
module: ietf-bgp-srte
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
   +--ro routes
      +--ro route* [prefix neighbor add-path-id]
         +--ro prefix                union
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast:
   +--ro routes
      +--ro route* [prefix neighbor add-path-id]
         +--ro prefix                union
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-labeled-unicast:
   +--ro routes
      +--ro route* [prefix neighbor add-path-id]
         +--ro prefix                union
```

```
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid
```

```
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-labeled-unicast:
   +--ro routes
      +--ro route* [prefix neighbor add-path-id]
         +--ro prefix                union
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
   +--ro routes
      +--ro route* [rd prefix neighbor add-path-id]
         +--ro rd                    rt-types:route-distinguisher
         +--ro prefix                union
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid
   augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/bgp:b
gp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast:
   +--ro routes
      +--ro route* [rd prefix neighbor add-path-id]
         +--ro rd                    rt-types:route-distinguisher
         +--ro prefix                union
         +--ro neighbor              inet:ip-address
         +--ro add-path-id           uint32
         +--ro automatic-steering
            +--ro color?         -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/color
            +--ro end-point?     -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/end-point
            +--ro co-flag?       enumeration
            +--ro binding-sid?   -> /rt:routing/control-plane-protocols/control-p
lane-protocol/bgp:bgp/global/ietf-bgp-srte:segment-routing/policy-state/sr-policy
/binding-sid

 ...
```

4. Yang Module

```
<CODE BEGINS> file "ietf-bgp-srte@2019-07-07.yang"

module ietf-bgp-srte {

  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-srte";
```

```
  prefix "ietf-bgp-srte";

  import ietf-routing-types {
    prefix rt-types;
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-bgp {
    prefix bgp;
  }

  import ietf-bgp-types {
    prefix bgp-types;
  }

  import ietf-srv6-types {
    prefix srv6-types;
  }

  organization
     "IETF Inter-Domain Routing Working Group";

  contact
     "Inter-Domain working group - idr@ietf.org";

  description
    "This YANG module defines a data model to configure and
     manage segment routing extensions in BGP.

     Terms and Acronyms

     AF : Address Family

     BGP (bgp) : Border Gateway Protocol

     SR : Segment Routing

     SID : Segment Identifier

     SRv6 : Segment Routing with IPv6 Data plane
```

```
     VPN : Virtual Private Network

     VRF : Virtual Routing and Forwarding

    ";

  revision 2018-06-26 {
    description
      "Initial revision" ;
    reference "";
  }

  //
  // New identities and typedefs for SR extensions
  //

  // SR Policy SAFI identities
  identity IPV4_SRPOLICY {
    base bgp-types:afi-safi-type;
    description
     "IPv4 SR Policy (AFI,SAFI = 1,73)";
    reference "TBD";
  }

  identity IPV6_SRPOLICY {
    base bgp-types:afi-safi-type;
    description
      "IPv6 SR Policy (AFI,SAFI = 2,73)";
    reference "TBD";
  }

  typedef segment-type {
    type enumeration {
      enum segment-type-1 {
        value 1;
        description "SR-MPLS Label";
      }
      enum segment-type-2 {
        value 2;
        description "SRv6 SID";
      }
      enum segment-type-3 {
        value 3;
        description "IPv4 Prefix with optional SR Algorithm";
      }
      enum segment-type-4 {
        value 4;
        description "IPv6 Global Prefix with optional SR Algorithm for SR-MPLS";
```

```
      }
      enum segment-type-5 {
        value 5;
        description "IPv4 Prefix with Local Interface ID";
      }
      enum segment-type-6 {
        value 6;
        description "IPv4 Addresses for link endpoints as Local, Remote pair";
      }
      enum segment-type-7 {
        value 7;
        description "IPv6 Prefix and Interface ID for link endpoints as Local,
           Remote pair for SR-MPLS";
      }
      enum segment-type-8 {
        value 8;
        description "IPv6 Addresses for link endpoints as Local, Remote pair for
           SR-MPLS";
      }
      enum segment-type-9 {
        value 9;
        description "IPv6 Global Prefix with optional SR Algorithm for SRv6";
      }
      enum segment-type-10 {
        value 10;
        description "IPv6 Prefix and Interface ID for link endpoints as Local,
           Remote pair for SRv6";
      }
      enum segment-type-11 {
        value 11;
        description "IPv6 Addresses for link endpoints as Local, Remote pair for
           SRv6";
      }
    }
    description "SR segment type";
  }

  // Sid type union
  typedef sid-type {
    type union {
      type rt-types:mpls-label;
      type srv6-types:srv6-sid;
    }
    description "Type definition for Segment Identifier. This is
                 a union type which can be either a SR MPLS SID in the
                 form of a label,  or a SRv6 SID in the form of
                 an IPv6 address.";
    reference "TBD";
```

```
  }

  //
  // SR Policy Related Groupings
  //
  //Color and Endpoint of the SR Policy
  grouping sr-policy-color-endpoint {
    description "Common grouping for SR Policy Color and
                 Endpoint";
    leaf color {
      type uint32;
      description "Color of the policy";
    }

    leaf end-point {
      type inet:ip-address;
      description "Endpoint of the policy";
    }
  }
  // Authorized colors for On Demand SR Policy programming
  grouping sr-odn-auth-colors {
    description
      "Authorized colors for On Demand (dynamic) SR Policies
       towards BGP nexthops";
    container authorized-colors {
      config false;
      description
        "Authorized colors for On Demand (dynamic) SR policies
         towards BGP nexthops";
      list colors {
        key "color";
        description "List of SR Policy Colors";
        leaf color {
          type uint32;
          description "Color value";
        }
      }
    }
  }

  grouping sr-policy-cmn-state {
    description "Common state parameters applicable to
                 SR Policies";
    leaf policy-state {
      type enumeration {
        enum UP {
          description "SR Policy state UP";
        }
```

```
        enum DOWN {
          description "SR Policy state DOWN";
        }
      }
      description "SR Policy forwarding state";
    }

    leaf binding-sid {
      type sid-type;
      description "Binding SID of the SR Policy";
    }

    leaf steering-disabled {
      type empty;
      description "This attribute is set if steering
                   is disabled on this SR policy";
    }

    leaf ref-count {
      type uint32;
      description "Count of routes steering over this policy";
    }
  }

  //
  // SR Policy State grouping
  //
  grouping sr-policy-state {
    description "SR Policy State";
    container policy-state {
      config false;
      description "SR Policy State";
      list sr-policy {
        key "color end-point";
        description "List of SR Policies";

        uses sr-policy-color-endpoint;

        // State of the SR Policy in BGP
        uses sr-policy-cmn-state;
      }
    }
  }

  grouping sr-exp-policy-cp-state {
    description "State of BGP signaled SR Policy (explicit)
                 candidate paths";
    container explicit-policies {
```

```
      config false;
      description "BGP signaled explicit SR Policies";
      list sr-policy {
        key "distinguisher color end-point";
        description "List of BGP signaled explicit SR Policies";
        leaf distinguisher {
          type uint32;
          description "Distinguisher of the SR Policy
                       candidate path";
        }

        uses sr-policy-color-endpoint;

        leaf preference {
          type uint32;
          description "Preference of the SR Policy candidate path";
        }

        container explicit-binding-sid {
          description "Explicitly supplied Binding SID
                       for this policy";
          leaf binding-sid {
            type sid-type;
            description "Binding SID value";
          }
          leaf strict {
            type boolean;
            description "Boolean indicating that the node
                         must use only the supplied Binding SID
                         for this SR Policy.
                         reference: TBD";
          }
          leaf drop-on-invalid {
            type boolean;
            description "Boolean to indicate drop upon invalid
                         policy, behavior. This overwrites the
                         default behavior of fallback to IGP path
                         , when SR Policy is (or becomes) invalid.
                         reference: TBD";
          }
        }

        leaf usable {
          type boolean;
          description "Boolean to indicate that the SR Policy is
                       usable on this node.
                       reference: TBD";
        }
```

```
        leaf registered {
          type boolean;
          description "Boolean to indicate that the SR policy
                       is registered with policy manager to
                       install the corresponding forwarding entry";

        }

        uses segment-lists;
        // TODO: Segment Lists and other parameters from SR Policy model
        //       to be imported here.
      }
    }
  }

  grouping segment-lists {
    description
      "Segment lists grouping";
    container segment-lists {
      description "Segment-lists properties";

      list segment-list {
        key "weight";
        description "Segment-list";
        leaf weight {
          type uint32;
          description "Segment-list weight";
        }
        container segments {
          description
            "Segments for given segment list";

          list segment {
            key "index";
            description "Segment/hop at the index";
            uses segment-properties;
          }
        }
      }
    }
  }

  grouping segment-properties {
    description "Segment properties grouping";
    leaf index {
      type uint32;
      description "Segment index";
    }
```

```
    leaf type {
      type segment-type;
      description "Segment type";
    }
    container segment-types {
      description "Types of segments";
      container segment-type-1 {
        description
           "Segment declared by MPLS label";
        leaf sid-value {
           type rt-types:mpls-label;
           description "MPLS label value";
        }
      }
      container segment-type-2 {
        description
           "Segment declared by SRv6 SID value";
        leaf sid-value {
           type srv6-types:srv6-sid;
           description "SRv6 SID value";
        }
      }
      container segment-type-3 {
        description
           "Segment declared by IPv4 Prefix with optional SR Algorithm";
        leaf ipv4-address {
          type inet:ipv4-address;
          description "Segment IPv4 address";
        }
        leaf algorithm {
          type uint8;
          description "Prefix SID algorithm identifier";
        }
      }
      container segment-type-4 {
        description
           "Segment declared by IPv6 Global Prefix with optional
            SR Algorithm for SR-MPLS";
        leaf ipv6-address {
          type inet:ipv6-address;
          description "Segment IPv6 address";
        }
        leaf algorithm {
          type uint8;
          description "Prefix SID algorithm identifier";
        }
      }
      container segment-type-5 {
```

```
        description
           "Segment declared by IPv4 Prefix with Local Interface ID";
        leaf ipv4-address {
          type inet:ipv4-address;
          description "Node IPv4 address";
        }
        leaf interface-identifier {
          type uint32;
          description "local interface identifier";
        }
      }
      container segment-type-6 {
        description
           "Segment declared by IPv4 Addresses for link endpoints
            as Local, Remote pair";
        leaf local-ipv4-address {
          type inet:ipv4-address;
          description "Segment local IPv4 adjacency address";
        }
        leaf remote-ipv4-address {
          type inet:ipv4-address;
          description "Segment remote IPv4 adjacency address";
        }
      }
      container segment-type-7 {
        description
           "Segment declared by IPv6 Prefix and Interface ID for
           link endpoints as Local, Remote pair for SR-MPLS";
        leaf local-ipv6-address {
          type inet:ipv6-address;
          description "Local link IPv6 address";
        }
        leaf local-interface-identifier {
          type uint32;
          description "Local interface identifier";
        }
        leaf remote-ipv6-address {
          type inet:ipv6-address;
          description "Remote link IPv6 address";
        }
        leaf remote-interface-identifier {
          type uint32;
          description "Remote interface identifier";
        }
      }
      container segment-type-8 {
        description
           "Segment declared by IPv6 Addresses for link endpoints as
```

```
           Local, Remote pair for SR-MPLS";
        leaf local-ipv6-address {
          type inet:ipv6-address;
          description "Segment local IPv6 adjacency address";
        }
        leaf remote-ipv6-address {
          type inet:ipv6-address;
          description "Segment remote IPv6 adjacency address";
        }
      }
      container segment-type-9 {
        description
          "Segment declared by IPv6 Global Prefix with optional
           SR Algorithm for SRv6";
        leaf ipv6-address {
          type inet:ipv6-address;
          description "Segment IPv6 prefix";
        }
        leaf algorithm {
          type uint8;
          description "Prefix SID algorithm identifier";
        }
      }
      container segment-type-10 {
        description
           "Segment declared by IPv6 Prefix and Interface ID for
           link endpoints as Local, Remote pair for SRv6";
        leaf local-ipv6-address {
          type inet:ipv6-address;
          description "Local link IPv6 address";
        }
        leaf local-interface-identifier {
          type uint32;
          description "Local interface identifier";
        }
        leaf remote-ipv6-address {
          type inet:ipv6-address;
          description "Remote link IPv6 address";
        }
        leaf remote-interface-identifier {
          type uint32;
          description "Remote interface identifier";
        }
      }
      container segment-type-11 {
        description
           "Segment declared by IPv6 Addresses for link endpoints as
            Local, Remote pair for SRv6";
```

```
          leaf local-ipv6-address {
            type inet:ipv6-address;
            description "Segment local IPv6 adjacency address";
          }
          leaf remote-ipv6-address {
            type inet:ipv6-address;
            description "Segment remote IPv6 adjacency address";
          }
      }
    }
  }
  grouping sr-odn-policies {
    description "SR On Demand (dynamic) SR Policies";
    container installed-policies {
      config false;
      description "BGP triggered On Demand (dynamic) SR Policies
                   corresponding to the BGP nexthops";
      list sr-policy {
        key "color end-point";
        description "SR Policy list";
        uses sr-policy-color-endpoint;
      }
    }
  }

  grouping sr-policy-steering-state {
    description "Per route Automatic Steering parameters";
    container automatic-steering {
      description "Per route Automatic Steering parameters";
      leaf color {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
" +
               "bgp:bgp/bgp:global/ietf-bgp-srte:segment-routing/" +
               "ietf-bgp-srte:policy-state/ietf-bgp-srte:sr-policy/" +
               "ietf-bgp-srte:color";
        }
        description "Color of the SR Policy being used for
                     Automatic Steering";
      }
      leaf end-point {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
" +
               "bgp:bgp/bgp:global/ietf-bgp-srte:segment-routing/" +
               "ietf-bgp-srte:policy-state/ietf-bgp-srte:sr-policy/" +
               "ietf-bgp-srte:end-point";
        }
        description "End-point of the SR Policy being used
                     for Automatic Steering";
```

```
      }
      leaf co-flag {
        type enumeration {
          enum 00 {
            description "Color-Only flag 00";
          }
          enum 01 {
            description "Color-Only flag 01";
          }
          enum 10 {
            description "Color-Only flag 10";
          }
        }
        default "00";
        description "Color-Only (CO) flags applicable for
                     Automatic Steering of this route";
      }
      leaf binding-sid {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
" +
                "bgp:bgp/bgp:global/ietf-bgp-srte:segment-routing/" +
                "ietf-bgp-srte:policy-state/ietf-bgp-srte:sr-policy/" +
                "ietf-bgp-srte:binding-sid";
        }
        description "Binding SID of the SR Policy";
      }
    }
  }

  grouping route-key-leafs {
    description "Grouping for key leafs identifying a route";
    leaf prefix {
      type union {
         type inet:ip-prefix;
         type string;
      }
      description "BGP Prefix. This is a temp definition to
                   cover ip-prefix and other NLRI formats.
                   Import the type once defined in base
                   BGP RIB model";
    }
    leaf neighbor {
      type inet:ip-address;
      description "BGP Neighbor";
    }
    leaf add-path-id {
      type uint32;
      description "Add-path ID";
```

```
    }
  }

  grouping common-bgp-route-grouping {
    description "BGP route list" ;
    container routes {
      config false;
      description "BGP Route in local RIB";
      list route {
        key "prefix neighbor add-path-id";
        description "BGP route list";
        uses route-key-leafs;
      }
    }
  }

  grouping common-bgp-vpn-route-grouping {
    description "BGP route list" ;
    container routes {
      config false;
      description "BGP VPN Route in local RIB";
      list route {
        key "rd prefix neighbor add-path-id";
        description "Route List";

        leaf rd {
          type rt-types:route-distinguisher;
          description "Route Distinguisher";
        }
        uses route-key-leafs;
      }
    }
  }

  //
  // BGP Specific Paramters
  //
  // Augment AF with route list
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast" {
    description
      "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast" {
    description
      "Augment BGP SAFI route";
```

```
    uses common-bgp-route-grouping;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-labeled-unicast
" {
    description
      "Augment BGP SAFI route";
    uses common-bgp-route-grouping;

  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-labeled-unicast
" {
    description
      "Augment BGP SAFI route";
    uses common-bgp-route-grouping;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast"
{
    description
      "Augment BGP SAFI route";
    uses common-bgp-vpn-route-grouping;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast"
{
    description
      "Augment BGP SAFI route";
    uses common-bgp-vpn-route-grouping;
  }

  // SR Policy Related
  // On Demand authorized colors table
  // SR Policy state data
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global" {
    description
      "Segment Routing parameters in BGP global model";
    container segment-routing {
      description "Segment Routing parameters";
      container on-demand-policies {
        description
          "Segment Routing On Demand Nexthop
           (ODN) SR Policies";
        uses sr-odn-auth-colors;
        uses sr-odn-policies;
      }
      uses sr-policy-state;
    }
  }
```

```
  // Steering state in overlay BGP routes
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:ipv4-unicast/ietf-bgp-srte:routes/ietf-bgp-srte:route
" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:ipv6-unicast/ietf-bgp-srte:routes/ietf-bgp-srte:route
" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:ipv4-labeled-unicast/ietf-bgp-srte:routes/ietf-bgp-sr
te:route" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:ipv6-labeled-unicast/ietf-bgp-srte:routes/ietf-bgp-sr
te:route" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:l3vpn-ipv4-unicast/ietf-bgp-srte:routes/ietf-bgp-srte
:route" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/" +
          "bgp:afi-safi/bgp:l3vpn-ipv6-unicast/ietf-bgp-srte:routes/ietf-bgp-srte
:route" {
    description
      "Augment BGP SAFI route with steering info";
    uses sr-policy-steering-state;
  }

  // BGP Signaled SR Policy explicit candidate paths state
  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi" {
    description "Augment IPv4 SR Policy SAFI list entry";
```

```
    container ipv4-srpolicy {
      when "../afi-safi-name = 'bgp-types:IPV4_SRPOLICY'" {
       description
         "Include this container for IPv4 SR Policy specific
          configuration";
      }
      description "IPv4 SR Policy specific parameters";
      uses sr-exp-policy-cp-state;
    }
  }

  augment "/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/" +
          "bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi" {
    description "Augment IPv6 SR Policy SAFI list entry";
    container ipv6-srpolicy {
      when "../afi-safi-name = 'bgp-types:IPV6_SRPOLICY'" {
        description
          "Include this container for IPv6 SR Policy specific
           configuration";
      }
      description "IPv6 SR Policy specific parameters";
      uses sr-exp-policy-cp-state;
    }
  }
}
<CODE ENDS>
```

5. Contributors

Dhanendra Jain
Cisco Systems
US

Email: dhanendra.ietf@gmail.com

Zhichun Jiang
Cisco Systems
US

Email: zcjiang@tencent.com

Zafar Ali
Cisco Systems
US

Email: zali@cisco.com

Sharmila Palani
Cisco Systems
US

Email: spalani@cisco.com

6. IANA Considerations

7. Security Considerations

The transport protocol used for sending the BGP Segment Routing data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

This draft does not change any underlying security issues inherent in [I-D.ietf-idr-bgp-model].

8. Acknowledgements

TBD.

9. References

9.1. Normative References

[I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-10 (work in progress), November 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-11 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

9.2. Informative References

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

Authors' Addresses

Krishna Deevi
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: kdeevi@cisco.com

Kamran Raza
Cisco
2000 Innovation Drive
Kanata, ON K2K-3E8
CA

Email: skraza@cisco.com

Kausik Majumdar
Commscope

Email: kausik.majumdar@comscope.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Zhichun Jiang
Tencent

Email: zcjiang@tencent.com

TEAS Working Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

J. Dong
Z. Li
Huawei Technologies
F. Qin
China Mobile
G. Yang
China Telecom
J. Guichard
Futurewei Technologies

February 22, 2021

Scalability Considerations for Enhanced VPN (VPN+) draft-dong-teas-enhanced-vpn-vtn-scalability-02

Abstract

Enhanced VPN (VPN+) aims to provide enhancements to existing VPN services to support the needs of new applications, particularly including the applications that are associated with 5G services. VPN+ could be used to provide network slicing, and may also be of use in more generic scenarios, such as enterprise services which have demanding requirement. With the requirement for VPN+ services increase, scalability would become an important factor for the deployment of VPN+. This document describes the scalability considerations in the control plane and data plane to enable VPN+ services, some optimization mechanisms are also described.

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 August 26, 2021.

Table of Contents

1. Introduction

Virtual Private Networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated connectivity over a common network infrastructure. The VPN service is provided with two network layers: the overlay and the underlay. The underlay is responsible for establishing network connectivity and managing network resources to meet the service requirement. The overlay is used to distribute the membership and reachability information of the tenants, and provide logical separation of service delivery between different tenants.

Enhanced VPN service (VPN+) [I-D.ietf-teas-enhanced-vpn] is targeted at new applications which require better isolation between tenants and/or services, and have more stringent performance requirements than can be provided with existing VPNs. To meet the requirement of VPN+ services, Virtual Transport Networks (VTN) need to be created, each has a subset of the underlay network topology and a set of network resources allocated to meet the requirements of one or a group of VPN+ services. The VPN together with the corresponding VTN in the underlay provide the VPN+ service.

[I-D.ietf-teas-enhanced-vpn] provides some general analysis of the scalability of VPN+. This document gives detailed analysis of the scalability considerations when enabling VPN+ services. The focus of this document is mainly on the scalability of the underlay of VPN+, i.e. the VTN.

2. VPN+ Scalability Requirements

As described in [I-D.ietf-teas-enhanced-vpn], VPN+ services may require additional state to be introduced into the network to take advantage of the enhanced functionality. This introduces some scalability considerations to the network. This section gives some analysis of the number of VPN+ services that might be needed in a network.

There are several use cases where VPN+ may be needed, and these determine how many VPN+ will be required in a network. One typical use case of VPN+ is to deliver IETF network slice [I-D.ietf-teas-ietf-network-slice-definition] for applications or services in 5G and other scenarios, thus the number of IETF network slices needed could reflect the number of VPN+ services. With the development and evolution of 5G, it is expected that more and more network slices will be deployed. The number of network slices required is relevant to how network slicing will be used, and the progress of 5G for the vertical industrial services. The potential number of network slices is analyzed by classifying the network slicing deployment into three typical scenarios:

1. Network slicing can be used by a network operator internally to isolate different types of services. For example, in a converged multi-service network, different network slices can be created to carry mobile transport service, fixed broadband service and enterprise services respectively, each type of service could be managed by a separate department or management team. Some service types, such as multicast service may also be deployed in a dedicated network slice. It is also possible that an infrastructure network operator provides network slices to other network operators as a wholesale service. In this scenario, the

number of network slices in a network would be relatively small, such as on the order of 10 or so. This could be the typical case in the beginning of the network slicing deployment.

2. Network slicing can be used to provide isolated and customized virtual networks for tenants in different vertical industries. At the early stage of the vertical industrial service deployment, a few top tenants in some typical industries will begin to use network slicing to support their business, such as smart grid, manufacturing, public safety, on-line gaming, etc. Considering the number of the vertical industries, and the number of top tenants in each industry, the number of network slices may increase to the order of 100.

3. With the evolution of 5G, network slicing could be widely used by both vertical industrial tenants and enterprise tenants which require guaranteed or predictable service performance. The total amount of network slices may increase to the order of 1000 or more. However, it is expected that the number of network slices would still be less than the number of traditional VPN services in the network.

In 3GPP [TS23501], a 5G network slice is identified using Single Network Slice Selection Assistance Information (S-NSSAI), which is a 32-bit identifier comprised of 8-bit Slice/Service Type (SST) and 24-bit Slice Differentiator (SD). This allows the mobile networks (RAN and CN) to provide a large number of network slices. Although it is possible that multiple network slices in RAN and CN can be mapped to the same IETF network slice, the number of IETF network slices may still be comparable with the number of 5G network slices. Thus the scalability of IETF network slices needs to be taken into consideration.

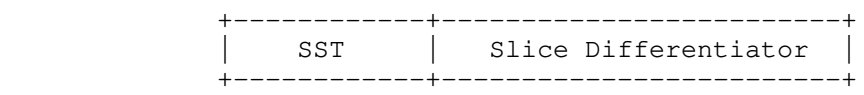


Figure 1. Format of S-NSSAI in 3GPP

VPN+ needs to meet the scalability requirement of network slicing in different scenarios. The increased number of VPN+ will introduce additional complexity and overhead to both the control plane and data plane, especially in the aspects related to the underlying VTNs. Although multiple VPN+ services can be mapped to the same VTN as the underlay, there still can be scalability challenges with the increased number of VTNs.

3. VPN+ Scalability Considerations

In this section, the scalability in the control plane and data plane is analyzed to understand the possible gaps in meeting the scalability requirement of VPN+.

3.1. Control Plane Scalability

As described in [I-D.ietf-teas-enhanced-vpn], the control plane of VPN+ could be based on the hybrid of a centralized controller and the distributed control plane.

3.1.1. Distributed Control Plane

At part of the construction of VPN+ services, it is necessary to create different VTNs that provide customized topology and resource attributes. The attributes and state information of each VTN needs to be exchanged in the control plane. The scalability of the distributed control plane for the establishment and maintenance of VTNs needs to be considered in the following aspects:

- The number of control protocol instances maintained on each node
- The number of protocol sessions maintained on each link
- The number of routes advertised by each node
- The amount of attributes associated with each route
- The number of route computation (i.e. SPF computation) executed on each node

As the number of VTNs increases, it is expected that for some of the above aspects, the overhead in the control plane may increase dramatically. For example, the overhead of maintaining separated control protocol instances (e.g. IGP instances) for different VTNs is considered higher than maintaining the information of separated VTNs in the same control protocol instance, and the overhead of maintaining separate protocol sessions for different VTNs is considered higher than using a shared protocol session for the information exchange of multiple VTNs. To meet the requirement of the increasing number of VTNs, It is suggested to choose the control plane mechanisms which could improve the scalability while still provide the required functionality.

3.1.2. Centralized Control Plane

Although the SDN approach can reduce the amount of control plane overhead in the distributed control plane, it may transfer some of the scalability concerns from network nodes to the centralized controller, thus the scalability of the controller also needs to be considered.

To provide global optimization for the Traffic Engineered (TE) paths in different VTNs, the controller needs to keep the topology and resource information of all the VTNs up to date. To achieve this, the controller may need to maintain a communication channel with each network node in the network. When there is significant change in the network, or multiple VTNs requires global optimization concurrently, there may be a heavy processing burden at the controller, and a heavy load in the network surrounding the controller for the distribution of the updated network state.

3.2. Data Plane Scalability

To provide different VPN+ services with the required isolation and performance characteristics, it is necessary to allocate different sets of network resources to different VTNs. As the number of VPN+ increases, the number of VTNs will increase accordingly. This requires the underlying network to provide finer-granular network resource partitioning, which means the amount of state about the reserved network resources to be maintained on network nodes will also increase.

In data plane, traffic of different VPN+ services need to be processed separately according to the topology and resource constraints of the associated VTN , thus the identifier of VTN needs to be carried either directly or implicitly in the data packet. Different representations of the VTN information in data packet can have different scalability implications.

One approach is to reuse some existing fields in the data packet to additionally identify the VTN the packet belongs to. This avoids the cost of defining new fields in the data packet, while since it introduces additional semantics to an existing field, it may change the processing of the existing field in packet forwarding. To distinguish different VTNs, the number of identifiers which were used to identify a node or link may be increased in proportion to the number of the VTNs, which may cause scalability problem in some networks.

An alternative approach is to introduce a dedicated field in the packet for VTN identification. This could avoid the impact to the

existing fields in the packet. And if this new field carries a global-significant VTN identifier, it could be used together with the existing fields to determine the VTN-specific packet forwarding. The potential issue with this approach is the difficulty in introducing a new field in some types of the data plane.

In addition, the introduction of per VTN packet forwarding has impact on the scalability of the forwarding entries on network nodes, as a network node needs to maintain separate forwarding entries for a target node in each VTN it participates.

3.3. Gap Analysis of Existing Mechanisms

One candidate approach to build VTN is to use Segment Routing (either SR-MPLS or SRv6) as the data plane, and define and distribute the customized topology and resource attribute of each VTN based on Multi-topology [RFC4915] [RFC5120], Flex-Algo [I-D.ietf-lsr-flex-algo] or the combination of these mechanisms in the control plane. As the number of VTNs increases, there may be several scalability concerns with this approach:

1. The number of SR SIDs needed will increase dependent upon the number of VTNs in the network, which will bring challenges both to the SID information distribution in the control plane and to the installation of forwarding entries for the SIDs in data plane.

2. The number of SPF computation will increase in proportion to the number of VTNs in the network, which can introduce significant overhead of the computing resources on network nodes.

3. The maximum number of network topology supported by OSPF Multi-topology is 128, and the maximum number of Flex-Algo is 128, which may not meet the required number of VTNs in some networks.

4. Possible Scalability Optimizations

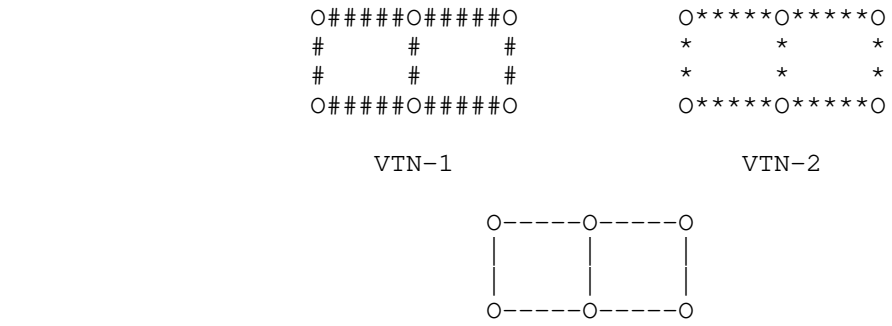
4.1. Control Plane Optimizations

For the distributed control plane, several optimizations can be considered to reduce the overhead and improve the scalability.

The first optimization mechanism is to reduce the amount of control plane sessions used for the establishment and maintenance of the VTNs. For multiple VTNs which have the same peering relationship between two adjacent network nodes, it is proposed that one single control session is used for the establishment of multiple VTNs. Information of different VTNs can be exchanged over the same control

session, with necessary identification information to distinguish them in the control messages. This could reduce the overhead of maintaining a large number of control protocol sessions, and could also reduce the amount of control plane message flooding in the network.

The second optimization mechanism is to decompose the attributes of a VTN into different groups, so that different types of attribute can be advertised and processed separately in control plane. For a VTN, there are two basic types of attributes: the topology attribute and the associated network resource attribute. In a network, it is possible that multiple VTNs share the same topology, and multiple VTNs may share the same set of network resource on particular network segments. It is more efficient if only one copy of the topology attribute is advertised, then multiple VTNs sharing the same topology could refer to the topology information. More importantly, the result of topology-based route computation could be shared by these VTNs, so that the overhead of per-VTN route computation could be reduced. Similarly, information of a subset of network resources reserved on network segments could be advertised once and then be used by multiple VTNs. This methodology could also apply to other attributes of VTN which may be introduced later and can be processed independently.



Shared Network Topology

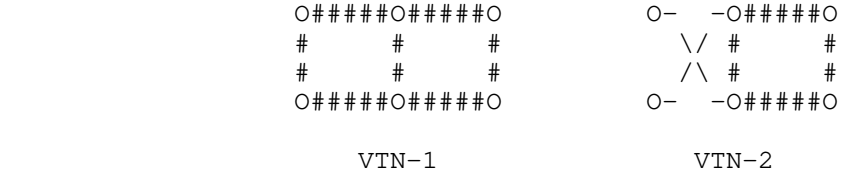
Legend

O Virtual node
Virtual links with a set of reserved resources
*** Virtual links with another set of reserved resources

Figure 2. Topology Sharing between VTNs

FIG-2

Figure 2 gives an example of multiple VTNs which share the same topology attribute. As shown in the figure, VTN-1 and VTN-2 have the same topology, while the link resource attributes of each VTN are different. In this case, only one copy of the network topology information needs to be advertised, and the topology-based route computation result can be used by both VTNs to generate the routing tables.



Legend

O Virtual node
Virtual links with a set of reserved resource
--- Virtual links with another set of reserved resource

Figure 3. Resource Sharing between VTNs

Figure 3 gives another example of multiple VTNs which shares the same set of network resources on some links. In this case, information about the reserved resource on each link only needs to be advertised once, then both VTN-1 and VTN-2 could refer to the link resource for constraint based path computation.

For the centralized control plane, it is suggested that the centralized controller is deployed as a complementary mechanism to the distributed control plane rather than a replacement, so that the VTN specific path computation burden in control plane could be shared by both the centralized controller and the network nodes, thus the scalability of both systems could be improved.

4.2. Data Plane Optimizations

To support more VPN+ services while keeping the amount of data plane state at a reasonable scale, one possible approach is to classify a set of VPN+ services which have similar service characteristics and performance requirements into a group, and such group of VPN+ is mapped to one VTN, which is allocated with an aggregated set of network topology and resources to meet the service requirement of the whole group of VPN+. Different groups of VPN+ need to be mapped to different VTNs with different set of network resources allocated. With appropriate grouping of VPN+ services, a reasonable number of

VTNs with network resources reservation and aggregation could still meet the service requirements.

Another optimization in the data plane is to decouple the identifier used for topology-based forwarding and the identifier used for the resource-specific processing introduced by VTN. One possible mechanism is to introduce a dedicated field in the packet header to uniquely identify the set of local network resources allocated to a VTN on each network node for the processing and forwarding of the received packet. Then the existing identifier in the packet header used for topology based forwarding is kept unchanged. The benefit is the number of existing topology-specific identifiers will only increase in proportion to the number of topologies rather than the number of VTNs, so that its scalability will not be impacted by the increase of VTN. Since this new VTN field will be used together with the existing fields to determine the VTN-specific packet forwarding, this probably requires network nodes to support a hierarchical forwarding table in the data plane. Figure 4 shows the concept of using different data plane identifiers for topology-based and VTN resource-based packet processing respectively.

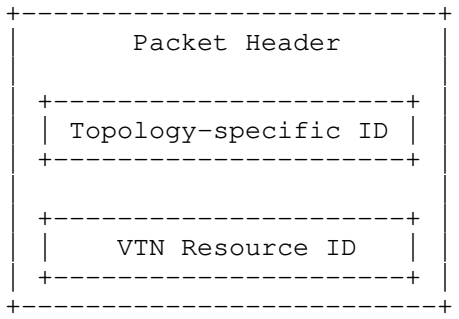


Figure 4. Decoupled Data Plane Identifiers

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or one of the extension headers to carry the VTN identifier for the resource-specific forwarding, while keeping the destination IP address field used for routing towards the destination prefix in the corresponding topology. Note that the VTN ID needs to be parsed by every node along the path which is capable of VTN-specific forwarding. In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated MPLS label to identify the VTN instance, while the existing MPLS labels could be used for topology-based packet forwarding towards the associated destination prefix. This requires that both labels be parsed by each node along the forwarding path of the packet. Another option with MPLS data plane is to

introduce a new VTN header which follows the MPLS label stack. The detailed extensions in IPv6 and MPLS encapsulation are out of the scope of this document.

5. Solution Evolution for Improved Scalability

Based on the analysis in this document, the control plane and data plane for VPN+ needs to evolve to support the increasing number of VPN+ services in the network.

As the first step, by introducing resource-awareness to segment routing SIDs [I-D.ietf-spring-resource-aware-segments], and using Multi-Topology or Flex-Algo as the control plane, it could provide a solution for building a limited number of VTNs in the network to meet the requirement of a small number of VPN+ services in the network. This mechanism is considered as the basic SR VTN.

As the number of required VPN+ services increases, more VTNs may need to be created, then the control plane scalability could be improved by decoupling the topology attribute from other attributes (e.g. resource attribute) of VTN, so that multiple VTNs could share the same topology or resource attribute. This mechanism is considered as the optimized SR VTN. Both the basic and the optimized SR VTN mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

If the data plane scalability becomes a concern, dedicated data plane VTN identifiers can be introduced to decouple the topology-specific identifiers from the VTN-specific resource identifier in the data plane, this could help to reduce the number of SR SIDs needed to support . This mechanism is considered as resource-independent VTNs.

6. Security Considerations

TBD

7. IANA Considerations

This document makes no request of IANA.

8. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

9. Acknowledgments

The authors would like to thank Adrian Farrel for the review and discussion of this document.

10. Informative References

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[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", draft-ietf-spring-resource-aware-segments-01 (work in progress), January 2021.

[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", February 2021, <https://tools.ietf.org/html/draft-ietf-spring-sr-for-enhanced-vpn>.

[I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-06 (work in progress), July 2020.

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[TS23501] "3GPP TS23.501", 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

Authors' Addresses

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: jie.dong@huawei.com

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

Email: lizhenbin@huawei.com

Fengwei Qin
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing
China

Email: qinfengwei@chinamobile.com

Guangming Yang
China Telecom
No.109 West Zhongshan Ave., Tianhe District
Guangzhou
China

Email: yangguangm@chinatelecom.cn

James N Guichard
Futurewei Technologies
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@futurewei.com

BESS WorkGroup
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

D. Rao
S. Agrawal
C. Filsfils
K. Talaulikar
Cisco Systems
B. Decraene
Orange
D. Steinberg
Steinberg Consulting
L. Jalil
Verizon
J. Guichard
Futurewei
W. Henderickx
Nokia

February 22, 2021

BGP Color-Aware Routing Problem Statement
draft-dskc-bess-bgp-car-problem-statement-02

Abstract

This document explores the scope, use-cases and requirements for a BGP based routing solution to establish end-to-end intent-aware paths across a multi-domain service provider network environment.

Table of Contents

1. Introduction

1.1. Objective

This document explores the scope, use-cases and requirements for a BGP based routing solution to establish end-to-end intent-aware paths across a multi-domain service provider network environment.

The targeted design outcome is to define the technology and protocol extensions that may be required in a manner that addresses the widest application.

To introduce the problem space that the document focuses on, let us start with the BGP-based delivery of an intent across several SR-MPLS/MPLS domains.

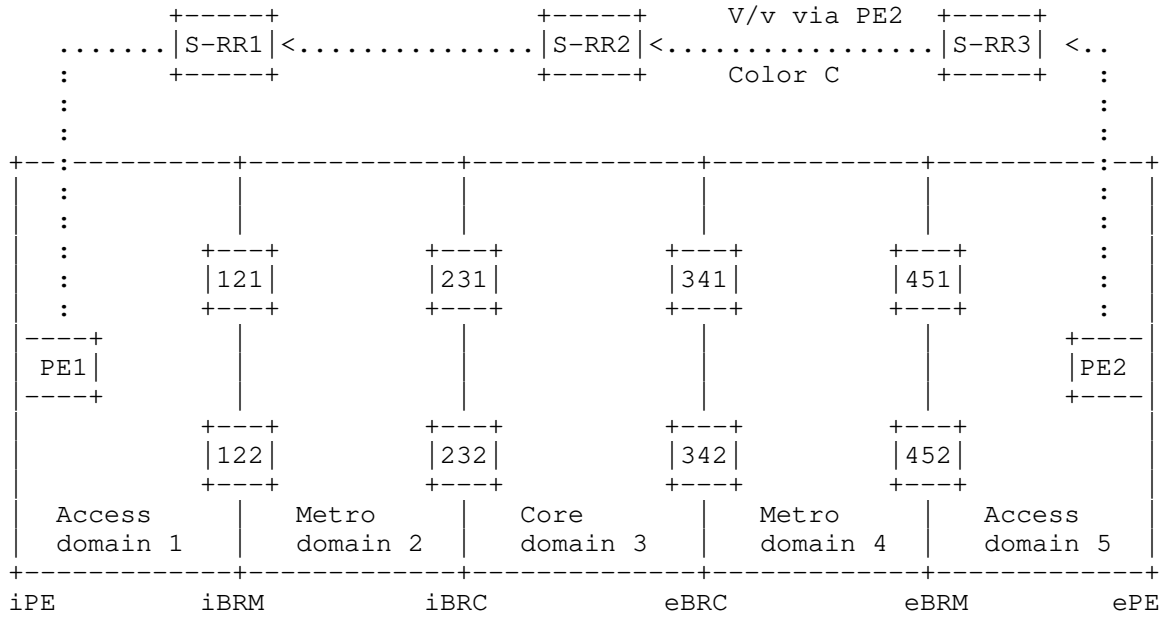


Figure 1: Reference large-scale multi-domain network topology

The figure above shows a reference large-scale multi-domain network topology. PE1 and PE2 are PEs; the other nodes are border routers

(BR) between domains in different tiers of the network. A VPN route is advertised via service RRs (S-RR) between an egress PE (PE2) and an ingress PE (PE1).

BGP must provide reachability from PE1 to PE2 based on various intent. For instance, BGP may provide reachability to PE2 using either low latency or best effort.

A VPN route having a requirement of low latency routing will select the BGP reachability information to PE2 that is based on low latency.

The problem space is then widened to include any intent (including NFV chains and their location), any dataplane and the application of the intent-based routing to the Service/VPN routes. All of this is detailed in the rest of the document.

1.2. State-of-the-art

The following solution is widely deployed - [I-D.ietf-spring-segment-routing-policy]:

- In reference figure above, an Egress PE PE2 advertises a BGP VPN route V/v with a BGP Color Extended Community C [I-D.ietf-idr-tunnel-encaps] to indicate the service intent that PE2 requests for the traffic bound to V/v. Note: The Color Extended Community may be applied to any BGP service route. For simplicity in this document, we will use a VPN route example.

- An ingress PE1 steers V-destined packets onto an SR Policy bound to (C, PE2).

- C may express any of the following requirements:

  * Minimization of a cost metric vs a latency metric.

  * Exclusion/Inclusion of SRLG and/or Link Affinity.

    + In inter-domain context, exclusion/inclusion of entire domains.

  * Inclusion of virtual network function chains [I-D.ietf-spring-sr-service-programming].

- An SR-PCE (or a set of them) computes the end-to-end path and installs it at PE1 as an SR Policy. The end-to-end path may seamlessly cross multiple domains.

The SR-PCE solution being defined at the IETF [RFC8664]and being widely deployed is reminded in this introduction as a useful "state-of-the-art" context to consider when defining the BGP-based alternative solution.

1.2.1. Color

The solution must reuse the Color concept defined in [I-D.ietf-spring-segment-routing-policy]. The color is a 32-bit numerical value that, today, associates an SR-policy with an intent (e.g., low latency).

1.2.2. Colored vs Color-Aware

The solution must support the ability to distinguish BGP routes that require the usage of a particular intent from BGP routes that are actually satisfying a particular intent. Hence, this document defines the notion of colored and color-aware routes.

- Colored: Egress PE PE2 colored its BGP VPN route V/v to indicate the intent that it requests for the traffic bound to V/v.
- Color-Aware: A new BGP solution which signals multiple "ways" to reach a given destination (e.g. PE2)
- Steering a colored VPN route to a color-aware route
  - If PE2 signals a VPN route V/v with color C
  - If PE1 installs that VPN route
  - If PE1 learns about a BGP Color-Aware Route R/r to PE2 for color C
  - Then PE1 steers packets destined to V/v via R/r
- Note the similarity with the state-of-the-art reference:
  - The steering onto an SR Policy bound to (C, PE2) is replaced by the steering on a Color-Aware BGP route (C, PE2)
  - The data model is the same "resolution via (C, PE2)"
  - The difference is how the (C, PE2) path is obtained: BGP signaling vs SR-PCE signaling

1.2.3. Per-Destination and Per-Flow Steering

Ingress PE PE1 steers packets destined for a service (VPN) route V/v via BGP Color-Aware Route R/r to PE2

- Per-Destination Steering: Incoming packets on PE1 match BGP service route V/v to be steered based on the destination IP address of the packets.

- Per-Flow Steering: Incoming packets on PE1 match BGP service route V/v to be steered based on the combination of the destination IP address and additional elements in the packet header (i.e., IP flow). Such a packet lookup may recurse on a forwarding array where some of the entries are BGP color-aware routes to PE2. A given flow is mapped to a specific entry in this array i.e. via a specific BGP color-aware route to PE2.

1.3. Why a BGP-based alternative is needed

- An operator with an existing Seamless-MPLS/BGP-LU deployment [I-D.ietf-mpls-seamless-mpls] may consider a BGP based extension as a more incremental approach.

- There may be an expectation that BGP would support a larger scale.

- Opacity of a remote domain due to trust boundaries within an inter-domain construction.

1.4. Color Domains

With the use of Color to represent intent, it is useful to describe the concept of a color domain distinct from a network domain.

- Domain: A domain (or network domain) refers to a unit of isolation or hierarchy in the network topology; for example, access, metro and or core domains. From a routing perspective, a domain may have a distinct IGP area or instance; or a distinct BGP ASN.

- Color Domain: A color domain may represent a collection of one or more network domains with a single, consistent color/intent mapping.

- Color re-mapping may happen at color domain boundaries.

- Deployments under a single authority are expected to use the same color/intent mapping across all network domains.

A solution must distinguish the actual protocol boundaries (IGP, ASN) from the color domain boundaries.

1.5. BGP Color-Aware Routing

A BGP solution that is solving this problem statement is called BGP Color-Aware Routing, and is referred to as BGP CAR in this document.

2. Intent bound to a Color

The BGP CAR solution must support the following intents bound to a color:

- Minimization of a cost metric vs a latency metric
  - Minimization of different metric types, static and dynamic
- Exclusion/Inclusion of SRLG and/or Link Affinity and/or minimum MTU/number of hops
- Bandwidth management
- In the inter-domain context, exclusion/inclusion of entire domains, and border routers
- Inclusion of one or several virtual network function chains
  - Located in a regional domain and/or core domain, in a DC
- Localization of the virtual network function chains
  - Some functions may be desired in the regional DC or vice versa
- Per-Destination and Per-Flow steering

3. BGP CAR Use-cases

The BGP CAR route may be a transport route or a service route (in this document, we use the term VPN instead of service for simplicity).

3.1. BGP Transport CAR

- Transport Intent
  - Intent-aware routing between PEs connected across multiple transit domains

+ Set up BGP based end-to-end paths stitching intent-aware intra-domain segments

o The network diagram below illustrates the reference network topology used in this section for Transport CAR:

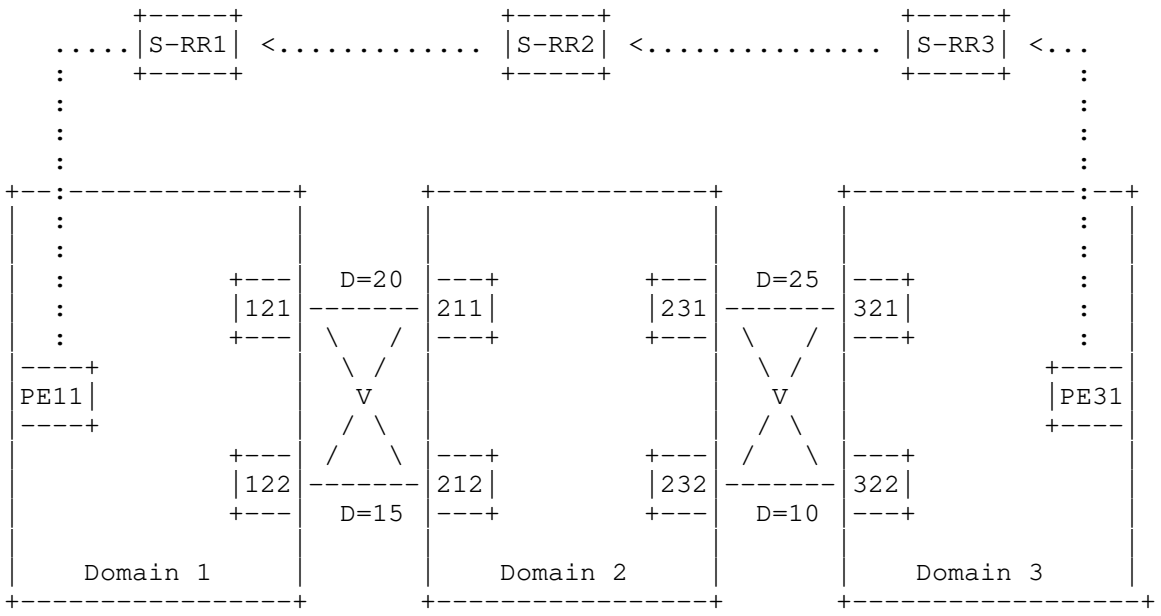


Figure 2: Transport CAR Reference Topology

The following network design assumptions apply to the reference topology above, as an example:

* Independent ISIS/OSPF SR instance in each domain.

* eBGP peering link between ASBRs (121-211, 121-212, 122-211, 122-212, 231-321, 231-322, 232-321 and 232-322).

* Peering links have equal cost metric.

* Peering links have delay configured or measured as shown by "D". D=50 for cross peering links.

* VPN service is running from PE31 to PE11 via service RRs (S-RRn in figure).

o The following sections illustrate a few examples of intent use-cases applicable to transport routes.

3.1.1. Use-case of minimization of a cost metric vs a latency metric

- In the reference topology of Figure 2

  Each domain has Algo 0 and Flex Algo 128

  Algo 0 is for minimum cost metric(cost optimized).

  Flex Algo 128 definition is for minimum delay (low latency).

- Cost Optimized

  * Color C1 - Minimum cost intent. (Here, a BGP CAR route with Color C1 is being used, instead of BGP-LU.)

  * On PE11, VPN routes colored with C1 are steered via (C1, PE31) BGP CAR route

    + BGP CAR for C1 sets up path(s) between PEs for end-to-end minimum cost.

    + (2) These paths traverse over intra-domain Algo 0 in each domain and account for the peering link cost between ASBRs.

    + Example: PE11 learns (C1, PE31) CAR route via several equal paths:

      1. One such path is through FA0 to node 121, links 121-211, FA0 to 231, link 231-321, FA0 to PE31

      2. Another such path is through FA0 to node 122, link 122-212, FA0 to 232, link 232-322, FA0 to PE31.

- Minimize latency

  * Color C2 - Minimum latency intent.

  * On PE11, VPN routes colored with C2 are steered via (C2, PE31) BGP CAR route.

    + BGP CAR for C2 advertises paths between PEs for minimum end-to-end delay.

    + (2) These paths traverse over intra-domain Flex Algo 128 in each domain and account for peering link delay between ASBRs.

+ (3) Example: PE11 learns (C2, PE31) BGP CAR route and best path is through FA128 to node 122, link 122-212, FA128 to 232, link 232-322, FA128 to PE31.

3.1.2. Use-case of exclusion/inclusion of link affinity

- Color C3 - Intent to Minimize cost metric and avoid purple links
- In the reference topology of Figure 2

  Each domain has Flex Algo 129 and some links have purple affinity.

  Flex Algo 129 definition is set to minimum cost metric and avoid purple links (within domain).

  Peering cross links are colored purple by policy.

- On PE11, VPN routes colored with C3 are steered via (C3, PE31) BGP CAR route.
  * BGP CAR for C3 sets up paths between PEs for minimum end-to-end cost and avoiding purple link affinity.
  * These paths traverse over intra domain Flex Algo 129 in each domain and accounts for peering link cost between ASBR and avoiding purple links.
  * Example: PE11 learns (C3, PE31) BGP CAR route via 2 paths.
    1. First path is through FA 129 to node 121, link 121-211, FA129 to 231, link 231-321, FA129 to PE31.
    2. Second path is through FA129 to node 122, link 122-212, FA129 to 232, link 232-322, FA129 to PE31.

3.1.3. Use-case of exclusion/inclusion of domains

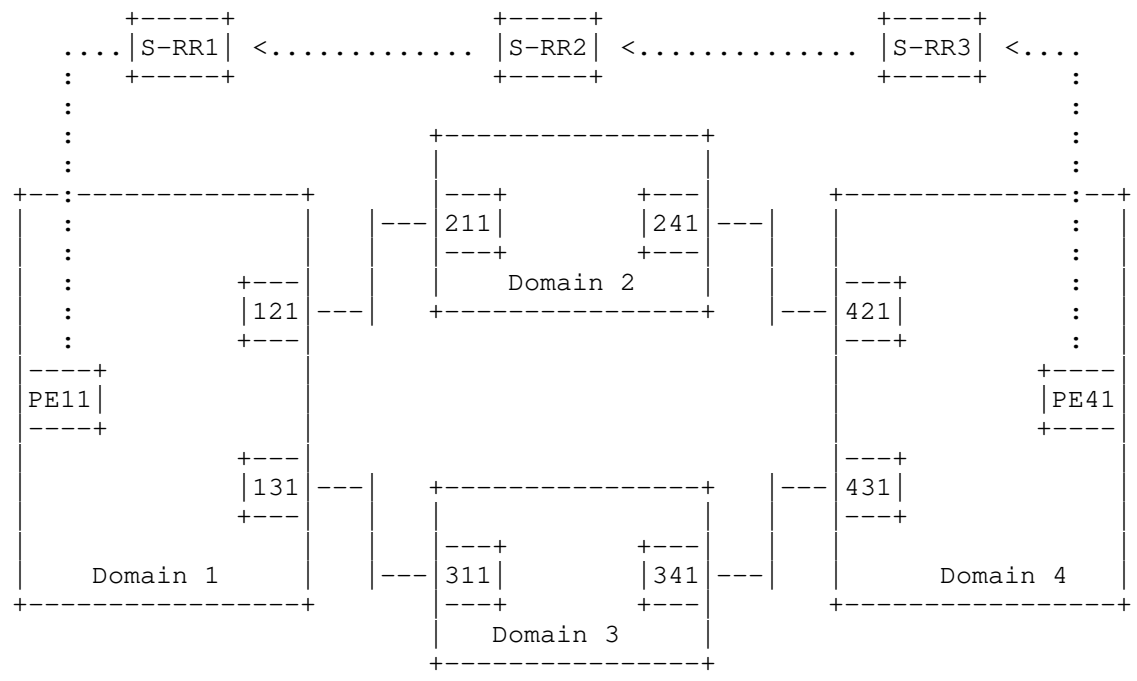


Figure 3

Color C4 - Avoid sending selected traffic via Domain 3

- VPN routes advertised from PEs with Color C4
- BGP CAR for Color C4 should only set up paths between PE11 and PE41 that exclude Domain 3

3.1.4. Use-case of virtual network function chains in local and core domains

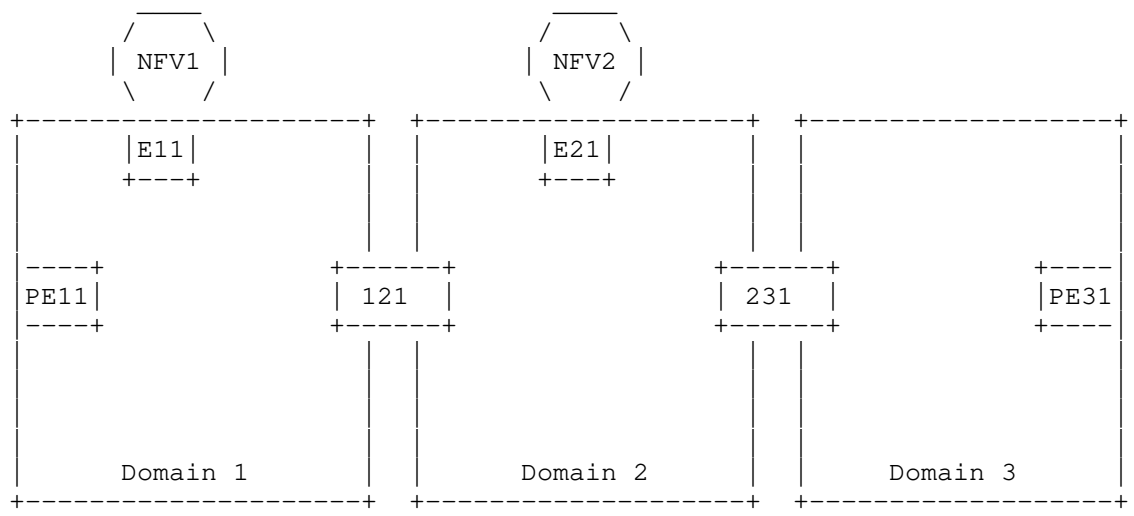


Figure 4

- Color intent
  - C5 - Routing via min-cost paths
  - C6 - Routing via a local NFV service chain situated at E11
  - C7 - Routing via a centrally located NFV service chain situated at E21
- Forwarding of packets from PE11 towards PE31:
  - (C5, PE31) mapped packets are sent via nodes 121, 231 to PE31
  - (C6, PE31) mapped packets are sent to E11 and then post-service chain, via 121, 231 to PE31
  - (C7, PE31) mapped packets are sent via 121 to E21 and then post-service chain, via 231 to PE31

3.2. BGP VPN CAR

- VPN (Service layer) intent
  - Extend the signaling of intent awareness end-to-end: CE site to CE site across provider networks

+ Provide ability for a CE to select paths through specific PEs for a given intent

  - Example-1: Certain intent in transport not available via specific PEs

  - Example-2: Certain CE-PE connection does not support specific intent

  - Example-3: Site access via certain CE does not support specific intent. For instance, link connecting a specific CE to a DC hosting loss-sensitive service may have better quality than a link from another CE

+ Provide ability for a CE to send traffic indicating a specific intent (via suitable encapsulation) to the PE for optimal steering.

* Intent aware routing support for multiple service (VPN) interworking models

  + Beyond options such as iBGP or Inter-AS Option C that inherently extend from PE to PE

    1. Inter-AS Option A

    2. Inter-AS Option B

    3. GW based interworking(L3VPN, EVPN)

  + Interworking with existing L3VPN deployments, both PEs and CEs

o The network diagram below illustrates the reference network topology used in this section for VPN CAR.

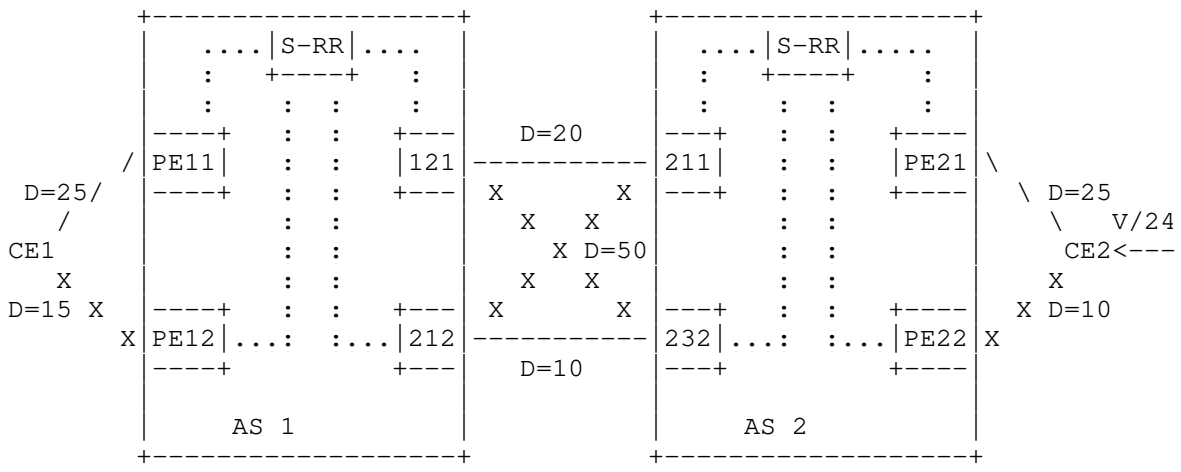


Figure 5: VPN CAR reference topology

The following network design assumptions apply to the reference topology above, as an example:

* Independent ISIS/OSPF SR instance in each AS.

* eBGP peering link between VPN ASBRs 121-211, 121-212, 122-211, 122-212.

* VPN service is running between PEs via service RRs in each AS to local ASBRs. Between ASBRs, its Option-B i.e. next hop self for VPN SAFI.

* CE1 is dual homed to PE11 and PE12. Similarly, CE2 is dual homed to PE21 and PE22.

* Peering links have equal cost metric

* Peering links have delay configured or measured as shown by "D".

* CE2 advertises prefix V/24 to CE1. It is advertised as RD:V/24 between PEs, including color-awareness

o The following sections illustrate a few examples of intent use-cases applicable to VPN (service) routes.

3.2.1. Use-case of minimization of a cost metric vs a latency metric

- In the reference topology of Figure 5

  Each AS has Flex Algo 0 and 128.

  Flex Algo 0 is for minimum cost metric(cost optimized).

  Flex Algo 128 definition is for minimum delay (low latency).

- Cost Optimized
  - Color C1 - Minimum cost intent.
  - On CE1, flows requiring cost optimized paths to V/24 are steered over (C1, V/24) route.
    - BGP CAR for C1 sets up paths between CEs for minimum end-to-end cost.
    - This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C1 awareness.
    - It also needs BGP VPN CAR between PEs and ASBRs for RD:V/24 prefix and color C1 awareness (C1, RD:V/24).
    - Paths traverse over PE-CE links, intra-domain Flex Algo 0 in each AS and peering links between ASBRs, minimizing cost for VPN.
    - Example: CE1 learns (C1, V/24) CAR route through several equal cost paths:
      1. One path is through link CE1-PE11, FA0 to 121, link 121-211, FA0 to PE21 and link PE21-CE2.
      2. Another such path is through CE1-PE12, FA0 to node 122, link 122-212, FA0 to PE22, link PE22-CE2.
- Minimize latency
  - Color C2 - Minimum latency intent
  - On CE1, flows requiring low latency paths to prefix V/24 are steered over (C2, V/24) CAR route.
    - BGP CAR for C2 sets up paths between CEs for minimum end-to-end delay.

+ This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C2 awareness.

+ It also needs BGP VPN CAR between PEs and ASBR for RD:V/24 prefix and color C2 awareness (C2, RD:V/24).

+ Paths traverse over intra-domain Flex Algo 128 in each AS and accounts for inter ASBR link delays and PE-CE link delays for the VPN.

+ Example: CE1 learns (C2, V/24) CAR best route through link CE1-PE12, FA128 to 122, link 122-212, FA128 to PE22 and link PE22-CE2.

3.2.2. Use-case of exclusion/inclusion of link affinity

- Color C3 - Intent to Minimize cost metric and avoid purple links

- In the reference topology of Figure 5

  Each AS has Flex Algo 129 and some links have purple affinity.

  Flex Algo 129 definition is set to minimum cost metric and avoid purple links (within AS).

  ASBR cross links are colored purple by policy. Bottom PE-CE links are colored purple as well by policy

- On CE1, flows requiring minimum cost path avoiding purple links to V/24 are steered over (C3, V/24) BGP CAR route.

  * BGP CAR for C3 setup paths between CEs for minimum end-to-end cost and avoiding purple link affinity.

  * This advertisement needs BGP CAR between PE-CE for V/24 prefix and color C3 awareness

  * It also needs BGP VPN CAR between PEs and ASBRs for RD:V/24 prefix and color C3 awareness (C3, RD:V/24).

  * The path avoids purple PE-CE links, traverses over intra-domain Flex Algo 129 in each AS and avoids purple links between VPN ASBRs.

  * Example: CE1 learns (C3, V/24) CAR route through link CE1-PE11, FA129 to 121, link 121-211, FA129 to PE21 and link PE21-CE2.

3.2.3. Use-case of virtual network function chains in local and core domains

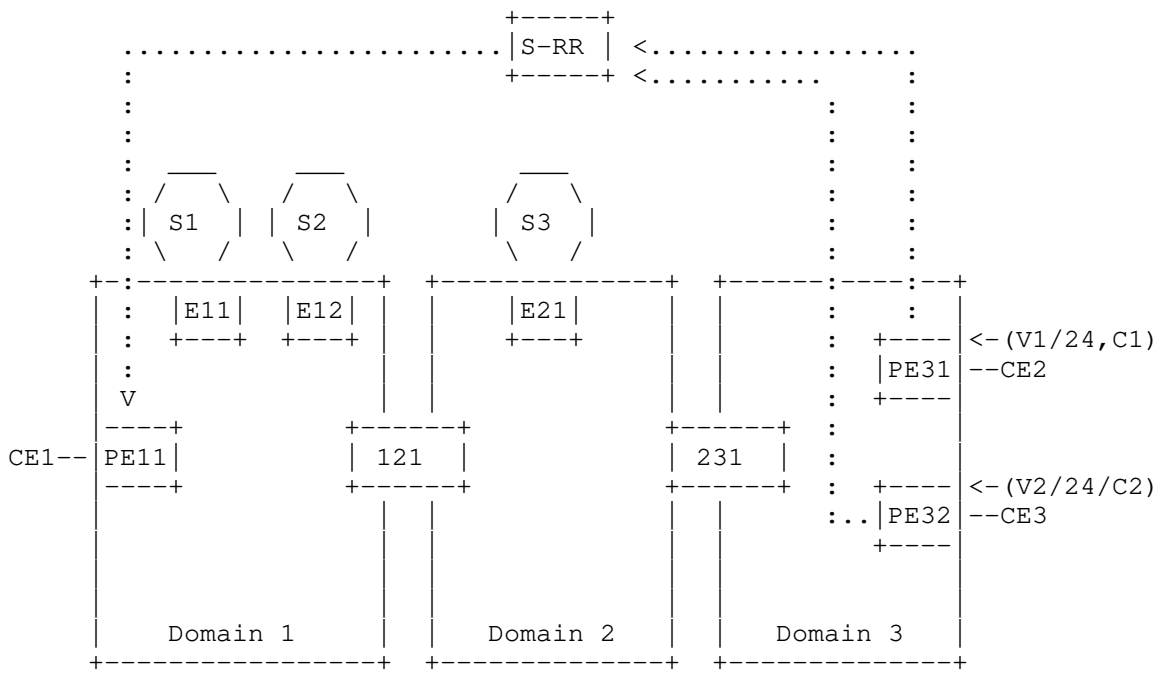


Figure 6

- Color intent
  - C1 - Routing via NFV service chain comprising of [S1, S2] attached to E11 and E12
  - C2 - Routing via NFV service [S3] attached to E21
- CE1, CE2, CE3 are sites of VPN1.
- Prefix V1/24 colored with C1 from CE2, and advertised as RD:V1/24 with C1 by PE31 to PE11 via S-RR
- Prefix V2/24 colored with C2 from CE3, and advertised as RD:V2/24 with C2 by PE32 to PE11 via SS-RR
- From PE11:

* [V1/24, C1] mapped packets are sent via S1, S2 and then routed to PE31, CE2

* [V2/24, C2] mapped packets are sent via S3 and then routed to PE32, CE3

4. Deployment Requirements

- Co-existence, compatibility and interworking with currently deployed SR-PCE based multi-domain color-aware solution

- Support different multi-domain deployment designs

  * Multiple IGP domains within a single AS (Seamless MPLS)

    + Inter-connect at node level (ABR)

  * Multiple BGP AS domains

    + Inter-connect via peering links (ASBR)

- Support end-to-end path crossing transport domains with different technologies and encapsulations

  * LDP-MPLS

  * RSVP-TE-MPLS

  * SR-MPLS

  * SRv6

  * IPv4/IPv6

- Support interworking between domains with different encapsulations (e.g, SR-MPLS and SRv6)

- Support multiple transport encapsulations within a domain for co-existence and migration

- Provide a BGP-based control-plane solution for the use-case illustrated in [RFC8604] together with deployment design guidelines for the leverage of anycast and binding SIDs.

5. Scalability

5.1. Scale Requirements

- Support for massive scaled transport network
  - Number of Remote PE's: >= 300k
  - Number of Colors C: >= 5
- Scalable MPLS dataplane solution
  - With one label per (C, Remote PE), the 1M MPLS dataplane does not work.
  - A notion of hierarchy or segment list is required.
    - E.g. the SR-PCE builds the end-to-end path as a list of segments such that no single node needs to support a data-plane scaling in the order of (Remote PE * C)
    - The solution is thus not a direct extension of BGP-LU
  - Additionally, PE and transit nodes (ABRs) may be devices with limited forwarding table space
  - Devices may have constraints on packet processing (e.g., label operations, number of labels pushed) and performance
- Ability to abstract the topology from remote domains - for scale, stability and faster convergence
  - Abstracting PE and/or ABR related state and network events
- Support for an Emulated-PULL model for the BGP signaling
  - The SR-PCE solution natively supports a PULL model: when PE1 installs a VPN route V/v via (C, PE2), PE1 requests its serving SR-PCE to compute the SR Policy to (C, PE2). I.e. PE1 does not learn unneeded SR policies.
  - BGP Signaling is natively a PUSH model.
  - Emulated-PULL refers to the ability for a BGP CAR node PE1 to "subscribe" to (C, PE2) route such that only the related paths are signaled to PE1.

* The subscription and related filtering solution must apply to any BGP CAR node

  + Transport CAR routes

    1. Ability for a node (PE/ABR/RR) to signal interest for routes of specific colors.

    2. PEs only learn routes that they need - remote VPN endpoints (PEs/ASBRs) or transit nodes (ABRs, ASBRs).

    3. ABRs also only learn and propagate routes they need locally in domain

  + Service/VPN CAR routes

    1. Ability for a node (PE) to signal interest for a specific (Egress PE, Color) transport route

    2. CEs learn routes that they need - interested colors

    3. PEs learn routes that they need - interested VPNs, colors

  + Automation of the subscription/filter route

    1. Similar to the SR-PCE solution, when an ingress PE1 installs VPN V/v via (C, PE2), PE1 originates its subscription/filter route for (C, PE2).

  + Efficient propagation and processing of subscription/filter routes.

  + Ability to perform aggregation and suppression of subscription/filter routes at nodes in the route propagation path to reduce explosion and churn in propagation of the filter routes themselves.

  + The solution may be optional for networks that do not have the large scaling requirements

5.2. Scale Analysis

It is useful to analyze the multiple scaling requirements and specifically the data plane constraints in the context of a few common reference designs and use-cases.

A couple of example scenarios are listed below for reference.

o Seamless-MPLS design, with IGP Flex-Algo in each domain

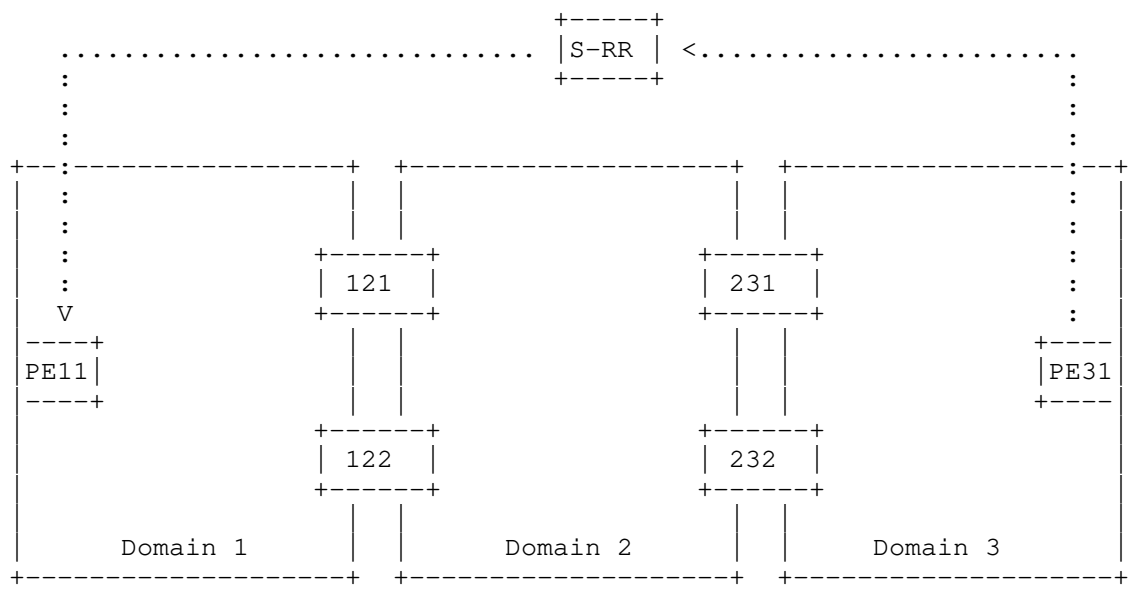


Figure 7

o Inter-AS Option C VPN design, with IGP Flex-Algo in each domain, and eBGP peering between domains

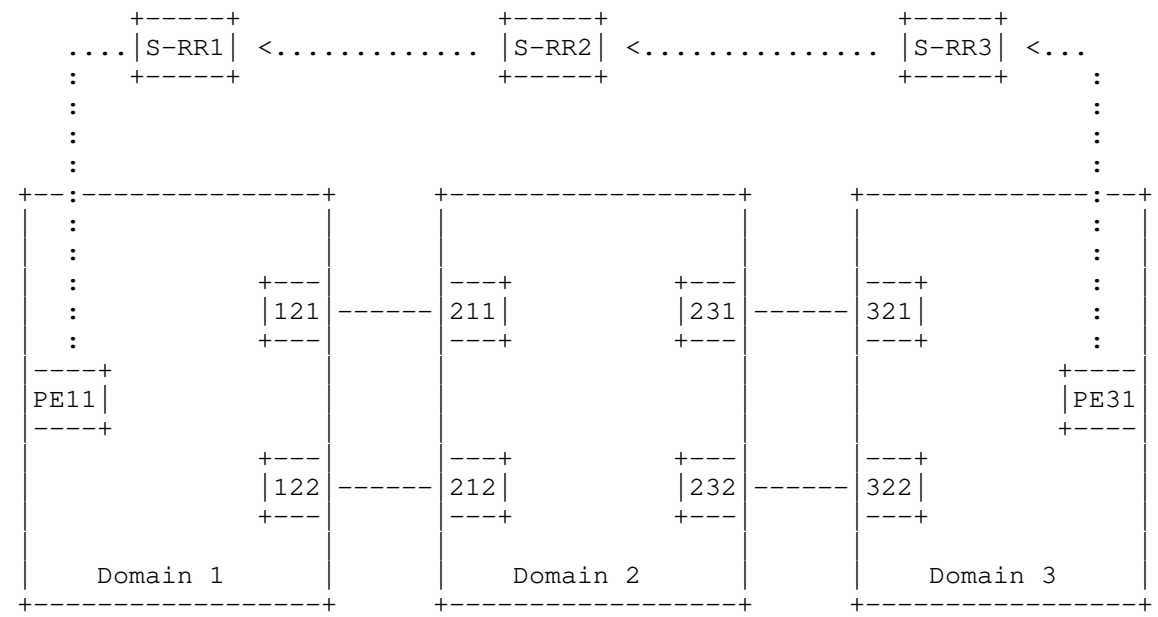


Figure 8

6. Network Availability

- The BGP CAR solution should provide high network availability for typical deployment topologies, with minimum loss of connectivity in different network failure scenarios.

- The network failure scenarios, applicable technologies and design options described in [I-D.ietf-mpls-seamless-mpls] should be used as a reference.

- In the Seamless-MPLS reference topology in previous section:

  * Failure of intra-domain links should limit loss of connectivity (LoC) to < 50ms. E.g., PE11 to a P node (not shown), 121 to a P node in Domain1 or Domain2)

  * Failure of an intra-domain node (P node in any domain) should limit LoC to < 50ms

  * Failure of an ABR node (e.g., 121, 231) should limit LoC to < 1sec

    * Failure of a remote PE node (e.g., PE3) should limit LoC to < 1sec

  o In the Inter-AS Option C VPN reference topology in previous section:

    * Failure of intra-domain links should limit LoC to < 50ms. E.g., PE11 to a P node (not shown), 121 to a P node in Domain1 or Domain2)

    * Failure of an intra-domain node (P node in any domain) should limit LoC to < 50ms

    * Failure of an ASBR node (e.g., 121, 211) should limit LoC to < 1sec

    * Failure of a remote PE node (e.g., PE3) should limit LoC to < 1sec

    * Failure of an external link (e.g., 121-211) should limit LoC to < 1sec

  o The solution should explore and describe additional techniques and design options that are applicable to further improve handling of the failure cases listed above.

7. BGP Protocol Requirements

  o Support signaling and distribution of different Color-Aware routes to reach a participating node, e.g., a PE. Intent should be indicated by the notion of a Color as defined in SR Policy Architecture.

    * Signal different instances of a prefix distinguished by color

    * Signal intent associated with a given route

  o Support for a flexible NLRI definition to accommodate both efficiency of processing (e.g., packing) and future extensibility

    * Avoid limitations associated with existing SAFI NLRI definitions. For example, 24-bit label.

  o Support for validation of paths

    * Reachability of next-hop in control plane

    * Availability and programming of encapsulation in data plane

    * Validation of intent

  o Next-hop resolution for Color-Aware route

    * Flexibility to use different intra-domain and inter-domain mechanisms - IGP-FA, SR-TE, RSVP-TE, IGP, BGP-LU etc.

    * Recursive resolution over BGP Color-Aware routes

    * Ability to carry end-to-end cumulative metric for a given color

    * Support setting up an end-to-end Color-Aware path using a different/less preferred or best-effort paths in domains where a particular intent is not available

  o Separation of transport and VPN service semantics.

    * Allow for different route distribution planes for service vs transport routes.

  o Support signaling of different transport encapsulations

  o Support for signaling multiple encapsulations for co-existence and migration

  o Generation of BGP Color-Aware routes sourced from IGP-FA, SR-TE policies and BGP-LU from a domain

  o Support signaling across domains with different color mappings for a given intent.

8. Future Considerations

Multicast service intent

9. Acknowledgements

Many people contributed to this document.

The authors would especially like to thank Jim Uttaro for his guidance on the work and feedback on many aspects of the problem statement. We would also like to thank Daniel Voyer, Luay Jalil and Robert Raszuk for their review and valuable suggestions.

We also express our appreciation to Bruno Decreane, Keyur Patel, Jim Guichard, Alex Bogdanov, Dirk Steinberg, Hannes Gredler and Xiaohu Hu for discussions on several topics that have helped provide input to

the document. We also thank Huaimo Chen for his valuable review comments.

The authors would like to thank Stephane Litkowski for his detailed review and for making valuable suggestions to improve the quality of the document. We would also like to thank Kamran Raza and Kris Michelson for their review and comments on the document and to Simon Spraggs, Jose Liste and Jiri Chaloupka for their early inputs on the problem statement.

10. References

10.1. Normative References

[I-D.agrawal-spring-srv6-mpls-interworking]
Agrawal, S., Ali, Z., Filsfils, C., Voyer, D., and Z. Li, "SRv6 and MPLS interworking", draft-agrawal-spring-srv6-mpls-interworking-03 (work in progress), August 2020.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-idr-bgp-ipv6-rt-constrain]
Patel, K., Raszuk, R., Djernaes, M., Dong, J., and M. Chen, "IPv6 Extensions for Route Target Distribution", draft-ietf-idr-bgp-ipv6-rt-constrain-12 (work in progress), April 2018.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-21 (work in progress), January 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.voyer-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (work in progress), July 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <https://www.rfc-editor.org/info/rfc4360>.

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <https://www.rfc-editor.org/info/rfc4684>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <https://www.rfc-editor.org/info/rfc5512>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <https://www.rfc-editor.org/info/rfc5701>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

10.2. Informative References

[I-D.filsfils-spring-sr-policy-considerations]
Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M., and P. Mattes, "SR Policy Implementation and Deployment Considerations", draft-filsfils-spring-sr-policy-considerations-06 (work in progress), October 2020.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-03 (work in progress), December 2020.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[RFC3906] Shen, N. and H. Smit, "Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels", RFC 3906, DOI 10.17487/RFC3906, October 2004, <https://www.rfc-editor.org/info/rfc3906>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <https://www.rfc-editor.org/info/rfc4272>.

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

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <https://www.rfc-editor.org/info/rfc6952>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/info/rfc7911>.

Authors' Addresses

Dhananjaya Rao
Cisco Systems
USA

Email: dhrao@cisco.com

Swadesh Agrawal
Cisco Systems
USA

Email: swaagraw@cisco.com

Clarence Filsfils
Cisco Systems
Belgium

Email: cfilsfil@cisco.com

Ketan Talaulikar
Cisco Systems
India

Email: ketant@cisco.com

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Dirk Steinberg
Steinberg Consulting
Germany

Email: dws@dirksteinberg.de

Luay Jalil
Verizon
USA

Email: luay.jalil@verizon.com

Jim Guichard
Futurewei
USA

Email: james.n.guichard@futurewei.com

Wim Henderickx
Nokia
Belgium

Email: wim.henderickx@nokia.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 23, 2021

G. Fioccola
T. Zhou
Huawei
M. Cociglio
Telecom Italia

January 19, 2021

Segment Routing Header encapsulation for Alternate Marking Method
draft-fz-spring-srv6-alt-mark-00

Abstract

This document describes how the Alternate Marking Method can be used as the passive performance measurement tool in an SRv6 network. It defines how Alternate Marking data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header.

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 July 23, 2021.

Table of Contents

1. Introduction

[RFC8321] and [RFC8889] describe a passive performance measurement method, which can be used to measure packet loss, latency and jitter on live traffic. Since this method is based on marking consecutive batches of packets, the method is often referred as Alternate Marking Method.

This document defines how the Alternate Marking Method ([RFC8321]) can be used to measure packet loss and delay metrics for Segment Routing with IPv6 data plane (SRv6).

[RFC8754] defines the Segment Routing Header (SRH) and how it is used by nodes that are Segment Routing (SR) capable.

[I-D.fioccola-v6ops-ipv6-alt-mark] reported a summary on the possible implementation options for the application of the Alternate Marking Method in an IPv6 domain. [I-D.ietf-6man-ipv6-alt-mark] defines a new TLV that can be encoded in the Option Headers (both Hop-by-hop or Destination) for the purpose of the Alternate Marking Method application in an IPv6 domain.

This document defines how Alternate Marking data is carried as SRH TLV, that can be can be piggybacked in the packet and transported as part of the SRH. The usage of SRH TLV is introduced in [RFC8754].

2. Application of the Alternate Marking to SRv6

The Alternate Marking Method requires a marking field. A possibility is already offered by [I-D.ietf-6man-ipv6-alt-mark] while the use of a new TLV to be encoded in the SRH is defined in this document.

Since [I-D.ietf-6man-ipv6-alt-mark] defines the IPv6 Application of the Alternate Marking Method through both Hop-by-Hop and Destination Options Header, it is applicable also to SRv6 network. Indeed the use of Destination Option Header carrying Alternate Marking bits coupled with SRH allows to monitor every node along the SR path.

This document introduces the SRH TLV carrying Alternate Marking bits and this can be a preferred approach in case of SRv6 network since it does not rely on the use of Destination Option Header.

The optimization of both implementation and scaling of the Alternate Marking Method is also considered and a way to identify flows is required. The Flow Monitoring Identification field (FlowMonID), as introduced in the next sections, goes in this direction and it is used to identify a monitored flow.

Note that the FlowMonID is different from the Flow Label field of the IPv6 Header ([RFC8200]). Flow Label is used for application service, like load-balancing/equal cost multi-path (LB/ECMP) and QoS. Instead, FlowMonID is only used to identify the monitored flow. The reuse of flow label field for identifying monitored flows is not considered since it may change the application intent and forwarding behaviour. Furthermore the flow label may be changed en route and this may also violate the measurement task. Those reasons make the definition of the FlowMonID necessary for IPv6. Flow Label and FlowMonID within the same packet have different scope, identify different flows, and associate different uses.

An important point that will also be discussed in this document is the the uniqueness of the FlowMonID and how to allow disambiguation of the FlowMonID in case of collision.

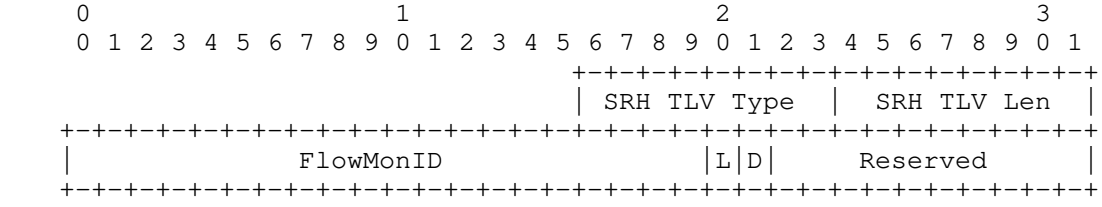
3. Definition of the SRH AltMark TLV

The desired choice is to define a new TLV for the SRH extension headers, carrying the data fields dedicated to the alternate marking method.

This enables the Alternate Marking Method to take advantage of the network programmability capability of SRv6 ([I-D.ietf-spring-srv6-network-programming]). Specifically, the ability for an SRv6 endpoint to determine whether to process or ignore some specific SRH TLVs is based on the SID function. The nodes that are not capable of supporting the Alternate Marking functionality do not have to look or process the SRH AltMark TLV and can simply ignore it. This also enables collection of Alternate Marking data only from the supporting segment endpoints.

3.1. Data Fields Format

The following figure shows the data fields format for enhanced alternate marking TLV. This AltMark data is expected to be encapsulated as SRH TLV.



where:

- SRH TLV Type: 8 bit identifier of the type of Option/TLV that needs to be allocated. Unrecognised Types MUST be ignored on receipt.

- SRH TLV Len: The length of the Data Fields of this TLV in bytes.

- FlowMonID: 20 bits unsigned integer. The FlowMon identifier is described hereinafter.

- L: Loss flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];

- D: Delay flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];

- Reserved: is reserved for future use. These bits MUST be set to zero on transmission and ignored on receipt.

The Flow Monitoring Identification (FlowMonID) is required for some general reasons:

First, it helps to reduce the per node configuration. Otherwise, each node needs to configure an access-control list (ACL) for each of the monitored flows. Moreover, using a flow identifier allows a flexible granularity for the flow definition.

Second, it simplifies the counters handling. Hardware processing of flow tuples (and ACL matching) is challenging and often incurs into performance issues, especially in tunnel interfaces.

Third, it eases the data export encapsulation and correlation for the collectors.

The FlowMon identifier field is to uniquely identify a monitored flow within the measurement domain. The field is set at the source node. The FlowMonID can be uniformly assigned by the central controller or algorithmically generated by the source node. The latter approach cannot guarantee the uniqueness of FlowMonID but it may be preferred for local or private network, where the conflict probability is small due to the large FlowMonID space.

It is important to note that if the 20 bit FlowMonID is set independently and pseudo randomly there is a chance of collision. So, in some cases, FlowMonID could not be sufficient for uniqueness.

This issue is more visible when the FlowMonID is pseudo randomly generated by the source node and there needs to tag it with additional flow information to allow disambiguation. While, in case of a centralized controller, the controller should set FlowMonID by considering these aspects and instruct the nodes properly in order to guarantee its uniqueness.

4. Use of the SRH AltMark TLV

SRv6 leverages the Segment Routing header which consists of a new type of routing header. Like any other use case of IPv6, Hop-by-Hop and Destination Options are useable when SRv6 header is present. Because SRv6 is a routing header, destination options before the routing header are processed by each destination in the route list.

SRH TLV can also be used to encode the AltMark Data Fields for SRv6 and to monitor every node along the SR path. For SRv6, it may be preferred to use the SRH TLV, while for all the other cases with IPv6 data plane the use of the Hop-by-Hop and Destination Option to carry AltMark data fields (as described in [I-D.ietf-6man-ipv6-alt-mark]) is the best choice.

It is to be noted that the SR nodes implementing the Alternate Marking functionality follows the MTU and other considerations

outlined in [I-D.voyer-6man-extension-header-insertion]. Furthermore, in a SRv6 network, the intermediated nodes that are not in the SID list do not consider the SRH, therefore they cannot support and dig into the SRH TLV.

It is possible to summarize the procedure for AltMark data encapsulation in SRv6 SRH:

* Ingress Node: As part of the SRH encapsulation, the ingress node of an SR domain or an SR Policy [I-D.ietf-spring-segment-routing-policy] MAY add the AltMark TLV in the SRH of the data packet, if it supports AltMark functionality and based on local configuration.

* Intermediate SR Node: The intermediate SR node is any node receiving an IPv6 packet where the destination address of that packet is a local SID. If an intermediate SR node is not capable of processing AltMark TLV, it simply ignores it. While, if an intermediate SR node is capable of processing AltMark TLV, it checks if SRH AltMark TLV is present in the packet using procedures defined in [RFC8754] and process it.

* Egress Node: The Egress node is the last node in the segment-list of the SRH. The processing of AltMark TLV at the Egress node is similar to the processing of AltMark TLV at the Intermediate SR Nodes.

5. Alternate Marking Method Operation

[RFC8321], [RFC8889] describe the Alternate Marking Method in general. While [I-D.ietf-6man-ipv6-alt-mark] describe in detail the application and the Operation of the methodology for IPv6.

6. Security Considerations

The security considerations of SRv6 are discussed in [RFC8754] and [I-D.ietf-spring-srv6-network-programming], and the security considerations of Alternate Marking in general and its application to IPv6 are discussed in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark].

Alternate Marking is a feature applied to a "controlled domain", where one or several operators decide on leveraging and configuring Alternate Marking according to their needs. Additionally, operators need to properly secure the Alternate Marking domain to avoid malicious configuration and use, which could include injecting malicious packets into a domain.

7. IANA Considerations

The SRH TLV Type should be assigned in IANA's Segment Routing Header TLVs Registry.

This draft requests to allocate a SRH TLV Type for Alternate Marking TLV data fields under registry name "Segment Routing Header TLVs" requested by [RFC8754].

SRH TLV Type	Description	Reference
TBD	AltMark Data Fields TLV	This document

8. Acknowledgements

TBD

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

9.2. Informative References

[I-D.fioccola-v6ops-ipv6-alt-mark] Fioccola, G., Velde, G., Cociglio, M., and P. Muley, "IPv6 Performance Measurement with Alternate Marking Method", draft-fioccola-v6ops-ipv6-alt-mark-01 (work in progress), June 2018.

[I-D.ietf-6man-ipv6-alt-mark] Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-02 (work in progress), October 2020.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.voyer-6man-extension-header-insertion]
Voyer, D., Filsfils, C., Dukes, D., Matsushima, S., Leddy, J., Li, Z., and J. Guichard, "Deployments With Insertion of IPv6 Segment Routing Headers", draft-voyer-6man-extension-header-insertion-10 (work in progress), November 2020.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8889] Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <https://www.rfc-editor.org/info/rfc8889>.

Authors' Addresses

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

Mauro Cociglio
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: mauro.cociglio@telecomitalia.it

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 13, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
N. Vaghamshi
Reliance
M. Nagarajah
Telstra
R. Foote
Nokia

February 09, 2021

Enhanced Performance and Liveness Monitoring in Segment Routing Networks
draft-gandhi-spring-sr-enhanced-plm-04

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document defines procedures for Enhanced Performance and Liveness Monitoring (PLM) for end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, those reduce the deployment and operational complexities in a network.

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 August 13, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic

through a specific, user-defined paths using a stack of Segments. Built-in Performance Measurement as well as Liveness Monitoring for Connectivity Verification (CV) and Continuity Check (CC) are essential requirements to provide Service Level Agreements (SLAs) in SR networks.

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762]. It eliminates the need for control protocol by using configuration and management model to provision and manage test sessions. The STAMP can be used for Performance Measurement (PM) in SR networks as well as liveness monitoring and connectivity loss detection of SR paths. However, the STAMP requires protocol support on the Session-Reflector to process the STAMP test packets as packets need to be punted from the forwarding fast path (to slow path or control plane) on the Session-Reflector and STAMP reply test packets need to be generated. This limits the scale for number of STAMP test sessions and faster fault detection intervals.

For Liveness Monitoring, Seamless Bidirectional Forwarding Detection (S-BFD) [RFC7880] can be used in SR networks. However, S-BFD requires protocol support on the BFD-Reflector to process the S-BFD packets as packets need to be punted from the forwarding fast path and generate the reply packets thereby limiting the scale for number S-BFD sessions and faster fault detection intervals. In addition, S-BFD protocol is not defined to enable performance measurement in a network.

Enabling multiple protocols, S-BFD for liveness monitoring and STAMP for performance measurement increases the deployment and operational complexities a network. Also, implementing multiple protocols in a hardware significantly increases the development cost.

This document defines procedures for Enhanced Performance and Liveness Monitoring (PLM) for end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, those reduce the deployment and operational complexities in a network. The procedures use the new test packet formats those have the timestamps at the same locations as the base STAMP test packets to leverage the existing hardware support for STAMP.

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. Abbreviations

S-BFD: Seamless Bidirectional Forwarding Detection.

BSID: Binding Segment ID.

ECMP: Equal Cost Multi-Path.

EB: Endpoint Behaviour.

HMAC: Hashed Message Authentication Code.

MBZ: Must be Zero.

MPLS: Multiprotocol Label Switching.

PLM: Performance and Liveness Monitoring.

PM: Performance Measurement.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

SSID: Sender Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

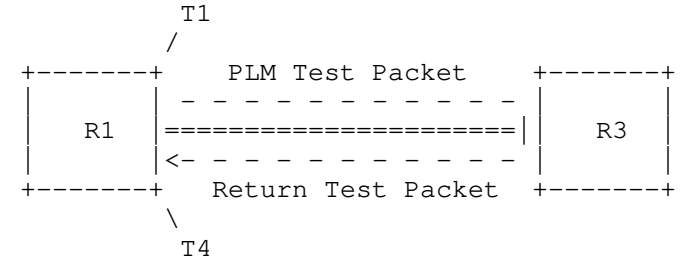
TC: Traffic Class.

TTL: Time To Live.

2.3. Reference Topology

In the reference topology shown below, the Session-Sender R1 initiates a PLM test packet and the Session-Reflector R3 transmits a PLM return test packet. The PLM return test packet is transmitted back to the Session-Sender R1 on the same path or a different path in the reverse direction.

The Session-Sender R1 and Session-Reflector R3 are connected via an SR path [RFC8402]. The SR path may be an SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R3 (called tail-end).



Reference Topology

3. Overview

3.1. Loopback Mode

In loopback mode, the Session-Sender R1 initiates PLM test packets and the Session-Reflector R3 forwards them just like data packets for the regular traffic back to the Session-Sender R1. The PLM test packets are not punted at the Session-Reflector and does not process them and generate PLM return test packets. The Session-Reflector must not drop the loopback PLM test packets, for example, due to a local policy provisioned. No PLM test session is created on the Session-Reflector.

The Source and Destination IP addresses in the PLM test packets are set to the Session-Reflector and the Session-Sender IP addresses, respectively (representing the reverse direction path). The Source and Destination UDP ports in the PLM test packets follow the procedure defined in [RFC8762]. The IPv4 Time To Live (TTL) and IPv6 Hop Limit (HL) are set to 255.

3.2. Loopback Mode Enabled with Network Programming Function

In loopback mode enabled with network programming function, both transmit (T1) and receive (T2) timestamps in data plane are collected by the PLM test packets transmitted in loopback mode as shown in Figure 1. The network programming function optimizes the "operations of punt and generate the PLM test packet" on the Session-Reflector as timestamping is implemented in forwarding fast path in hardware. This helps to achieve higher test session scale and faster failure detection interval.

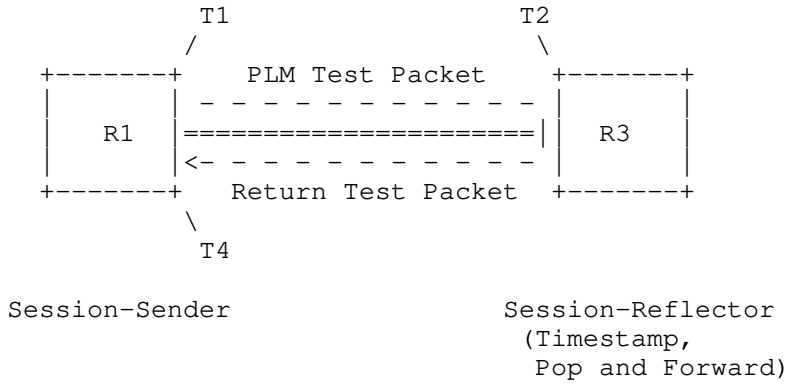


Figure 1: Loopback Mode Enabled with Network Programming Function

The Session-Sender adds transmit timestamp (T1) in the payload of the PLM test packet and clears the receive (T2) timestamp. The Session-Reflector adds the receive timestamp (T2) in the payload of the received PLM test packet in forwarding fast path in hardware without punting the test packet to the slow path (or control-plane). The network programming function enables Session-Reflector to add the receive timestamp (T2) at a specific offset in the payload which is locally provisioned consistently in the network. The payload of the PLM test packet is not modified by the intermediate nodes.

The Session-Reflector only adds the receive timestamp if the source IP address (in case of SR-MPLS) or destination IP address (in case of SRv6) in the PLM test packet matches the local node address to ensure that the PLM test packet reaches the intended Session-Reflector and the receive timestamp is returned by the intended Session-Reflector.

3.3. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 2:

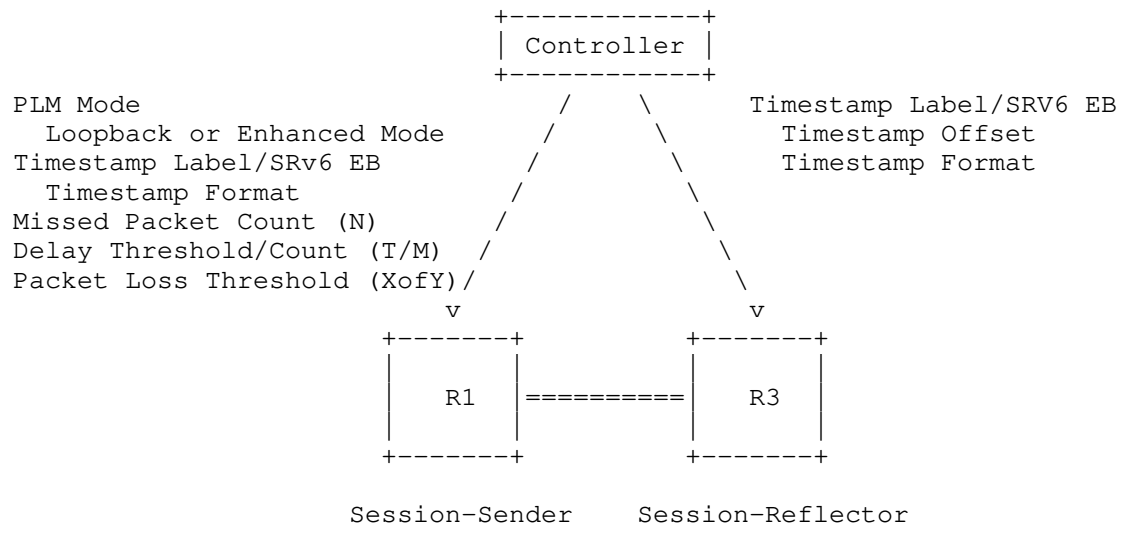


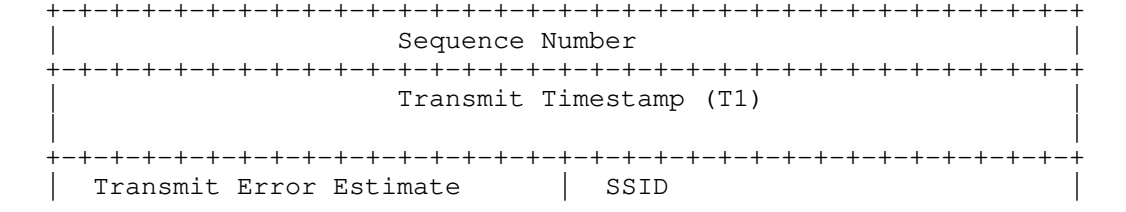
Figure 2: Example Provisioning Model

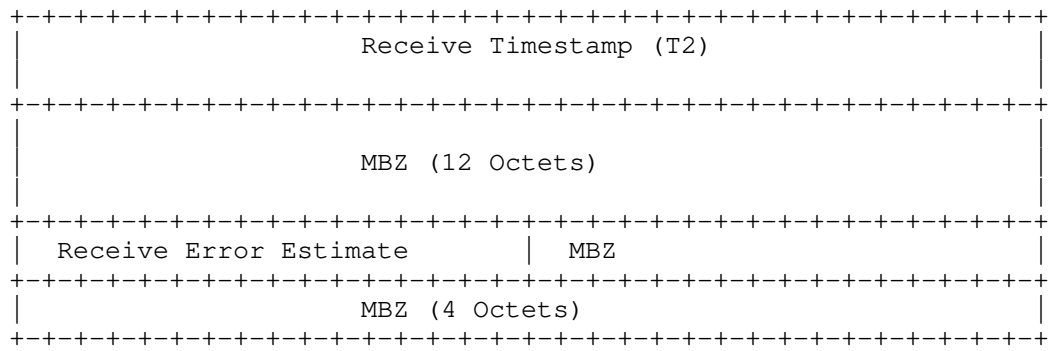
Example of PLM mode is loopback mode. The values for Timestamp Label and SRv6 Endpoint Behaviour may be provisioned as described in Section 6. Example of Timestamp Format is 64-bit PTPv2 [IEEE1588]. Example of Timestamp Offset is 16 and 32 bytes for the PLM test packet formats defined in this document. Example threshold values configured for generating notifications are: Missed Packet Count (N), Delay Exceeded Threshold and Packet Count (T/M) and Packet Loss Threshold (XofY), as described in Section 7.

The mechanisms to provision the Session-Sender and Session-Reflector are outside the scope of this document.

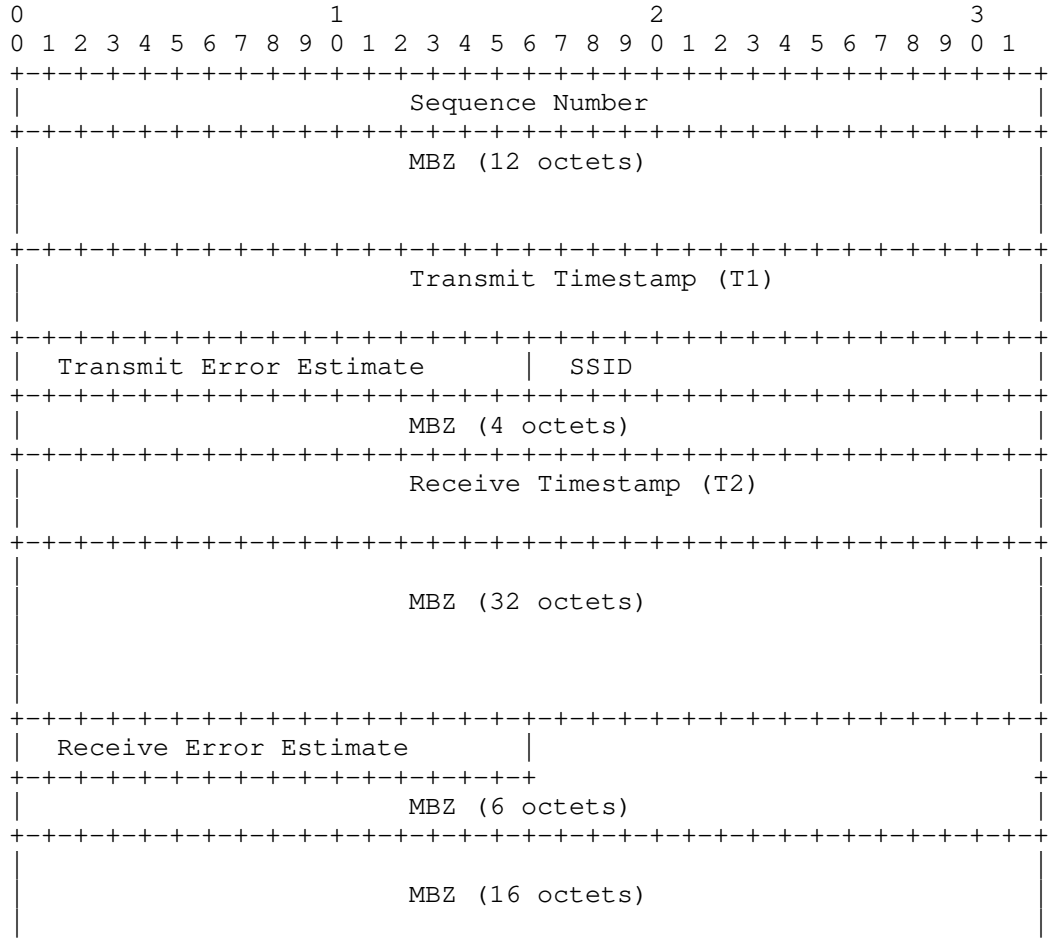
4. PLM Test Packet Formats

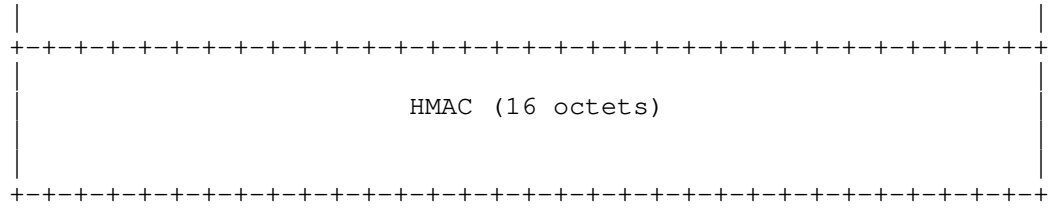
The PLM test packet formats for unauthenticated and authenticated modes are defined in this document as shown in Figure 3 those have the transmit and receive timestamps at the same locations as the base STAMP test packets to leverage the existing hardware support for STAMP.





PLM Test Packet Format in Unauthenticated Mode





PLM Test Packet Format in Authenticated Mode

Figure 3: PLM Test Packet Formats

Sequence Number is the sequence number of the PLM test packet according to its transmit order. It starts with zero and is incremented by one for each subsequent PLM test packet.

SSID (16-bits): PLM Sender Session Identifier. Uses the procedure for SSID defined in [RFC8762].

Transmit Timestamp and Transmit Error Estimate are the Session-Sender's transmit timestamp and error estimate for the PLM test packet, respectively.

Receive Timestamp and Receive Error Estimate are the Session-Reflector's receive timestamp and error estimate, respectively.

The timestamp and error estimate fields follow the definition and formats defined in Section 4.1.2 in [RFC8762]. The timestamp format used by default is 64-bit PTPv2 [IEEE1588].

HMAC: The use of the HMAC field is described in Section 4.4 of [RFC8762].

MBZ: Must be Zero. It MUST be all zeroed on the transmission and MUST be ignored on receipt.

5. PLM Procedure

For performance and liveness monitoring of an end-to-end SR path including SR Policy, PLM test packets in loopback mode are used.

For SR Policy, the PLM test packets are transmitted using the Segment List (SL) of the Candidate-Path [I-D.ietf-spring-segment-routing-policy]. When a Candidate-Path has more than one Segment Lists, multiple PLM test packets are sent, one using each Segment List. The PLM return test packets are received by the Session-Sender via IP/UDP [RFC0768] return path by default. The Segment List of the return SR path can be added in the PLM test

packet header to receive the return test packet on a specific path using the Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path].

5.1. PLM for SR-MPLS Policies

The PLM test packets are transmitted using the MPLS header for each Label Stack of the SR-MPLS Policy Candidate-Path(s) as shown in Figure 4. In case of IP/UDP return path, the MPLS header is removed by the Session-Reflector. The Label Stack can contain a reverse SR-MPLS path to receive the PLM return test packet on a specific path. In this case, the MPLS header will not be removed by the Session-Reflector.

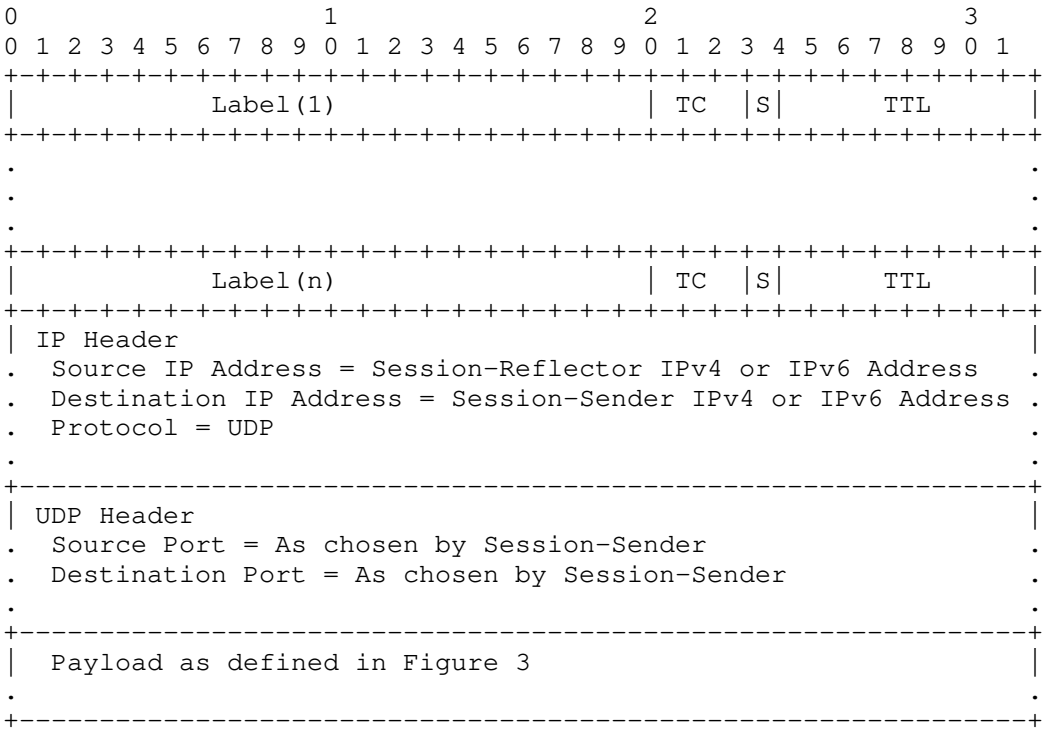


Figure 4: Example PLM Test Packet for SR-MPLS

5.2. PLM for SRv6 Policies

The PLM test packets for SRv6 data plane are transmitted using the Segment Routing Header (SRH) [RFC8754] for each Segment List of the SRv6 Policy Candidate-Path(s) as shown in Figure 5. In case of IP/UDP return path, the SRH is removed by the Session-Reflector. The

Segment List can contain a reverse SRv6 path to receive the PLM return test packet on a specific path. In this case, the SRH will not be removed by the Session-Reflector. When the PLM return test packet contains an SRH at the Session-Sender, the procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the UDP header in the received PLM test packets.

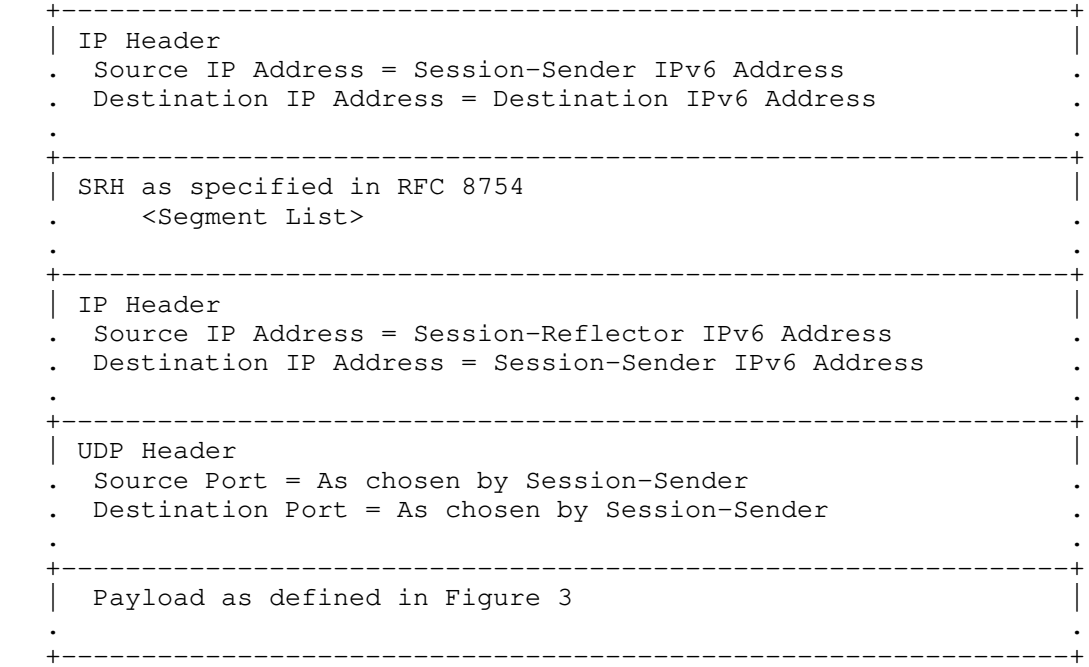


Figure 5: Example PLM Test Packet for SRv6

6. Enhanced PLM Procedure

The enhanced performance and liveness monitoring of an end-to-end SR path including SR Policy is defined using the PLM test packets in loopback mode enabled with network programming function.

6.1. Enhanced PLM with Timestamp Label for SR-MPLS Policies

In this document, two new Timestamp Labels are defined for SR-MPLS data plane to enable network programming function for "timestamp, pop and forward" the received test packet.

In the PLM test packets for SR-MPLS Policies, a Timestamp Label is added in the MPLS header as shown in Figure 6, to collect "Receive

Timestamp" field in the payload of the PLM test packet. The Label Stack for the reverse SR-MPLS path can be added after the Timestamp Label to receive the PLM return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Label, after timestamping the packet at a specific offset, the Session-Reflector pops the Timestamp Label and forwards the packet using the next label or IP header in the packet (just like the data packets for the regular traffic).

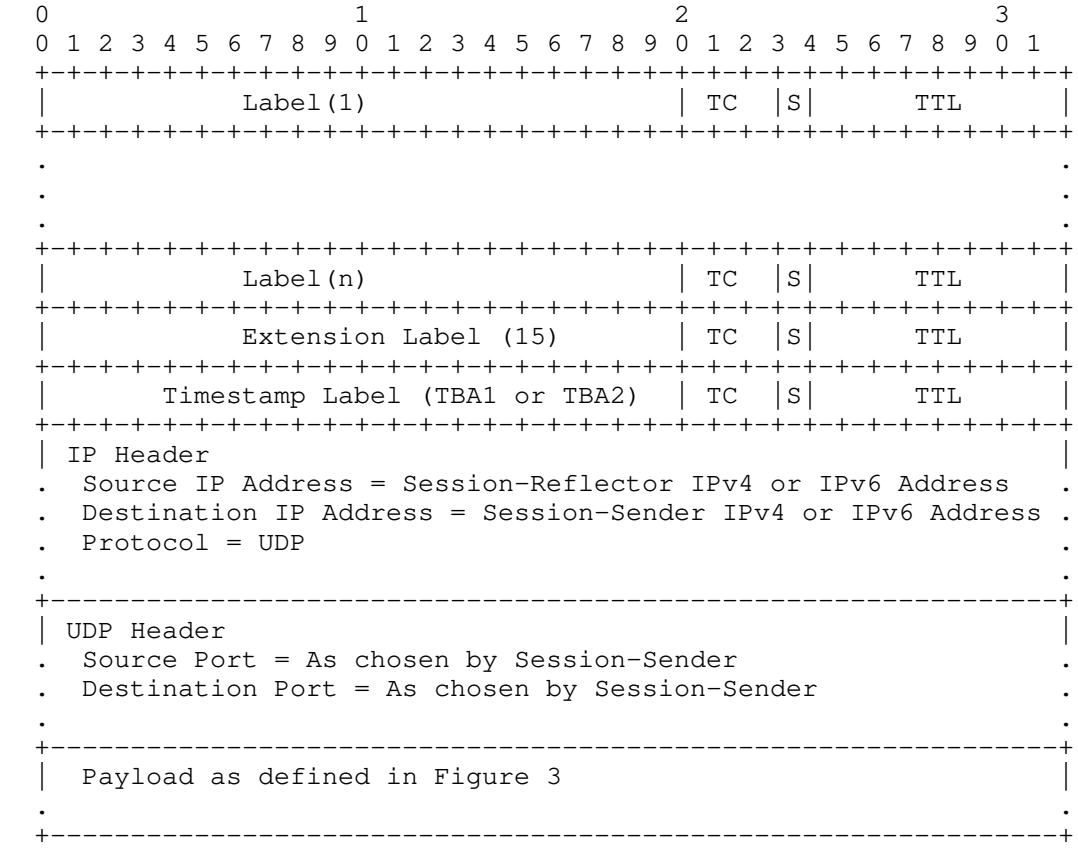


Figure 6: Example PLM Test Packet with Timestamp Label for SR-MPLS

6.1.1. Timestamp Label Allocation

The timestamp Labels for unauthenticated and authenticated modes can be allocated using one of the following methods:

- Labels (values TBA1 and TBA2) assigned by IANA from the "Extended Special-Purpose MPLS Values" [I-D.ietf-mpls-spl-terminology]. For

Label (value TBA1), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the unauthenticated mode, and Label (value TBA2) at byte-offset 32 from the start of the payload for the authenticated mode, both using the timestamp format 64-bit PTPv2.

- Labels allocated by a Controller from the global table of the Session-Reflector. The Controller provisions the labels on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.

- Labels allocated by the Session-Reflector. The signaling and IGP flooding extension for the labels (including timestamp offsets and timestamp formats) are outside the scope of this document.

6.1.2. Node Capability for Timestamp Label

The PLM Session-Sender needs to know if the Session-Reflector can process the Timestamp Label to avoid dropping PLM test packets. The signaling extension for this capability exchange is outside the scope of this document.

6.2. Enhanced PLM with Timestamp Endpoint Function for SRv6 Policies

The [I-D.ietf-spring-srv6-network-programming] defines SRv6 Endpoint Behaviours (EB) for SRv6 nodes. In this document, two new Timestamp Endpoint Behaviours are defined for Segment Routing Header (SRH) [RFC8754] to enable "Timestamp and Forward (TSF)" function for the received test packets.

In the PLM test packets for SRv6 Policies, Timestamp Endpoint Function (End.TSF) is carried with the target Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 7, to collect "Receive Timestamp" field in the payload of the PLM test packet. The Segment List for the reverse path can be added after the target SID to receive the PLM return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Endpoint (End.TSF) for the target SID which is local, after timestamping the packet at a specific offset, the Session-Reflector forwards the packet using the next SID or IP header in the packet (just like the data packets for the regular traffic).

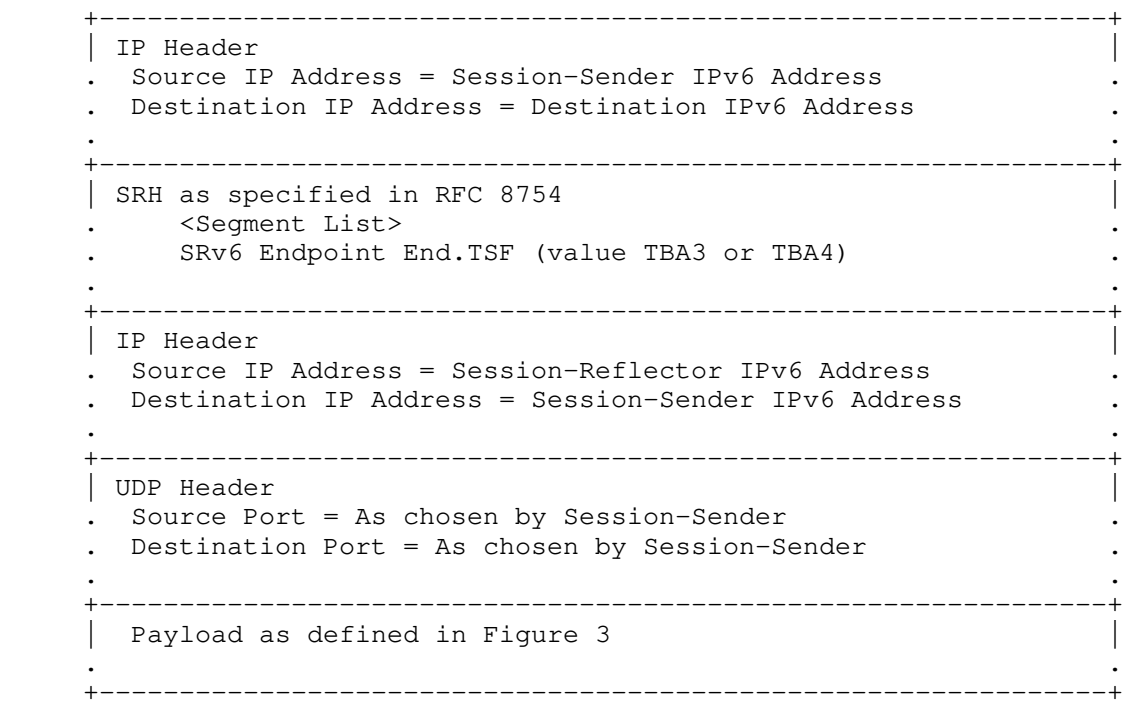


Figure 7: Example PLM Test Packet with Endpoint Function for SRv6

6.2.1. Timestamp Endpoint Function Assignment

The Timestamp Endpoint Functions for "Timestamp and Forward" can be signaled using one of the following methods:

- Timestamp Endpoint Functions (values TBA3 and TBA4) assigned by IANA from the "SRv6 Endpoint Behaviors Registry". For endpoint behaviour (value TBA3), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the unauthenticated mode, and endpoint behaviour (value TBA4) at byte-offset 32 from the start of the payload for the authenticated mode, both using the timestamp format 64-bit PTPv2.

- Timestamp Endpoint Functions assigned by a Controller. The Controller provisions the values on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.

- Timestamp Endpoint Functions assigned by the Session-Reflector. The signaling and IGP flooding extension for the endpoint

functions (including timestamp offsets and timestamp formats) are outside the scope of this document.

6.2.2. Node Capability for Timestamp Endpoint Function

The PLM Session-Sender needs to know if the Session-Reflector can process the Timestamp Endpoint Function to avoid dropping PLM test packets. The signaling extension for this capability exchange is outside the scope of this document.

7. ECMP Handling

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. The PLM test packets need to be sent to traverse different ECMP paths to monitor an end-to-end SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. In IPv4 header of the PLM test packets, sweeping of Destination Address from the 127/8 range can be used to exercise different IPv4 ECMP paths in both loopback modes as long as the forward and the return paths are SR-MPLS paths. In this case, the TTL field in the IPv4 header is set to 1.

The Flow Label field in the outer IPv6 header can also be used for sweeping to exercise different IPv6 ECMP paths.

8. Example PLM Failure Notifications

Liveness or connectivity success for an end-to-end SR path is initially notified as soon as one or more PLM return test packets are received at the Session-Sender.

Liveness or connectivity failure for an end-to-end SR path is notified when consecutive N number of PLM return test packets are not received at the Session-Sender, where N (Missed PLM Packet Count) is a locally provisioned value.

The round-trip packet loss for an end-to-end SR path is calculated using the Sequence Number in the PLM test packets. The packet loss metric is notified when X number of PLM test packets were lost out of last Y number of PLM test packets transmitted by the Session-Sender, where Threshold XofY is locally provisioned value.

Similarly, the delay metrics are notified, as an example, when consecutive M number of PLM test packets have measured delay values exceed user-configured threshold T, where M (Delay Exceeded Packet

Count) and T (Absolute and Percentage Delay Exceeded Threshold) are also locally provisioned values.

In both loopback modes, the timestamps T1 and T4 are used to measure round-trip delay. In loopback mode enabled with network programming function, the timestamps T1 and T2 are used to measure one-way delay.

In both loopback modes, a failure on the reverse direction path can cause the PLM return test packets to not reach the Session-Sender. This is also true in the case where the PLM return test packets were generated by the Session-Reflector e.g. to indicate Session-Sender of a failure on the forward direction path. As such, the test packet based methods have a limitation of false detection due to a reverse direction failure.

9. Security Considerations

The Performance and Liveness Monitoring is intended for deployment in the well-managed private and service provider networks. As such, it assumes that a node involved in a monitoring operation has previously verified the integrity of the path and the identity of the Session-Reflector.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the Session-Sender, of the timestamp fields in received PLM packets. The minimal state associated with these protocols also limits the extent of disruption that can be caused by a corrupt or invalid packet to a single test cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the test packets. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

The security considerations specified in [RFC8762] also apply to the procedures defined in this document.

10. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>). IANA is requested to allocate Timestamp Label value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	Timestamp Label for offset 16 for Unauthenticated Mode	This document
TBA2	Timestamp Label for offset 32 for Authenticated Mode	This document

IANA is requested to allocate, within the "SRv6 Endpoint Behaviors Registry" sub-registry belonging to the top-level "Segment Routing Parameters" registry [I-D.ietf-spring-srv6-network-programming], the following allocation:

Value	Endpoint Behavior	Reference
TBA3	End.TSF (Timestamp and Forward) for offset 16 for Unauthenticated Mode	This document
TBA4	End.TSF (Timestamp and Forward) for offset 32 for Authenticated Mode	This document

11. References

11.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

11.2. Informative References

[IEEE1588]
IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <https://www.rfc-editor.org/info/rfc7880>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-mpls-spl-terminology]
Andersson, L., Kompella, K., and A. Farrel, "Special Purpose Label terminology", draft-ietf-mpls-spl-terminology-06 (work in progress), January 2021.

[I-D.ietf-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-ietf-pce-binding-label-sid-05 (work in progress), October 2020.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-05 (work in progress), January 2021.

Acknowledgments

The authors would like to thank Greg Mirsky, Mach Chen, Kireeti Kompella, and Adrian Farrel for providing the review comments.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Navin Vaghamshi
Reliance

Email: Navin.Vaghamshi@ril.com

Moses Nagarajah
Telstra

Email: Moses.Nagarajah@team.telstra.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: August 14, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
M. Chen
Huawei
B. Janssens
Colt

February 10, 2021

Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks
draft-gandhi-spring-stamp-srpm-05

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document describes procedures for Performance Measurement in SR networks using the mechanisms defined in RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) and its optional extensions defined in RFC 8972 and draft-gandhi-ippm-stamp-srpm. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both links and end-to-end SR paths including 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 August 14, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762]. It eliminates the need for control protocol by using configuration and management model to provision and manage test sessions. [RFC8972] defines optional extensions for STAMP. [I-D.gandhi-ippm-stamp-srpm] defines STAMP extensions for SR networks.

The STAMP supports two modes of STAMP Session-Reflector: Stateless and Stateful as described in Section 4 of [RFC8762]. In Stateless mode, maintenance of each STAMP test session on Session-Reflector is avoided. In SR networks, as the state is in the packet, the signaling of the parameters and creating extra states in the network are undesired. Hence, Stateless mode of Session-Reflector is preferred in SR networks.

This document describes procedures for Performance Measurement in SR networks using the mechanisms defined in STAMP [RFC8762] and its optional extensions defined in [RFC8972] and [I-D.gandhi-ippm-stamp-srpm]. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both links and end-to-end SR paths including SR Policies [RFC8402].

2. Conventions Used in This Document

2.1. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SHA: Secure Hash Algorithm.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

SSID: STAMP Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

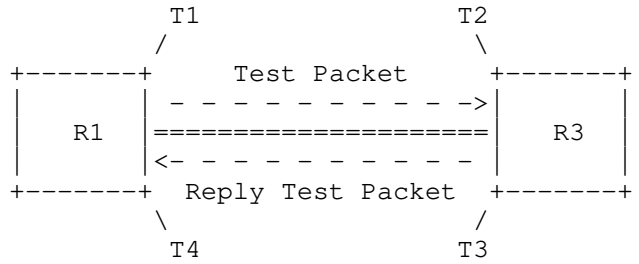
TTL: Time To Live.

2.2. Reference Topology

In the reference topology shown below, the STAMP Session-Sender R1 initiates a STAMP test packet and the STAMP Session-Reflector R3 transmits a reply test packet. The reply test packet is transmitted back to the STAMP Session-Sender R1 on the same path or a different path in the reverse direction.

The nodes R1 and R3 may be connected via a link or there exists an SR path [RFC8402]. The link may be a physical interface, virtual link, or Link Aggregation Group (LAG) [IEEE802.1AX], or LAG member link. The SR path may be an SR Policy

[I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R3 (called tail-end).



Reference Topology

3. Overview

For performance measurement in SR networks, the STAMP test packets defined in [RFC8762] and its optional extensions defined in [RFC8972] and [I-D.gandhi-ippm-stamp-srpm] are used as described in this document. The procedures are used to measure one-way, two-way and round-trip delay as well as packet loss metrics in an SR network.

For performance delay and packet loss measurement, STAMP Session-Sender test packets are transmitted in-band on the same path as the data traffic flow under measurement to measure the delay and packet loss experienced by the data traffic flow. It is also desired that Session-Reflector reply test packets are transmitted in-band on the same path in the reverse direction. This is achieved in SR networks by using the STAMP extensions defined in [I-D.gandhi-ippm-stamp-srpm].

A destination UDP port number is selected as described in [RFC8762]. The same destination UDP port is used for link and end-to-end SR path STAMP test sessions.

3.1. Example STAMP Reference Model

An example of a STAMP reference model and typical measurement parameters including the destination UDP port for STAMP test session is shown in the following Figure 1:

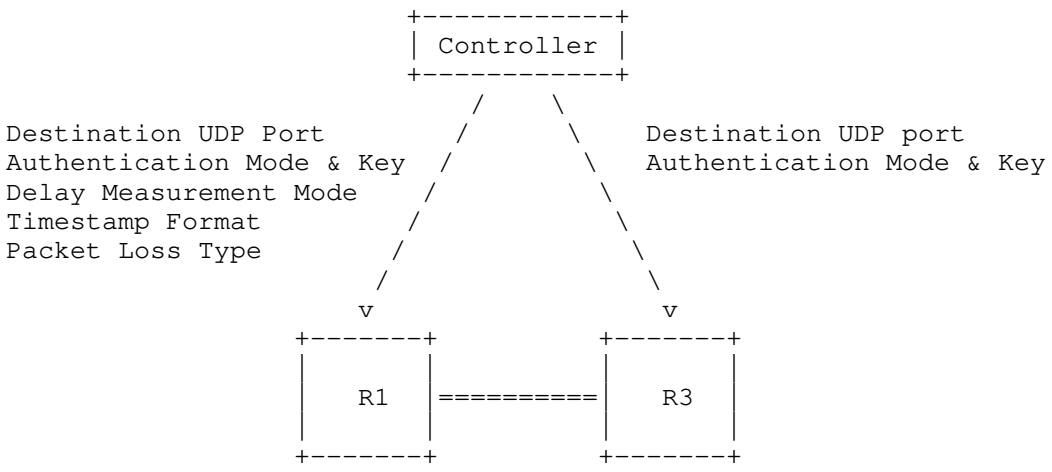


Figure 1: Example STAMP Reference Model

Example of the Timestamp Format is PTPv2 [IEEE1588] and NTP. Example of Delay Measurement Mode is one-way, two-way and round-trip mode as described in this document. Example of Packet Loss Type is round-trip packet loss [RFC8762].

When using the authenticated mode for delay measurement, the matching Authentication Type (e.g. HMAC-SHA-256) and Key are user-configured on STAMP Session-Sender and STAMP Session-Reflector [RFC8762].

The STAMP Session-Reflector R3 uses the timestamp format from the received STAMP test packet. In addition, the STAMP Session-Reflector R3 uses the parameters of the return path for the reply test packet from the received STAMP test packet, as described in this document.

Note that the controller in the reference model is not intended for signaling the SR parameters for STAMP test sessions between the STAMP Session-Sender and STAMP Session-Reflector. In addition, maintenance of each STAMP test session on Session-Reflector and creating extra state are avoided in an SR network.

The YANG data model defined in [I-D.ietf-ippm-stamp-yang] can be used to provision the STAMP Session-Sender and STAMP Session-Reflector.

4. Delay Measurement for Links and SR Paths

4.1. Session-Sender Test Packet

The content of an example STAMP Session-Sender test packet using an UDP header [RFC0768] is shown in Figure 2. The payload contains the STAMP Session-Sender test packet defined in [RFC8762].

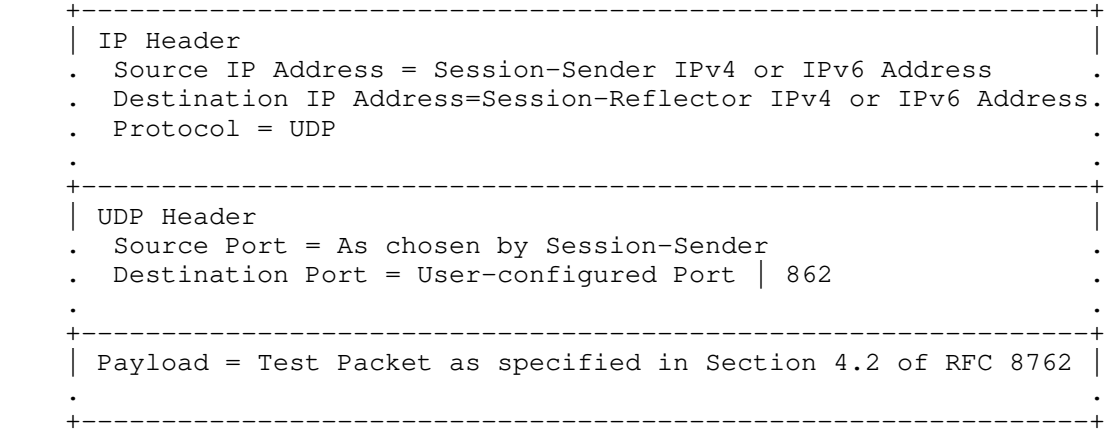


Figure 2: Example Session-Sender Test Packet

4.1.1. Session-Sender Test Packet for Links

The STAMP Session-Sender test packet as shown in Figure 2 is transmitted over the link for delay measurement. The local and remote IP addresses of the link are used as Source and Destination Addresses.

4.1.2. Session-Sender Test Packet for SR Paths

The delay measurement for end-to-end SR path in SR network is applicable to both end-to-end SR-MPLS and SRv6 paths including SR Policies.

The STAMP Session-Sender IPv4 or IPv6 address is used as the Source Address. The SR Policy endpoint IPv4 or IPv6 address is used as the Destination Address.

In the case of Color-Only Destination Steering, with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the loopback address from the range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 is used as the Destination Address, respectively.

4.1.2.1. Session-Sender Test Packet for SR-MPLS Policies

An SR-MPLS Policy may contain a number of Segment Lists. A STAMP Session-Sender test packet is transmitted for each Segment List of the SR-MPLS Policy. The content of an example STAMP Session-Sender test packet for an end-to-end SR-MPLS Policy is shown in Figure 3.

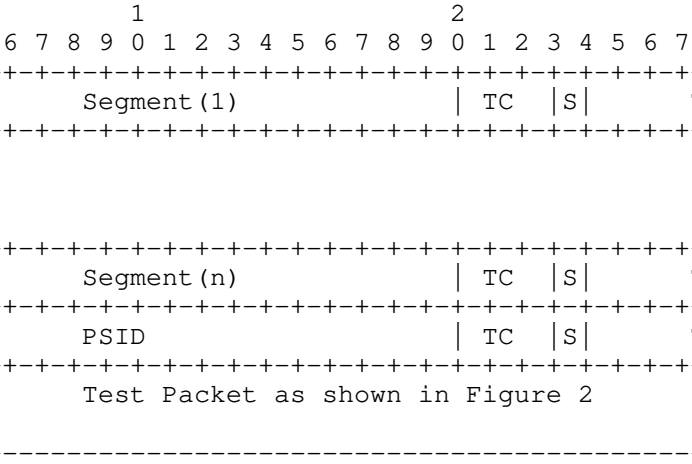


Figure 3: Example Session-Sender Test Packet for SR-MPLS Policy

The Segment List (SL) can be empty in case of a single-hop SR-MPLS Policy with Implicit NULL label.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of an SR-MPLS Policy can be carried in the MPLS header as shown in Figure 3, and can be used for direct measurement as described in Section 7.

4.1.2.2. Session-Sender Test Packet for SRv6 Policies

An SRv6 Policy may contain a number of Segment Lists. A STAMP Session-Sender test packet is transmitted for each Segment List of the SRv6 Policy. An SRv6 Policy can contain an SRv6 Segment Routing Header (SRH) carrying a Segment List as described in [RFC8754]. The content of an example STAMP Session-Sender test packet for an end-to-end SRv6 Policy is shown in Figure 4.

The SRv6 network programming is described in [I-D.ietf-spring-srv6-network-programming]. The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is used to process the

IPv6/UDP header in the received test packets on the Session-Reflector.

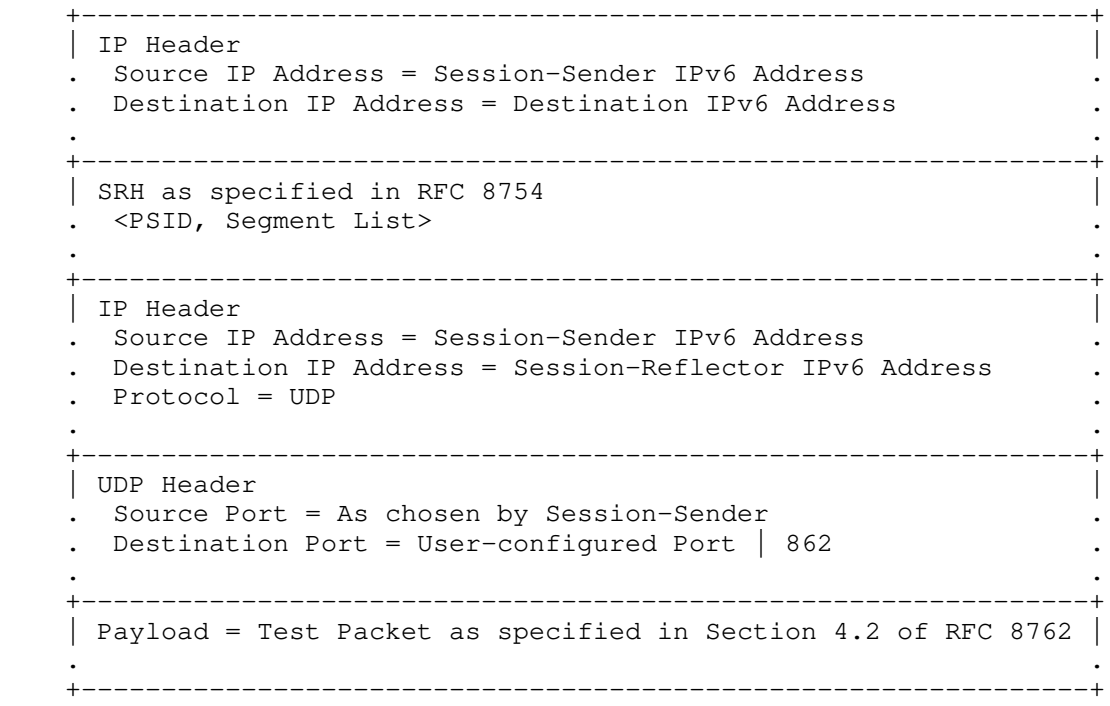


Figure 4: Example Session-Sender Test Packet for SRv6 Policy

The Segment List (SL) may be empty and no SRH may be carried.

The Path Segment Identifier (PSID) [I-D.ietf-spring-srv6-path-segment] of the SRV6 Policy can be carried in the SRH as shown in Figure 4 and can be used for direct measurement as described in Section 7.

4.2. Session-Reflector Test Packet

The STAMP Session-Reflector reply test packet is transmitted using the IP/UDP information from the received test packet. The content of an example STAMP Session-Reflector reply test packet is shown in Figure 5.

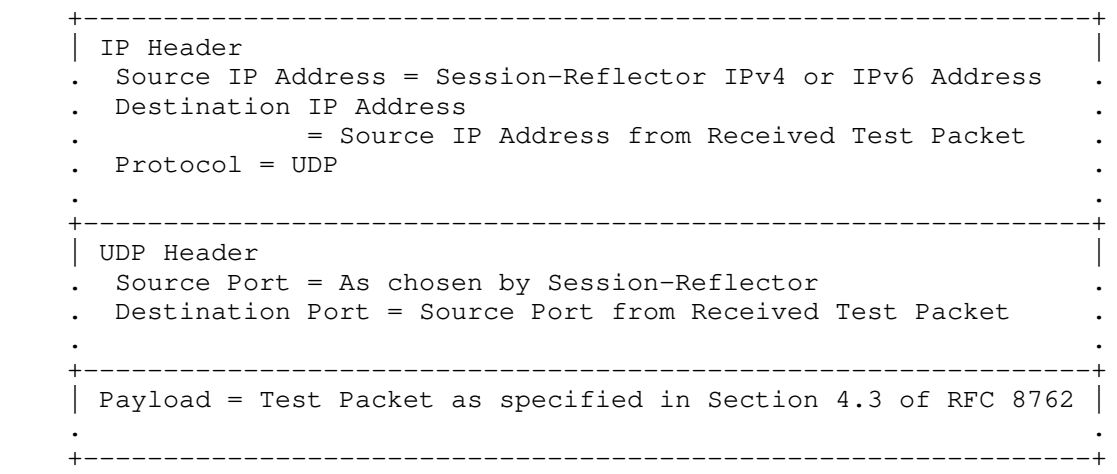


Figure 5: Example Session-Reflector Test Packet

4.2.1. One-way Delay Measurement Mode

In one-way delay measurement mode, a reply test packet as shown in Figure 5 is transmitted by the STAMP Session-Reflector, for both links and SR Policies. The reply test packet may be transmitted on the same path or a different path in the reverse direction.

The STAMP Session-Sender address may not be reachable via IP route from the STAMP Session-Reflector. The STAMP Session-Sender in this case can send its reachability path information to the STAMP Session-Reflector using the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

In this mode, as per Reference Topology, all timestamps T1, T2, T3, and T4 are collected by the test packets. However, only timestamps T1 and T2 are used to measure one-way delay as (T2 - T1).

4.2.2. Two-way Delay Measurement Mode

In two-way delay measurement mode, a reply test packet as shown in Figure 5 is transmitted by the STAMP Session-Reflector in-band on the same path in the reverse direction, e.g. on the reverse direction link or associated reverse SR path [I-D.ietf-pce-sr-bidir-path].

For two-way delay measurement mode for links, the STAMP Session-Reflector needs to transmit the reply test packet in-band on the same link where the test packet is received. The STAMP Session-Sender can request in the test packet to the STAMP Session-Reflector to transmit the reply test packet back on the same link using the Control Code

Sub-TLV in the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

For two-way delay measurement mode for end-to-end SR paths, the STAMP Session-Reflector needs to transmit the reply test packet in-band on a specific reverse path. The STAMP Session-Sender can request in the test packet to the STAMP Session-Reflector to transmit the reply test packet back on a given reverse path using a Segment List sub-TLV in the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

In this mode, as per Reference Topology, all timestamps T1, T2, T3, and T4 are collected by the test packets. All four timestamps are used to measure two-way delay as ((T4 - T1) - (T3 - T2)).

4.2.2.1. Session-Reflector Test Packet for SR-MPLS Policies

The content of an example STAMP Session-Reflector reply test packet transmitted in-band on the same path as the data traffic flow under measurement for two-way delay measurement of an end-to-end SR-MPLS Policy is shown in Figure 6.

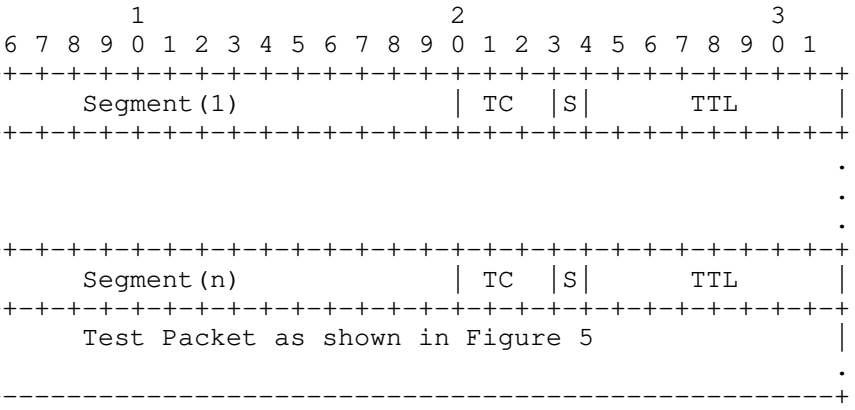


Figure 6: Example Session-Reflector Test Packet for SR-MPLS Policy

4.2.2.2. Session-Reflector Test Packet for SRv6 Policies

The content of an example STAMP Session-Reflector reply test packet transmitted in-band on the same path as the data traffic flow under measurement for two-way delay measurement of an end-to-end SRv6 Policy with SRH is shown in Figure 7.

The procedure defined for upper-layer header processing for SRv6 SIDs in [I-D.ietf-spring-srv6-network-programming] is also used to process

the IPv6/UDP header in the received reply test packets on the Session-Sender.

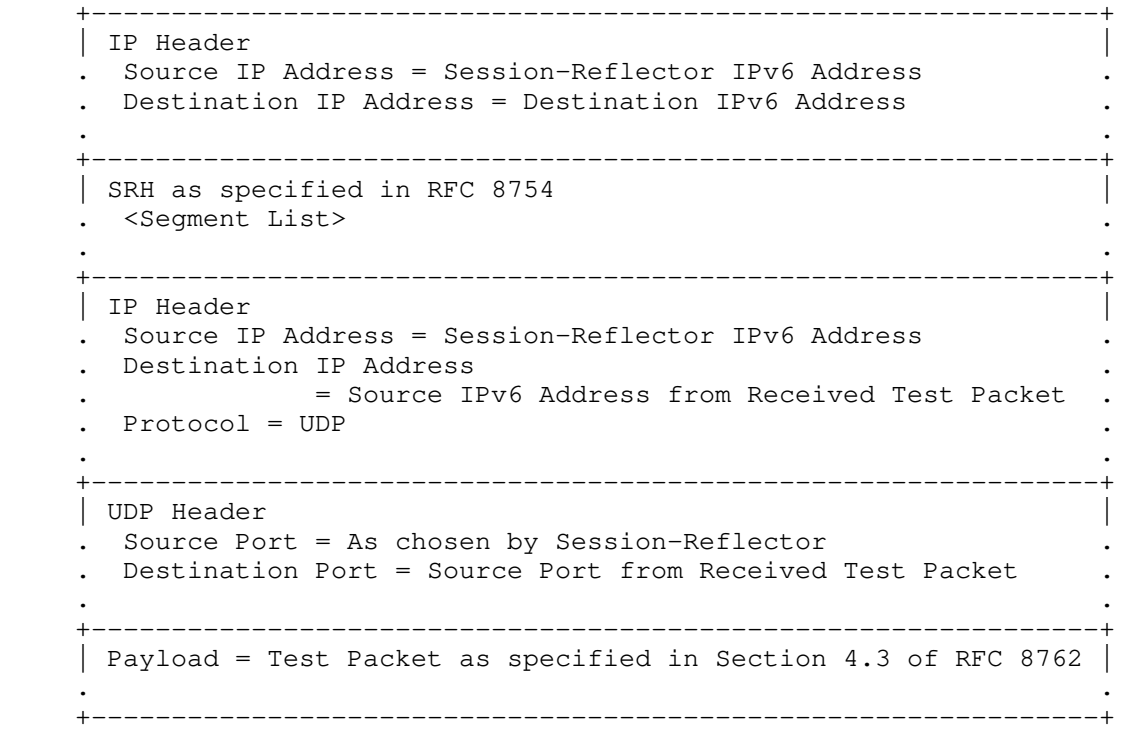


Figure 7: Example Session-Reflector Test Packet for SRv6 Policy

4.2.3. Round-trip Delay Measurement Mode

The STAMP Session-Sender test packets are sent in loopback mode to measure round-trip delay of a bidirectional path. The IP header of the STAMP Session-Sender test packet contains the Destination Address equals to the STAMP Session-Sender address and the Source Address equals to the STAMP Session-Reflector address. Optionally, the STAMP Session-Sender test packet can carry the return path information (e.g. return path label stack for SR-MPLS) as part of the SR header. This way, the received Session-Sender test packets are not punted out of the fast path in forwarding (to slow path or control-plane) at the STAMP Session-Reflector. Also, the Session-Reflector does not process them and generate reply test packets.

As the reply test packet is not generated by the STAMP Session-Reflector, the STAMP Session-Sender ignores the 'Session-Sender

Sequence Number', 'Session-Sender Timestamp', 'Session-Sender Error Estimate', and 'Session-Sender TTL' in the received test packet.

In this mode, as per Reference Topology, the timestamps T1 and T4 are collected by the test packets. Both these timestamps are used to measure round-trip delay as (T4 - T1).

4.3. Delay Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy]).

The procedures for performance measurement described in this document for P2P SR Policies are used for the P2MP SR Policies as listed below.

- The STAMP Session-Sender root node transmits test packets using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 8. The STAMP Session-Sender test packets may contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].

- The Destination Address is set to the loopback address from the range 127/8 for IPv4, or the loopback address ::1/128 for IPv6.

- Each STAMP Session-Reflector leaf node transmits its node address in the Source Address of the reply test packets shown in Figure 5. This allows the STAMP Session-Sender root node to identify the STAMP Session-Reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the delay for each P2MP leaf node individually.

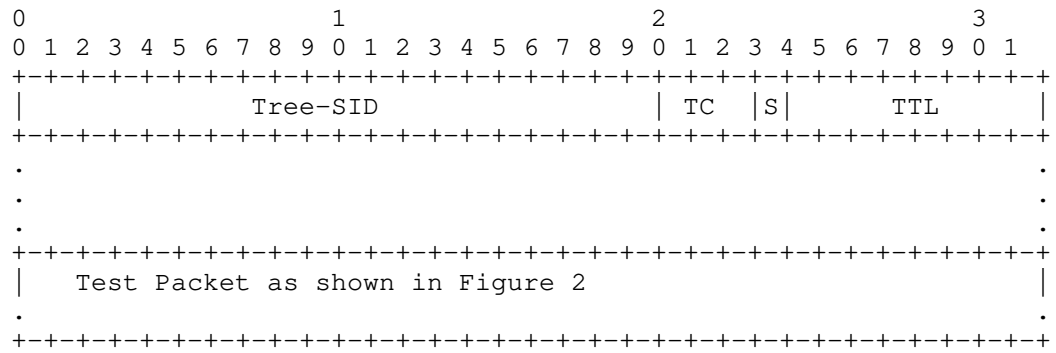


Figure 8: Example Session-Sender Test Packet with Tree-SID for SR-MPLS Policy

The round-trip delay measurement for a P2MP SR-MPLS Policy can use the Node SID of the Session-Sender in the MPLS header of the Session-Sender test packet.

4.4. Additional STAMP Test Packet Processing Rules

The processing rules described in this section are applicable to the STAMP test packets for links and end-to-end SR paths including SR Policies.

4.4.1. TTL

The TTL field in the IPv4 and MPLS headers of the STAMP Session-Sender and STAMP Session-Reflector reply test packets is set to 255, except in the following cases.

When using the Destination IPv4 Address from the range 127/8, the TTL field in the IPv4 header is set to 1.

For link delay, the TTL field in the STAMP test packet is set to 1 in one-way and two-way delay measurement modes.

4.4.2. IPv6 Hop Limit

The Hop Limit field in the IPv6 and SRH headers of the STAMP Session-Sender and STAMP Session-Reflector reply test packets is set to 255, except in the following cases.

When using the Destination IPv6 Address of loopback address ::1/128, the Hop Limit field in the IPv6 header is set to 1.

For link delay, the Hop Limit field in the STAMP test packet is set to 1 in one-way and two-way delay measurement modes.

4.4.3. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the STAMP test packets for links and end-to-end SR paths.

5. Packet Loss Measurement for Links and SR Paths

The procedure described in Section 4 for delay measurement using STAMP test packets can be used to detect (test) packet loss for links and end-to-end SR paths. The Sequence Number field in the STAMP test packet is used as described in Section 4 "Theory of Operation" of [RFC8762], to detect forward, reverse and round-trip packet loss.

6. Direct Measurement for Links and SR Paths

The STAMP "Direct Measurement" TLV (Type 5) defined in [RFC8972] can be used in SR networks. The STAMP test packets with this TLV are transmitted using the procedures described in Section 4 to collect the transmit and receive counters of the data flow for the links and end-to-end SR paths. Note that in this case, the STAMP test packets may follow the same or a different path than the data flow under direct measurement.

The PSID carried in the received data packet for the traffic flow under measurement can be used to measure receive data packets for end-to-end SR path on the STAMP Session-Reflector. The PSID in the received Session-Sender test packet header can be used to associate the receive traffic counter on the Session-Reflector for the end-to-end SR path.

7. Session Status for Links and SR Paths

The STAMP test session status allows to know if the performance measurement is active on the links and end-to-end SR paths. The STAMP test session status initially is declared succeeded when one or more reply test packets are received at the STAMP Session-Sender. The STAMP test session status is declared failed when consecutive N number of reply test packets are not received at the STAMP Session-Sender, where N is locally provisioned value.

8. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP

paths via transit nodes part of that Anycast group. The test packets need to be transmitted to traverse different ECMP paths to measure delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the delay measurement.

In IPv4 header of the STAMP Session-Sender test packets, sweeping of Destination Address from the range 127/8 can be used to exercise particular ECMP paths. Note that in the loopback mode for round-trip delay measurement, both the forward and the return paths must be SR-MPLS paths in this case.

As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping to exercise different IPv6 ECMP paths.

The "Destination Node Address" TLV [I-D.gandhi-ippm-stamp-srpm] can be carried in the STAMP Session-Sender test packet to identify the intended destination node, for example, when using IPv4 Destination Address from the range 127/8. The STAMP Session-Reflector must not transmit reply test packet if it is not the intended destination node in the "Destination Node Address" TLV [I-D.gandhi-ippm-stamp-srpm].

9. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far-end STAMP Session-Reflector.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the STAMP Session-Sender, of the counter or timestamp fields in received measurement reply test packets. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid packet to a single test cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the test packets. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, test packets for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

The security considerations specified in [RFC8762] and [RFC8972] also apply to the procedures described in this document.

10. IANA Considerations

This document does not require any IANA action.

11. References

11.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

[RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-Way Active Measurement Protocol Optional Extensions", RFC 8972, DOI 10.17487/RFC8972, January 2021, <https://www.rfc-editor.org/info/rfc8972>.

[I-D.gandhi-ippm-stamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Simple TWAMP (STAMP) Extensions for Segment Routing Networks", draft-gandhi-ippm-stamp-srpm-02 (work in progress), February 2021.

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

11.2. Informative References

[IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

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

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-02 (work in progress), October 2020.

[I-D.ietf-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-01 (work in progress), October 2020.

[I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-path-segment]
Li, C., Cheng, W., Chen, M., Dhody, D., and R. Gandhi, "Path Segment for SRv6 (Segment Routing in IPv6)", draft-ietf-spring-srv6-path-segment-00 (work in progress), November 2020.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-05 (work in progress), January 2021.

[I-D.ietf-ippm-stamp-yang]
Mirsky, G., Min, X., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-06 (work in progress), October 2020.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in segment routing. The authors would also like to thank Greg Mirsky, Gyan Mishra, Xie Jingrong, and Mike Koldychev for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu have helped improve the mechanisms described in this document.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: October 31, 2021

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
D. Voyer
Bell Canada
M. Chen
Huawei
B. Janssens
Colt
R. Foote
Nokia

April 29, 2021

Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks
draft-gandhi-spring-stamp-srpm-06

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document describes procedures for Performance Measurement in SR networks using the mechanisms defined in RFC 8762 (Simple Two-Way Active Measurement Protocol (STAMP)) and its optional extensions defined in RFC 8972 and further augmented in draft-gandhi-ippm-stamp-srpm. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both links and end-to-end SR paths including 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 October 31, 2021.

Table of Contents

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR takes advantage of the Equal-Cost Multipaths (ECMPs) between source and transit nodes, between transit nodes and between transit and destination nodes. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. Built-in SR Performance Measurement (PM) is one of the essential requirements to provide Service Level Agreements (SLAs).

The Simple Two-way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762] without the use of a control channel to pre-signal session parameters. [RFC8972] defines optional extensions for STAMP. [I-D.gandhi-ippm-stamp-srpm] augments that framework to define STAMP extensions for SR networks.

This document describes procedures for Performance Measurement in SR networks using the mechanisms defined in STAMP [RFC8762] and its optional extensions defined in [RFC8972] and further augmented in [I-D.gandhi-ippm-stamp-srpm]. The procedure described is applicable to SR-MPLS and SRv6 data planes and is used for both links and end-to-end SR paths including SR Policies [RFC8402].

2. Conventions Used in This Document

2.1. Abbreviations

BSID: Binding Segment ID.

DM: Delay Measurement.

ECMP: Equal Cost Multi-Path.

HMAC: Hashed Message Authentication Code.

LM: Loss Measurement.

MPLS: Multiprotocol Label Switching.

NTP: Network Time Protocol.

OWAMP: One-Way Active Measurement Protocol.

PM: Performance Measurement.

PSID: Path Segment Identifier.

PTP: Precision Time Protocol.

SHA: Secure Hash Algorithm.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

SSID: STAMP Session Identifier.

STAMP: Simple Two-way Active Measurement Protocol.

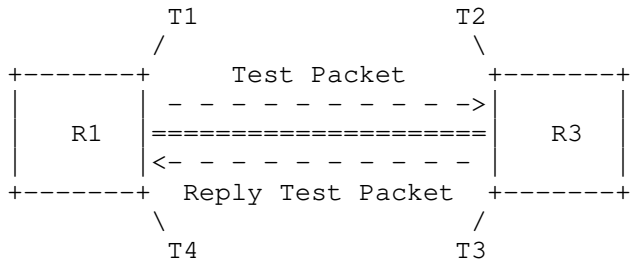
TC: Traffic Class.

TTL: Time To Live.

2.2. Reference Topology

In the Reference Topology shown below, the STAMP Session-Sender R1 initiates a STAMP test packet and the STAMP Session-Reflector R3 transmits a reply test packet. The reply test packet may be transmitted to the STAMP Session-Sender R1 on the same path (same set of links and nodes) or a different path in the reverse direction from the path taken towards the Session-Reflector.

The nodes R1 and R3 may be connected via a link or an SR path [RFC8402]. The link may be a physical interface, virtual link, or Link Aggregation Group (LAG) [IEEE802.1AX], or LAG member link. The SR path may be an SR Policy [I-D.ietf-spring-segment-routing-policy] on node R1 (called head-end) with destination to node R3 (called tail-end).



STAMP Session-Sender STAMP Session-Reflector

Reference Topology

3. Overview

For performance measurement in SR networks, the STAMP Session-Sender and Session-Reflector test packets defined in [RFC8762] are used. They are used in one-way, two-way (i.e. round-trip) and loopback measurement modes. Note that one-way and round-trip are referred to in [RFC8762] and are further described in this document because of the introduction of loopback measurement mode in SR networks. The procedures defined in this document are also used to infer packet loss in SR networks.

The STAMP test packets are transmitted on the same path as the data traffic flow under measurement to measure the delay and packet loss experienced by the data traffic flow.

Typically, the STAMP test packets are transmitted along an IP path between a Session-Sender and a Session-Reflector to measure delay and packet loss along that IP path. Matching the forward and reverse direction paths for STAMP test packets, even for directly connected nodes is not guaranteed.

It may be desired in SR networks that the same path (same set of links and nodes) between the Session-Sender and Session-Reflector be used for the STAMP test packets in both directions. This is achieved by using the optional STAMP extensions for SR-MPLS and SRv6 networks specified in [I-D.gandhi-ippm-stamp-srpm]. The STAMP Session-Reflector uses the return path parameters for the reply test packet from the received STAMP test packet, as described in [I-D.gandhi-ippm-stamp-srpm]. This way signaling and maintaining dynamic SR network state for the STAMP sessions on the Session-Reflector are avoided.

The optional STAMP extensions defined in [RFC8972] are used for direct measurement packet loss in SR networks.

3.1. Example STAMP Reference Model

An example of a STAMP reference model with some of the typical measurement parameters including the Reflector UDP port for STAMP test session is shown in the following Figure 1:

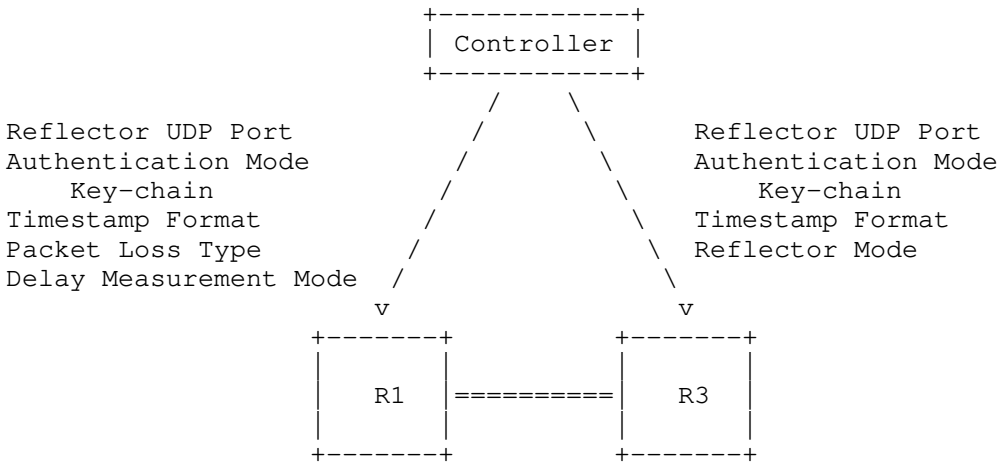


Figure 1: Example STAMP Reference Model

A reflector UDP port number is selected as described in [RFC8762]. The same reflector UDP port can be used for STAMP test sessions for link and end-to-end SR paths. In this case, the reflector UDP port does not distinguish between link or end-to-end SR path measurements.

Example of the Timestamp Format is Precision Time Protocol 64-bit truncated (PTPv2) [IEEE1588] and Network Time Protocol (NTP). By default, the Session-Reflector replies in kind to the timestamp format received in the received Session-Sender test packet, as indicated by the "Z" field in the Error Estimate field as described in [RFC8762].

The Session-Reflector mode can be Stateful or Stateless as defined in [RFC8762].

Example of Delay Measurement Mode is one-way, two-way (i.e. round-trip) and loopback mode as described in this document.

Example of Packet Loss Type can be round-trip, near-end (forward) and far-end (backward) packet loss as defined in [RFC8762].

When using the authenticated mode for the STAMP test sessions, the matching Authentication Type (e.g. HMAC-SHA-256) and Key-chain are user-configured on STAMP Session-Sender and STAMP Session-Reflector [RFC8762].

The controller shown in the example reference model is not intended for the dynamic signaling of the SR parameters for STAMP test sessions between the STAMP Session-Sender and STAMP Session-Reflector.

Note that the YANG data model defined in [I-D.ietf-ippm-stamp-yang] can be used to provision the STAMP Session-Sender and STAMP Session-Reflector.

4. Delay Measurement for Links and SR Paths

4.1. Session-Sender Test Packet

The content of an example STAMP Session-Sender test packet using an UDP header [RFC0768] is shown in Figure 2. The payload contains the STAMP Session-Sender test packet defined in [RFC8762].

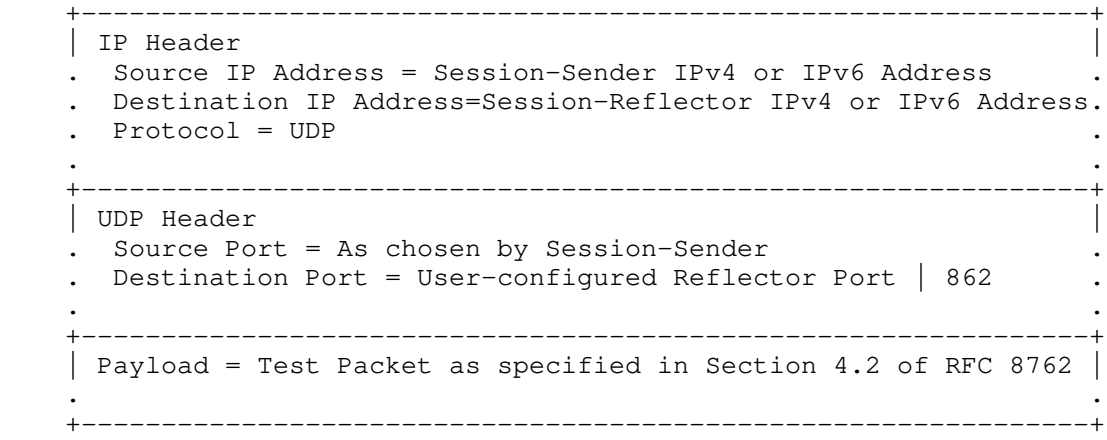


Figure 2: Example Session-Sender Test Packet

4.1.1. Session-Sender Test Packet for Links

The STAMP Session-Sender test packet as shown in Figure 2 is transmitted over the link under delay measurement. The local and remote IP addresses of the link are used as Source and Destination Addresses, respectively. For IPv6 links, the link local addresses [RFC7404] can be used in the IPv6 header. The Session-Sender may use the local Address Resolution Protocol (ARP) table, Neighbor

Solicitation or other bootstrap method to find the IP address for the links and refresh. An IPv4 address from the range 127/8 or IPv6 loopback address ::1/128 [RFC4291] must not be used to IP route test packets in a network.

4.1.2. Session-Sender Test Packet for SR Paths

The delay measurement for end-to-end SR path in an SR network is applicable to both end-to-end SR-MPLS and SRv6 paths including SR Policies.

The STAMP Session-Sender IPv4 or IPv6 address is used as the Source Address. The SR Policy endpoint IPv4 or IPv6 address is used as the Destination Address.

In the case of Color-Only Destination Steering, with IPv4 endpoint of 0.0.0.0 or IPv6 endpoint of ::0 [I-D.ietf-spring-segment-routing-policy], the loopback address from the range 127/8 for IPv4, or the loopback address ::1/128 for IPv6 [RFC4291] is used as the Session-Reflector Address, respectively.

4.1.2.1. Session-Sender Test Packet for SR-MPLS Policies

An SR-MPLS Policy may contain a number of Segment Lists (SLs). A STAMP Session-Sender test packet is transmitted for each Segment List of the SR-MPLS Policy. The content of an example STAMP Session-Sender test packet for an end-to-end SR-MPLS Policy is shown in Figure 3.

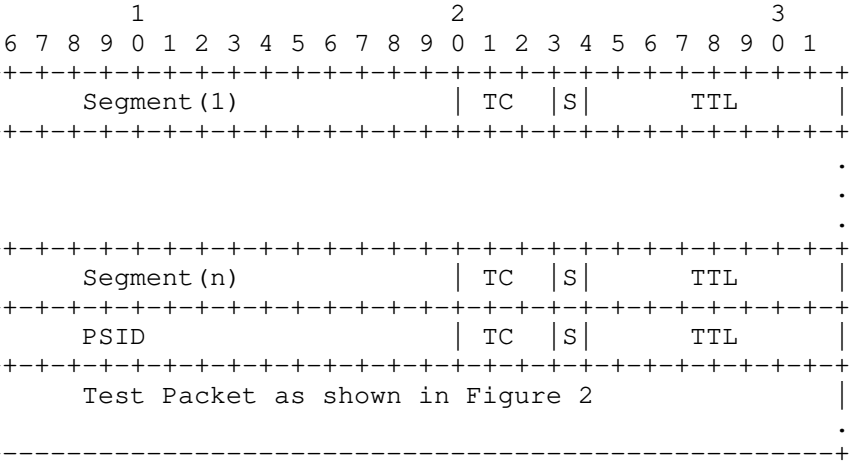


Figure 3: Example Session-Sender Test Packet for SR-MPLS Policy

The Segment List can be empty in case of a single-hop SR-MPLS Policy with Implicit NULL label.

The Path Segment Identifier (PSID) [I-D.ietf-spring-mpls-path-segment] of an SR-MPLS Policy can be carried in the MPLS header as shown in Figure 3, and can be used for direct measurement as described in Section 6, titled "Direct Measurement for Links and SR Paths".

4.1.2.2. Session-Sender Test Packet for SRv6 Policies

An SRv6 Policy may contain a number of Segment Lists. A STAMP Session-Sender test packet is transmitted for each Segment List of the SRv6 Policy. An SRv6 Policy can contain an SRv6 Segment Routing Header (SRH) carrying a Segment List as described in [RFC8754]. The content of an example STAMP Session-Sender test packet for an end-to-end SRv6 Policy is shown in Figure 4.

The SRv6 network programming is described in [RFC8986]. The procedure defined for Upper-Layer Header processing for SRv6 End SIDs in Section 4.1.1 in [RFC8986] is used to process the IPv6/UDP header in the received test packets on the Session-Reflector.

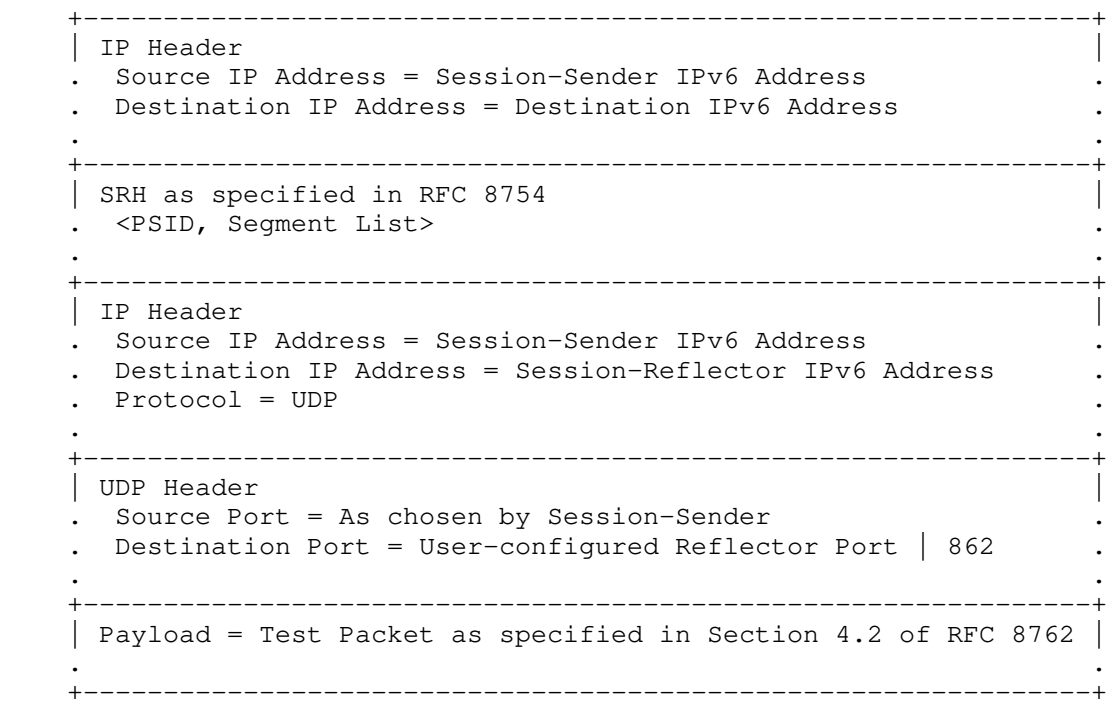


Figure 4: Example Session-Sender Test Packet for SRv6 Policy

The Segment List (SL) may be empty and no SRH may be carried.

The Path Segment Identifier (PSID) [I-D.ietf-spring-srv6-path-segment] of the SRV6 Policy can be carried in the SRH as shown in Figure 4 and can be used for direct measurement as described in Section 6, titled "Direct Measurement for Links and SR Paths".

4.2. Session-Reflector Test Packet

The STAMP Session-Reflector reply test packet uses the IP/UDP information from the received test packet as shown in Figure 5.

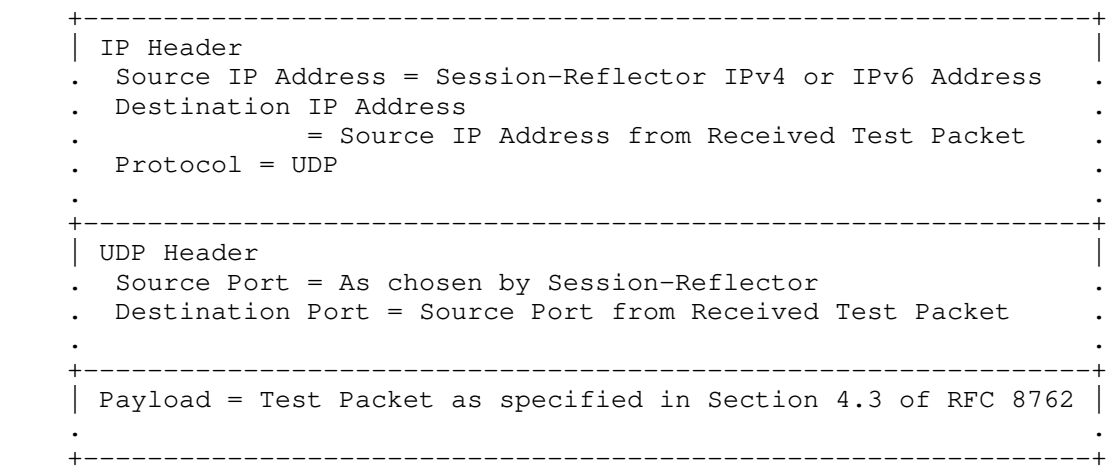


Figure 5: Example Session-Reflector Test Packet

4.2.1. One-way Measurement Mode

In one-way delay measurement mode, a reply test packet as shown in Figure 5 is transmitted by the STAMP Session-Reflector, for both links and end-to-end SR Policies. The reply test packet may be transmitted on the same path or a different path in the reverse direction.

The STAMP Session-Sender address may not be reachable via IP route from the STAMP Session-Reflector. The STAMP Session-Sender in this case can send its reachability path information to the STAMP Session-Reflector using the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

In this mode, as per Reference Topology, all timestamps T1, T2, T3, and T4 are collected by the test packets. However, only timestamps T1 and T2 are used to measure one-way delay as (T2 - T1). The one-way delay measurement mode requires the clock on the Session-Sender and Session-Reflector to be synchronized.

4.2.2. Two-way Measurement Mode

In two-way (i.e. round-trip) delay measurement mode, a reply test packet as shown in Figure 5 is transmitted by the STAMP Session-Reflector on the same path in the reverse direction, e.g. on the reverse direction link or associated reverse SR path [I-D.ietf-pce-sr-bidir-path].

For two-way delay measurement mode for links, the STAMP Session-Reflector needs to transmit the reply test packet on the same link where the test packet is received. The STAMP Session-Sender can request in the test packet to the STAMP Session-Reflector to transmit the reply test packet back on the same link using the Control Code Sub-TLV in the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

For two-way delay measurement mode for end-to-end SR paths, the STAMP Session-Reflector needs to transmit the reply test packet on a specific reverse path. The STAMP Session-Sender can request in the test packet to the STAMP Session-Reflector to transmit the reply test packet back on a given reverse path using a Segment List sub-TLV in the Return Path TLV defined in [I-D.gandhi-ippm-stamp-srpm].

In this mode, as per Reference Topology, all timestamps T1, T2, T3, and T4 are collected by the test packets. All four timestamps are used to measure two-way delay as ((T4 - T1) - (T3 - T2)). When clock synchronization on the Session-Sender and Session-Reflector nodes is not possible, the one-way delay can be derived using two-way delay divided by two.

4.2.2.1. Session-Reflector Test Packet for SR-MPLS Policies

The content of an example STAMP Session-Reflector reply test packet transmitted on the same path as the data traffic flow under measurement for two-way delay measurement of an end-to-end SR-MPLS Policy is shown in Figure 6.

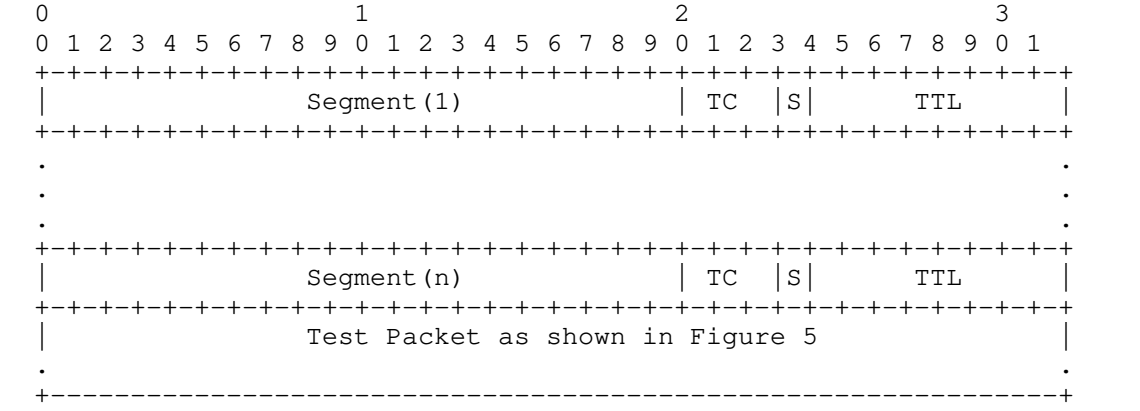


Figure 6: Example Session-Reflector Test Packet for SR-MPLS Policy

4.2.2.2. Session-Reflector Test Packet for SRv6 Policies

The content of an example STAMP Session-Reflector reply test packet transmitted on the same path as the data traffic flow under measurement for two-way delay measurement of an end-to-end SRv6 Policy with SRH is shown in Figure 7.

The procedure defined for Upper-Layer Header processing for SRv6 End SIDs in Section 4.1.1 in [RFC8986] is used to process the IPv6/UDP header in the received reply test packets on the Session-Sender.

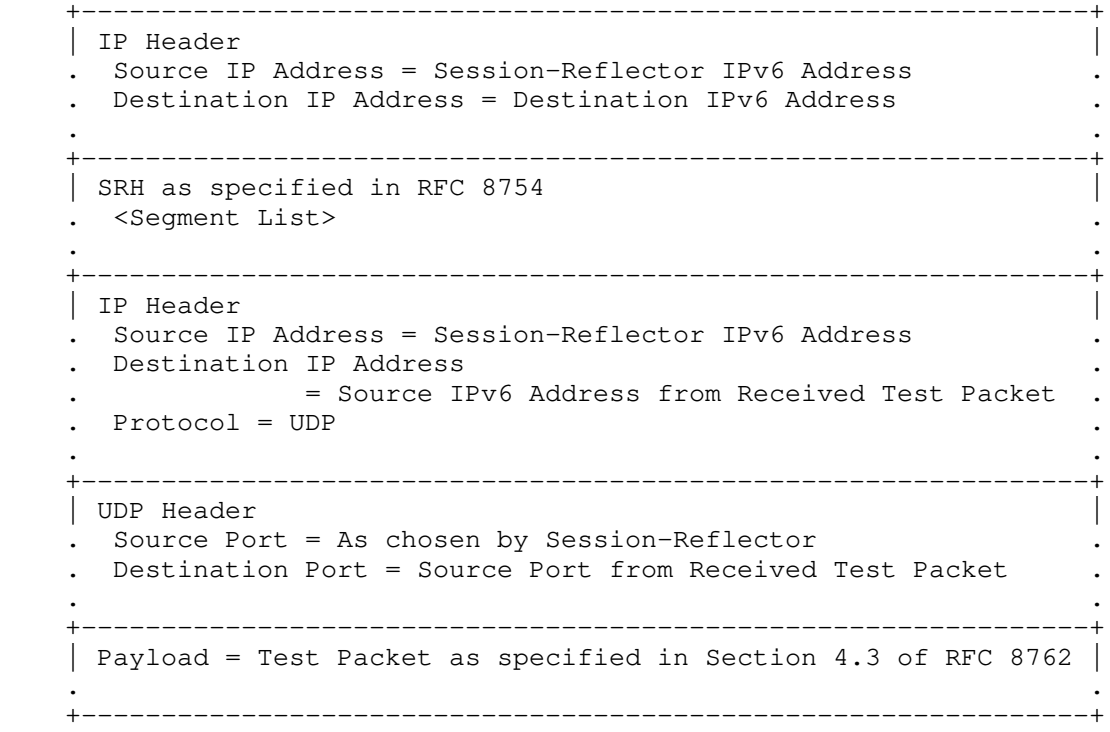


Figure 7: Example Session-Reflector Test Packet for SRv6 Policy

4.2.3. Loopback Measurement Mode

The STAMP Session-Sender test packets are transmitted in loopback mode to measure loopback delay of a bidirectional circular path. In this mode, the received Session-Sender test packets are not punted out of the fast path in forwarding (to slow path or control-plane) at the STAMP Session-Reflector. In other words, the Session-Reflector does not process them and generate reply test packets.

The IP header of the STAMP Session-Sender test packet contains the Destination Address equals to the STAMP Session-Sender address and the Source Address equals to the STAMP Session-Reflector address. The Session-Sender sets the Reflector UDP port that it uses to receive the test packet. Optionally, the STAMP Session-Sender test packet can carry the return path information (e.g. return path label stack for SR-MPLS) as part of the SR header.

The Session-Sender can use the SSID field in the reply test packet and/ or local configuration to know that the test session is using the loopback mode. As the reply test packet is not generated by the STAMP Session-Reflector, the STAMP Session-Sender ignores the 'Session-Sender Sequence Number', 'Session-Sender Timestamp', 'Session-Sender Error Estimate', and 'Session-Sender TTL' in the received test packet. The Session-Sender sets these fields to 0 upon transmission.

In this mode, as per Reference Topology, the timestamps T1 and T4 are collected by the test packets. Both these timestamps are used to measure loopback delay as (T4 - T1). When STAMP capability on the Session-Reflector node is not possible, the one-way delay can be derived using loopback delay divided by two. In this mode, the responder node processing time component reflects only the time required to loop the test packet from the incoming interface to the outgoing interface in forwarding plane.

4.3. Delay Measurement for P2MP SR Policies

The Point-to-Multipoint (P2MP) SR path that originates from a root node terminates on multiple destinations called leaf nodes (e.g. P2MP SR Policy [I-D.ietf-pim-sr-p2mp-policy]).

The procedures for delay and loss measurement described in this document for end-to-end P2P SR Policies are also equally applicable to the P2MP SR Policies. The procedure for one-way measurement is defined as following:

- The STAMP Session-Sender root node transmits test packets using the Tree-SID defined in [I-D.ietf-pim-sr-p2mp-policy] for the P2MP SR-MPLS Policy as shown in Figure 8. The STAMP Session-Sender test packets may contain the replication SID as defined in [I-D.ietf-spring-sr-replication-segment].

- The Destination Address is set to the loopback address from the range 127/8 for IPv4, or the loopback address ::1/128 for IPv6.

- Each STAMP Session-Reflector leaf node transmits its node address in the Source Address of the reply test packets shown in Figure 5.

This allows the STAMP Session-Sender root node to identify the STAMP Session-Reflector leaf nodes of the P2MP SR Policy.

- The P2MP root node measures the delay for each P2MP leaf node individually.

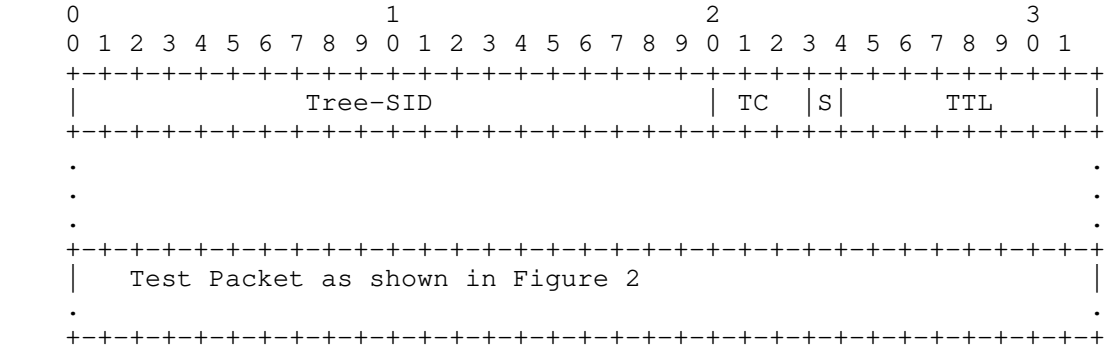


Figure 8: Example Session-Sender Test Packet with Tree-SID for SR-MPLS Policy

The considerations for two-way mode for P2MP SR Policy (e.g. for co-routed bidirectional SR-MPLS path) are outside the scope of this document.

4.4. Additional STAMP Test Packet Processing Rules

The processing rules described in this section are applicable to the STAMP test packets for links and end-to-end SR paths including SR Policies.

4.4.1. TTL

The TTL field in the IPv4 and MPLS headers of the STAMP Session-Sender and STAMP Session-Reflector test packets is set to 255, except in the following cases.

When using the Session-Reflector IPv4 Address from the range 127/8, the TTL field in the IPv4 header is set to 1, for otherwise, encapsulated packets.

For link delay, the TTL field in the STAMP test packet is set to 1 in one-way and two-way delay measurement modes.

4.4.2. IPv6 Hop Limit

The Hop Limit field in the IPv6 and SRH headers of the STAMP Session-Sender and STAMP Session-Reflector test packets is set to 255, except in the following cases.

When using the Session-Reflector IPv6 Address of loopback address ::1/128, the Hop Limit field in the IPv6 header is set to 1, for otherwise, encapsulated packets.

For link delay, the Hop Limit field in the STAMP test packet is set to 1 in one-way and two-way delay measurement modes.

4.4.3. Router Alert Option

The Router Alert IP option (RAO) [RFC2113] is not set in the STAMP test packets for links and end-to-end SR paths.

4.4.4. UDP Checksum

For IPv4 test packets, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the Session-Sender may set the UDP checksum value to 0 [RFC8085].

For IPv6 test packets, where the hardware is not capable of re-computing the UDP checksum or adding checksum complement [RFC7820], the Session-Sender and Session-Reflector may use the procedure defined in [RFC6936] for the UDP checksum.

5. Packet Loss Measurement for Links and SR Paths

The procedure described in Section 4 for delay measurement using STAMP test packets can be used to detect (test) packet loss for links and end-to-end SR paths. The Sequence Number field in the STAMP test packet is used as described in Section 4 "Theory of Operation" where Stateful and Stateless Session-Reflector operations are defined [RFC8762], to detect round-trip, near-end (forward) and far-end (backward) packet loss.

This method can be used for inferred packet loss measurement, however, it does not provide accurate data packet loss metric.

6. Direct Measurement for Links and SR Paths

The STAMP "Direct Measurement" TLV (Type 5) defined in [RFC8972] can be used in SR networks for data packet loss measurement. The STAMP test packets with this TLV are transmitted using the procedures

described in Section 4 to collect the transmit and receive counters of the data flow for the links and end-to-end SR paths.

The PSID carried in the received data packet for the traffic flow under measurement can be used to measure receive data packets (for receive traffic counter) for an end-to-end SR path on the STAMP Session-Reflector. The PSID in the received Session-Sender test packet header can be used to associate the receive traffic counter on the Session-Reflector for the end-to-end SR path.

The STAMP "Direct Measurement" TLV (Type 5) lacks the support to identify the Block Number of the Direct Measurement traffic counters, which is required for Alternate-Marking Method [RFC8321] for accurate data packet loss metric.

7. Session State for Links and SR Paths

The STAMP test session state allows to know if the performance measurement test is active. The threshold-based notification may not be generated if the delay values do not change significantly. For an unambiguous monitoring, the controller needs to distinguish the cases whether the performance measurement is active, or delay values are not changing to cross threshold.

The STAMP test session state initially is declared active when one or more reply test packets are received at the STAMP Session-Sender. The STAMP test session state is declared idle (or failed) when consecutive N number of reply test packets are not received at the STAMP Session-Sender, where N is locally provisioned value.

8. ECMP Support for SR Policies

An SR Policy can have ECMPs between the source and transit nodes, between transit nodes and between transit and destination nodes. Usage of Anycast SID [RFC8402] by an SR Policy can result in ECMP paths via transit nodes part of that Anycast group. The test packets need to be transmitted to traverse different ECMP paths to measure end-to-end delay of an SR Policy.

Forwarding plane has various hashing functions available to forward packets on specific ECMP paths. The mechanisms described in [RFC8029] and [RFC5884] for handling ECMPs are also applicable to the delay measurement.

In IPv4 header of the STAMP Session-Sender test packets, sweeping of Session-Reflector Address from the range 127/8 can be used to exercise ECMP paths. In this case, both the forward and the return paths must be SR-MPLS paths when using the loopback mode.

As specified in [RFC6437], Flow Label field in the outer IPv6 header can also be used for sweeping to exercise different IPv6 ECMP paths.

The "Destination Node Address" TLV [I-D.gandhi-ippm-stamp-srpm] can be carried in the STAMP Session-Sender test packet to identify the intended Session-Reflector, for example, in case of using IPv4 Session-Reflector Address from 127/8 range when the STAMP test packet is encapsulated by a tunneling protocol or an MPLS Segment list. The STAMP Session-Reflector must not transmit reply test packet if it is not the intended destination node in the "Destination Node Address" TLV [I-D.gandhi-ippm-stamp-srpm].

9. Security Considerations

The performance measurement is intended for deployment in well-managed private and service provider networks. As such, it assumes that a node involved in a measurement operation has previously verified the integrity of the path and the identity of the far-end STAMP Session-Reflector.

If desired, attacks can be mitigated by performing basic validation and sanity checks, at the STAMP Session-Sender, of the counter or timestamp fields in received measurement reply test packets. The minimal state associated with these protocols also limits the extent of measurement disruption that can be caused by a corrupt or invalid packet to a single test cycle.

Use of HMAC-SHA-256 in the authenticated mode protects the data integrity of the test packets. SRv6 has HMAC protection authentication defined for SRH [RFC8754]. Hence, test packets for SRv6 may not need authentication mode. Cryptographic measures may be enhanced by the correct configuration of access-control lists and firewalls.

The security considerations specified in [RFC8762] and [RFC8972] also apply to the procedures described in this document.

When using the procedures defined in [RFC6936], the security considerations specified in [RFC6936] also apply.

10. IANA Considerations

This document does not require any IANA action.

11. References

11.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

[RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-Way Active Measurement Protocol Optional Extensions", RFC 8972, DOI 10.17487/RFC8972, January 2021, <https://www.rfc-editor.org/info/rfc8972>.

[I-D.gandhi-ippm-stamp-srpm] Gandhi, R., Filsfils, C., Voyer, D., Chen, M., and B. Janssens, "Simple TWAMP (STAMP) Extensions for Segment Routing Networks", draft-gandhi-ippm-stamp-srpm-03 (work in progress), April 2021.

11.2. Informative References

[IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <https://www.rfc-editor.org/info/rfc2113>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <https://www.rfc-editor.org/info/rfc5884>.

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

[RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <https://www.rfc-editor.org/info/rfc6936>.

[RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <https://www.rfc-editor.org/info/rfc7404>.

[RFC7820] Mizrahi, T., "UDP Checksum Complement in the One-Way Active Measurement Protocol (OWAMP) and Two-Way Active Measurement Protocol (TWAMP)", RFC 7820, DOI 10.17487/RFC7820, March 2016, <https://www.rfc-editor.org/info/rfc7820>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

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

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-replication-segment] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", draft-ietf-spring-sr-replication-segment-04 (work in progress), February 2021.

[I-D.ietf-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-02 (work in progress), February 2021.

[I-D.ietf-spring-mpls-path-segment] Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-04 (work in progress), April 2021.

[I-D.ietf-spring-srv6-path-segment] Li, C., Cheng, W., Chen, M., Dhody, D., and R. Gandhi, "Path Segment for SRv6 (Segment Routing in IPv6)", draft-ietf-spring-srv6-path-segment-00 (work in progress), November 2020.

[I-D.ietf-pce-sr-bidir-path] Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-ietf-pce-sr-bidir-path-05 (work in progress), January 2021.

[I-D.ietf-ippm-stamp-yang] Mirsky, G., Min, X., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-07 (work in progress), March 2021.

[IEEE802.1AX]
IEEE Std. 802.1AX, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", November 2008.

Acknowledgments

The authors would like to thank Thierry Couture for the discussions on the use-cases for Performance Measurement in segment routing. The authors would also like to thank Greg Mirsky, Gyan Mishra, Xie Jingrong, and Mike Koldychev for reviewing this document and providing useful comments and suggestions. Patrick Khordoc and Radu Valceanu have helped improve the mechanisms described in this document.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Bart Janssens
Colt

Email: Bart.Janssens@colt.net

Richard Foote
Nokia

Email: footer.foote@nokia.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

X. Geng
M. Chen
F. Yang
Huawei Technologies
February 22, 2021

Segment Routing for Redundancy Protection
draft-geng-spring-sr-redundancy-protection-02

Abstract

Redundancy protection is one of the mechanisms to achieve service protection, following the principle of PREOF (Packet Replication/Elimination/Ordering Function). To empower the Segment Routing with the capability of redundancy protection, two types of Segment including Redundancy Segment and Merging Segment are introduced. The instantiation of Redundancy and Merging Segments can be applied to both segment routing over MPLS (SR-MPLS) and segment routing over IPv6 (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 .

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 August 26, 2021.

Table of Contents

1. Introduction

Service protection defined in [RFC8655] is initially required from the use cases in a variety of industries described in [RFC8578]. Together with other two techniques Resource allocation and Explicit routes, it targets to provide the deterministic flow transmission. Meanwhile, with the emerge of Cloud VR, Cloud Game, High-Definition Video applications running in the Internet, SLA (Service Level Agreement) guarantee becomes an important issue which requires new technologies and solutions for network.

Redundancy Protection is one of the mechanisms to achieve service protection, following the principle of PREOF (Packet Replication/ Elimination/Ordering Function) defined in [RFC8655]. Specifically, replicates the packets of flows into two or more copies, transports different copies through different paths in parallel, eliminates and orders the packets at end to provide redundancy protection.

Segment Routing (SR) leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions, called "segments". A segment can be associated to an arbitrary processing of the packet in the node identified by the segment.

This document extends the capabilities in SR paradigm to support the redundancy protection, including the definitions of new Segments and a variation of Segment Routing Policy. The new mechanism applies equally to both segment routing with MPLS data plane (SR-MPLS) and segment routing with IPv6 data plane (SRv6).

2. Terminology and Conventions

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 [I-D.ietf-spring-srv6-network-programming] and[RFC2119].

Redundancy Node: the start point of redundancy protection, which is a network device that could implement packet replication.

Merging Node: the end point of redundancy protection, which is a network node that could implement packet elimination and ordering (optionally).

Redundancy Policy: an extended SR policy which includes more than one active segment lists to support redundancy protection.

Flow Identification: information in SR data service to indicate one flow.

Sequence Number: information in SR data service to indicate the packet sequence of one flow.

Editor’s Note: Similar mechanism is defined as "Service Protection" in the [RFC8655]. In this document, we define a new term "Redundancy Protection" to distinguish with other service protection method. Some of the terms are similar as [RFC8655].

3. Redundancy Protection in Segment Routing Scenario

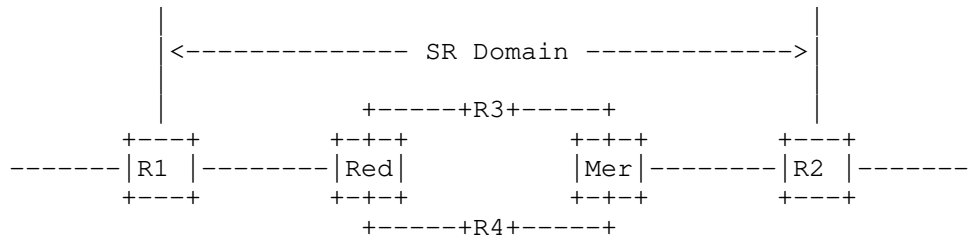


Figure 1: Example Scenario of Redundancy Protection in SR Domain

This figure shows an example of redundancy protection used in SR domain. When a flow is sent into SR domain, the process is:

1) R1 receives the traffic flow and encapsulates packets with a list of segments destined to R2, which is instantiated as a stack of MPLS labels or an ordered list of SRv6 SIDs. R1, R2, R3, R4, Red and Mer are SR-capable nodes.

2) R1 encapsulates the flow identification and sequence number to the packets. Flow identification identifies the specific flow, and sequence number distinguishes the packet sequence of a flow.

3) When the packet flow arrives in Red node, known as Redundancy Node, one flow is replicated into two copies. Each copy of flow is encapsulated with different newly-defined list of SIDs, and the last SID is always pointed to the SID of Mer node, known as Merging Node.

4) When the original flow and replicated flow go through different paths till Mer node, the first received packet of the flow is transmitted from Merging Node to R2, and the redundant packets are eliminated.

5) When there is any failures happened in one path, the service continues to deliver through the other path without break.

6) Sometimes, the packet will arrive out of order because of redundancy protection, the function of reordering may be necessary in the Merging Node.

In this example, service protection is supported by utilizing two packet flows transmitted over two forwarding paths. For a unidirectional flow, Red node supports replication function, and Mer node supports elimination and ordering functions.

4. Segment to Support Redundancy Protection

To achieve the Packet Replication/Elimination/Ordering Function, Redundancy Segment and Merging Segment are introduced.

4.1. Redundancy Segment

Redundancy Segment is a variation of Binding SID, and associated with a Redundancy Policy on redundancy node. Redundancy segment is associated with service instructions, indicating the following operations:

- Steering the packet into the corresponding redundancy policy
- Packet replication and encapsulation based on the redundancy policy, e.g., the number of replication copies

4.1.1. SR over MPLS

In the case of SR over MPLS, when the Active Segment is a Redundancy Segment, a redundancy policy is associated. According to the information of candidate paths in redundancy policy, packets are replicated, and the Incoming redundancy segment is swapped with different stacks of MPLS labels to forward the packet in different paths.

4.1.2. SRv6

In the case of SRv6, a new behavior End.R for Redundancy Segment is defined. In the following description, End.R behavior is specified in the encapsulation mode. The End.R behavior in the insertion mode is for further study.

When an SRv6-capable node (N) receives an IPv6 packet whose destination address matches a local IPv6 address instantiated as an SRv6 SID (S), and S is a Redundancy SID, N does:

```
S01.  When an SRH is processed {
S02.     If (Segments Left>0)   {
S03.          Decrement IPv6 Hop Limit by 1
S04.          Decrement Segments Left by 1
S05.          Update IPv6 DA with Segment List[Segments Left]
S06.          Create two new IPv6 headers with SRH-1 and SRH-2 respectively
S07.          Insert different policy-instructed segment lists into SRH-1 and SRH
-2
S08.          Create a duplication of the incoming packet
S09.          Encapsulate the original packet with the new IPv6+SRH-1 header
S10.          Encapsulate the duplicate packet with the new IPv6+SRH-2 header
S11.          Set IPv6 SA as the local address of this node
S12.          Set IPv6 DA of IPv6+SRH-1 to the first segment of SRH-1 SL
S13.          Set IPv6 DA of IPv6+SRH-2 to the first segment of SRH-2 SL
S14.          Copy flow identification and sequence number from current SRH to SR
H-1
S15.          Copy flow identification and sequence number from current SRH to SR
H-2
S16.          Set the outer Payload Length, Traffic Class, Flow Label,Hop Limit a
nd Next-Header fields
S17.          Submit the packet to the egress IPv6 FIB lookup and transmit
S18.     }
S19.     ELSE {
S20.          Drop the packet
S21.     }
S22.  }
```

4.2. Merging Segment

Merging Segment is associated with service instructions, indicates the following operations:

- Packet merging and elimination: forward the first received packets and eliminate the redundant packets

- Packet ordering(optional): reorder the packets if the packet arrives out of order

4.2.1. SR over MPLS

In the case of SR over MPLS, when the Active Segment is a Merging Segment and this packet is not a redundant packet, a CONTINUE operation is applied. Incoming merging segment is swapped with next segment.

4.2.2. SRv6

In the case of SRv6, a new behavior End.M for Merging Segment is defined.

When an SRv6-capable node (N) receives an IPv6 packet whose destination address matches a local IPv6 address instantiated as an SRv6 SID (S), and S is a Merging SID, N does:

```
S01.  When an SRH is processed {
S02.     If (Segments Left==0)   {
S03.          Acquire the sequence number of received packet and lookup it in a l
ocal table
S04.             If (the sequence number is not existed in table ) {
S05.                   Store the packet and record the sequence number in table
S06.                   Remove the outer IPv6+SRH header
S07.                   Decrement IPv6 Hop Limit by 1 in inner SRH
S08.                   Decrement Segments Left by 1 in inner SRH
S09.                   Update IPv6 DA with Segment List[Segments Left] in inner S
RH
S10.                   Submit the packet to the egress IPv6 FIB lookup and transm
it
S11.             }
S12.             ELSE {
S13.                   Drop the packet
S14.             }
S15.      }
S16.  }
```

5. Meta Data to Support Redundancy Protection

To support the redundancy protection function, Flow Identification and Sequence Number are required. Flow identification identifies the specific flow with target of redundancy protection. Sequence number distinguishes the packets within a flow by specifying the order of packets. The flow identification and sequence number is RECOMMENDED to be added at the ingress of SR domain, and MUST be added before the redundancy node. The flow identification and sequence number is carried in the service packets along the different paths to merging node. Merging node removes flow identifier and sequence number once the elimination and ordering is accomplished. Thus, an encapsulation of flow identification and sequence number is required to be defined in both SR over MPLS and SRv6 data plane.

6. Segment Routing Policy to Support Redundancy Protection

Redundancy Policy is a variation of SR Policy, and is identified through the tuple <redundancy node, redundancy ID, merging node>. Redundancy node is specified as IPv4/IPv6 address of the headend, which is able to do packet replication. Merging node is specified as IPv4/IPv6 address of the endpoint, which is able to do packet elimination and ordering (optional). Redundancy ID could be a specified value of "color" define in section 2.1 of [I-D.ietf-spring-segment-routing-policy], which indicates the SR policy as a redundancy policy. Redundancy ID could also be used to

distinguish different redundancy policies sharing the same redundancy node and merging node.

Redundancy Policy extends SR policy to include more than one ordered lists of segments between redundancy node and merging node, and all the ordered lists of segments are used at the same time to steer the copies of flow into different paths. In redundancy policy, Redundancy Segment MUST be specified, and the last segment of each ordered list of segments MUST be Merging Segment.

7. IANA Considerations

This document requires registration of End.R behavior and End.M behavior in "SRv6 Endpoint Behaviors" sub-registry of "Segment Routing Parameters" registry.

8. Security Considerations

TBD

9. Acknowledgements

The authors would like to thank Bruno Decraene, Ron Bonica, and James Guichard for their valuable comments.

10. References

10.1. Normative References

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

Authors' Addresses

Xuesong Geng
Huawei Technologies
Beijing
China

Email: gengxuesong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
Beijing
China

Email: mach.chen@huawei.com

Fan Yang
Huawei Technologies
Beijing
China

Email: shirley.yangfan@huawei.com

SPRING
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

S. Hegde
C. Bowers
Juniper Networks Inc.
X. Xu
Alibaba Inc.
A. Gulko
EdwardJones
A. Bogdanov
Google Inc.
J. Uttaro
ATT
L. Jalil
Verizon
M. Khaddam
Cox communications
A. Alston
Liquid Telecom
LM. Contreras
Telefonica

February 22, 2021

Seamless SR Problem Statement
draft-hegde-spring-mpls-seamless-sr-05

Abstract

This draft documents a set of use cases and requirements for end-to-end intent-based paths spanning multi-domain packet networks. The document explicitly focuses on use cases that require high scale and availability, which will likely benefit from distributed solutions. It is intended that the requirements in this document serve as a basis for future IETF work to develop distributed solutions for inter-domain intent-based transport paths.

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 August 26, 2021.

Table of Contents

1. Introduction

Evolving trends in wireless access technology, cloud applications, virtualization, and network consolidation all contribute to the increasing demands being placed on a common packet network. In order to meet these demands, a given network will need to scale horizontally in terms of its bandwidth, absolute number of nodes, and geographical extent. The same network will need to scale vertically in terms of the different services that it needs to simultaneously support.

In order to operate networks with large numbers of devices, network operators organize networks into multiple smaller network domains. Each network domain typically runs an IGP which has complete visibility within its own domain, but limited visibility outside of

its domain. Network operators will continue to use multiple domains to scale horizontally. These multi-domain networks will also need to scale vertically, to allow a common multi-domain network to carry all of an organization's services.

Evolving network requirements (e.g., 5G, native cloud) motivate network operators to deploy tunnels that span multiple AS's and maintain specific transport characteristics (e.g., bandwidth, latency). There is a need to provide flexible, scalable, and reliable end-to-end connectivity for multiple services across independent network domains.

2. Large scale networks

2.1. Service provider networks

Service Provider networks can contain many nodes distributed over a large geographic area. 5G networks can include as many as one million nodes, with the majority of those being radio access nodes. Radio and access nodes may be constrained by their memory and processing capabilities.

Service provider transport networks use multiple domains to support scalability. For this analysis, we consider a representative network design with four level of hierarchy: access domains, pre-aggregation domains, aggregation domains and a core. (See Figure 1). The separation of domains internal to the service provider can be performed by using either IGP or BGP.

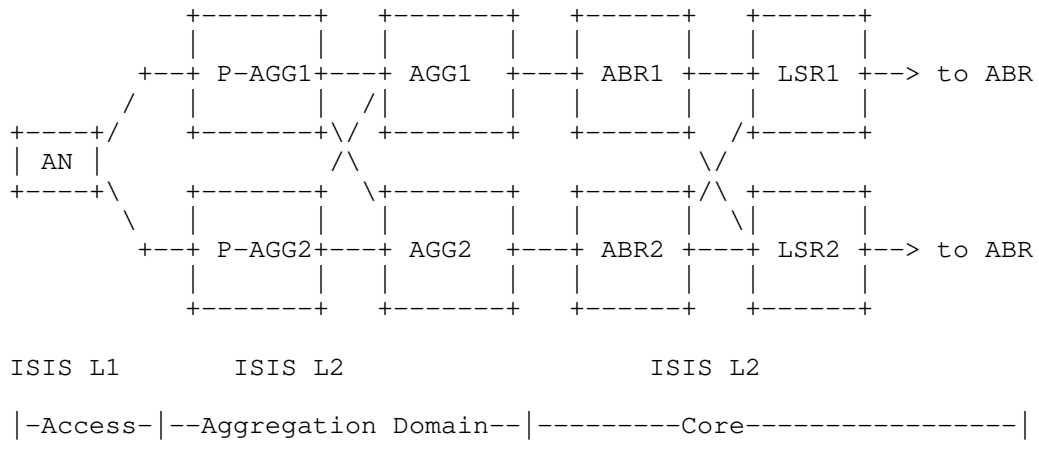


Figure 1: 5G network

5G networks support a variety of service use cases that require end-to-end slicing. In certain cases the end-to-end connectivity requires the ability to forward over intent-based paths. The inter-domain solution should support end-to-end Service Level Objectives(SLO) to allow the creation of network slices.

2.2. Cloud provider WAN networks

As WAN networks grow beyond several thousand nodes, it is often useful to divide the network into multiple IGP domains, as illustrated in Figure 2. Separate IGP domains increase service availability by establishing a constrained failure domain. Smaller IGP domains may also improve network performance and health by reducing the device scale profile (including protocol and FIB scale).

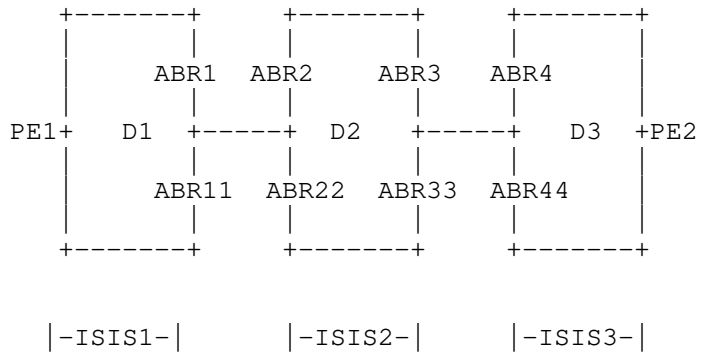


Figure 2: WAN Network

These large WAN networks often cross national boundaries. In order to meet data sovereignty requirements, operators need to maintain strict control over end-to-end traffic-engineered (TE) paths. A goal of a distributed inter-domain solution is to be able to create highly constrained inter-domain TE paths in a scalable manner.

Some deployments may use a centralized controller to acquire the topologies of multiple domains and build end-to-end constrained paths. This centralized approach can be scaled with hierarchical controllers. However, there is still significant risk of a loss of network connectivity to one or more controllers, which can result in a failure to satisfy the strict requirements of data sovereignty. The network should have pre-established TE paths end-to-end that don't rely on controllers in order to address these failure scenarios.

2.3. Data Center WAN Networks

Data centers are playing an increasingly important role in providing access to information and applications. Geographically diverse data centers usually connect via a high speed, reliable and secure DC WAN core network.

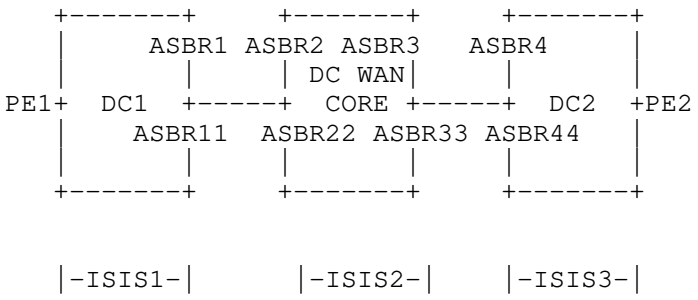


Figure 3: DCI Network

In many DC WAN deployments, applications require end-to-end path diversity and end-to-end low latency paths. The DC WAN networks may consist of large number of devices owing to global presence. In some DC WAN deployments the tunneling mechanisms used within the data centers are the same as those used in the DC WAN core. For example, a network may use MPLS in both data center and DC WAN core. Or it may use SRv6 in both data center and DC WAN core. This can simplify network deployments.

However, in some DC WAN deployments the traffic within data centers and the traffic over the DC WAN core use different tunneling mechanisms, such as SRv6 in the data center and MPLS in the DC WAN core. It is important for DC WAN network operators to have flexibility in the choice of tunneling mechanisms across domains.

3. Use Cases for Inter-domain Intent-based Transport

The use cases for inter-domain intent-based packet transport described in this section are intended to provide motivation for the requirements that follow.

3.1. Inter-domain Data Sovereignty

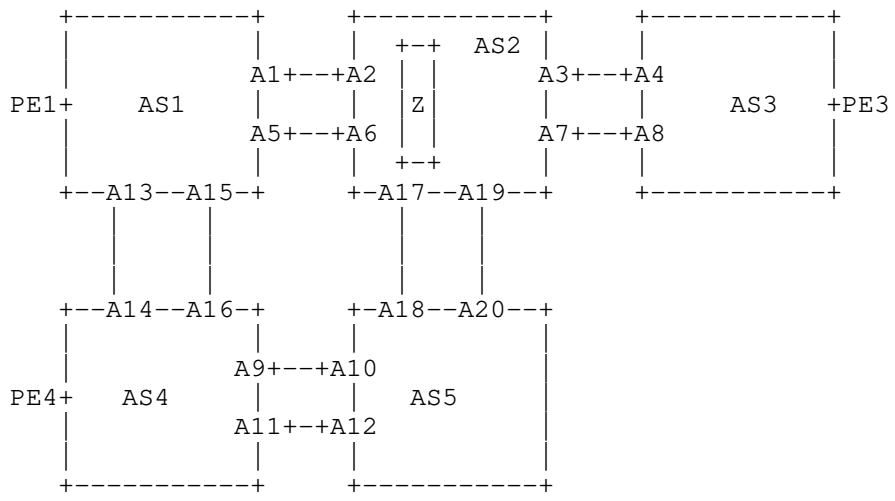


Figure 4: Multi domain Network

Figure Figure 4 depicts a WAN with multiple ASes. Each AS is resides serves a continent (e.g., Asia). Certain traffic from PE1 (in AS1) to PE3 (in AS3) must not traverse country Z in AS2. However, all paths from AS1 to AS3 traverse AS 2. The inter-domain solution should provide end-to-end path creation that traverses AS 2 but avoids country Z.

3.2. Inter-domain Low-Latency Services

Service provider networks running L2 and L3VPNs carry traffic for particular VPNs on low-latency paths that traverse multiple domains.

3.3. Network Mergers

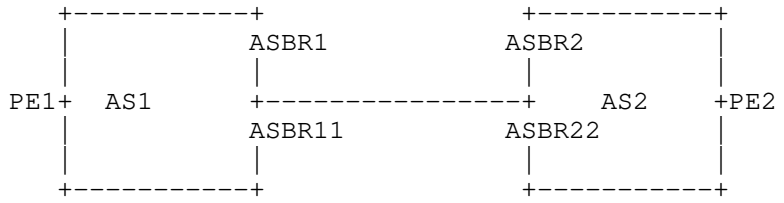


Figure 5: Network Mergers

In diagram Figure 5 above, AS1 and AS2 which were previously under independent administration, merge to come under a single administration. The network operator may decide to merge the two domains into single AS which would need bigger operational effort. Network operators may also retain the two ASes and build end-to-end paths across the two Ases. In this case, the paths in AS1 and AS2 corresponding to the same intent may use different representations in the two ASes. In some cases, organizations may continue to use option A or option B [RFC4364] style interconnectivity in which case the inter-domain solution should satisfy intent of the path on inter-domain links for the service prefixes. In other cases, organizations may prefer to use option C style connectivity from PE1 to PE2. In this case, an inter-domain solution should provide effective mechanisms to translate intent across domains without requiring renumbering of the intent representation.

3.4. Inter-domain Service Function Chaining

[RFC7665] defines service function chaining as an ordered set of service functions and automated steering of traffic through this set of service functions. There could be a variety of service functions such as firewalls, parental control, CGNAT etc. In 5G networks these functions may be completely virtualized or could be a mix of virtualized functions and physical appliances. It is required that the inter-domain solution caters to the service function chaining requirements. The service functions may be virtualized and spread across different data centers attached to different domains.

3.5. AS Confederation

BGP confederation allows the division of a public AS in multiple sub-AS, usually with private identifiers. From outside, the confederation is seen as a single and common AS, the public one. BGP sessions are maintained among sub-AS. In the internals of the confederation, each sub-AS can be configured and run autonomously, even though some BGP parameters (like e.g. LOCAL_PREF or MED) are preserved across sub-AS. Thus, it can be of interest to define end-to-end paths of specific characteristics in terms of SLOs across the sub-AS as well as internally to each sub-AS.

3.6. Inter-domain Multicast Use cases

Multicast services such as IPTV and multicast VPN also need to be supported across a multi-domain service provider network.

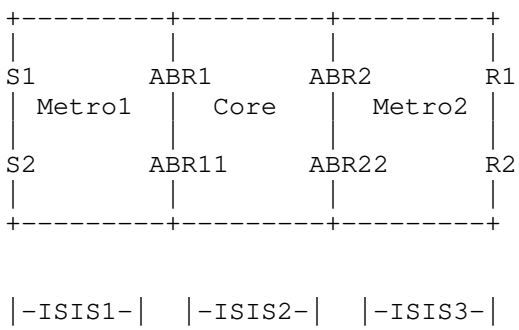


Figure 6: Multicast usecases

Figure 6 shows a simplified multi-domain network supporting multicast. Multicast sources S1 and S2 lie in a different domain from the receivers R1 and R2. Using multiple IGP domains presents a problem for the establishment of multicast replication trees. Typically, a multicast receiver does a reverse path forwarding (RPF) lookup for a multicast source. One solution is to leak the routes for multicast sources across the IGP domains. However, this can compromise the scaling properties of the multi-domain architecture. A distributed inter-domain solution should accommodate a mixture of existing and newer technologies to better facilitate coexistence and migration. A distributed solution should avoid leaking RPF routes into the IGP domains.

4. Requirements

The requirements described in this document are mostly applicable to network under a single administrative domain that are organized into multiple network domains. The requirements are also applicable to multi-AS networks with closely cooperating administration.

4.1. AS and IGP Domain Models

This section describes three different ways that multi-domain networks are organized today. The requirements in subsequent sections are applicable to all three types of multi-domain networks described below.

4.1.1. Multiple ASes connected with E-BGP

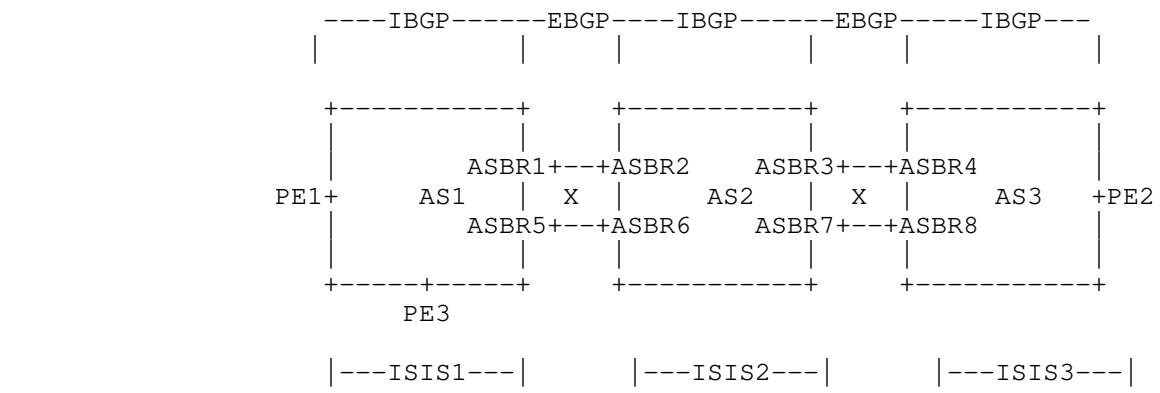


Figure 7: Multiple ASes connected with E-BGP

The above diagram Figure 7 shows three different ASes (AS1, AS2 and AS3.) ASBR1 to ASBR8 are border nodes between the ASes. A given ASBR runs E-BGP sessions with the ASBRs in adjacent ASes. The ASBR also runs I-BGP sessions with other ASBRs in the same AS. Route reflectors can also be used to achieve this full mesh of I-BGP information exchange.Similar scenario applies when considering BGP confederations [RFC5065].

4.1.2. Single AS multiple IGP domains

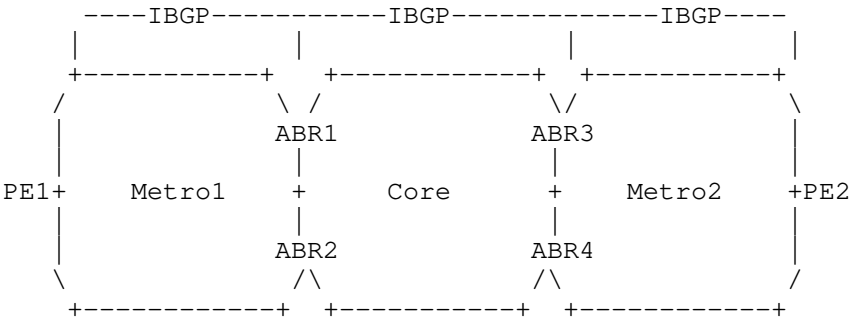


Figure 8: Single AS with Multiple IGP domains

The above diagram Figure 8 shows three different IGP domains, Metro1, Core, and Metro2. The three IGP domains may be realized with

multiple levels in ISIS or multiple areas in OSPF. They can also be realized using separate IGP instances.

This single-AS network uses I-BGP sessions. ABRs and PEs achieve a full mesh of I-BGP information sharing by configuring the ABRs as inline route reflectors.

4.1.3. Single AS, Multiple IGP domains with no common border node

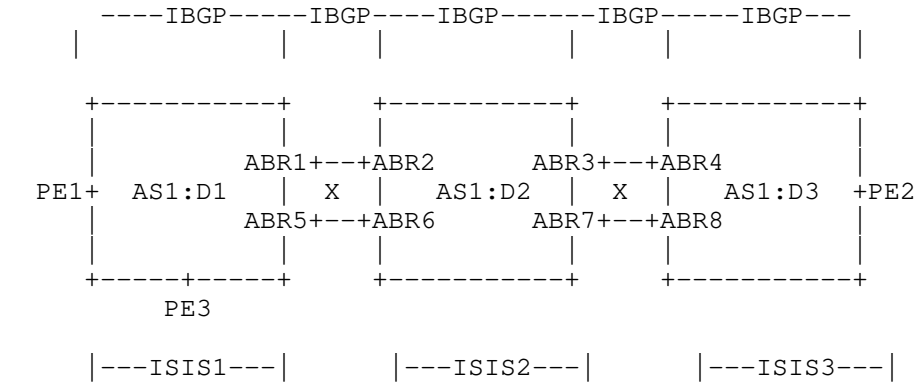


Figure 9: Single AS multiple IGP domains with no common Border node

The above diagram Figure 9 shows a single AS1 with three different IGP domains, D1, D2, and D3. ABR1 to ABR8 are border nodes for the IGP domains and they participate in only one IGP domain.

This single-AS network uses I-BGP sessions. ABRs and PEs achieve a full mesh of I-BGP information sharing by configuring the ABRs as inline route reflectors.

4.2. Transport tunneling Requirements

4.2.1. Unicast tunneling Requirements

The inter-domain solution should support the following unicast tunneling mechanisms:

SR-MPLS tunnels with IPv4 underlay

SR-MPLS tunnels with IPv6 underlay

SR-MPLS tunnels with dual stack underlay

SRv6 tunneling end-to-end

Segment routing TE tunnels and color-only policies as described in [I-D.ietf-idr-segment-routing-te-policy] (both SR-MPLS and SRv6)

Flex-algo [I-D.ietf-lsr-flex-algo] (both SR-MPLS and SRv6)

Pure IP fabric (incapable of supporting MPLS or SRv6 tunneling mechanisms)

RSVP and LDP based tunnels

The inter-domain solution should support the ability to have different domains running different unicast tunneling mechanisms.

The solution should support inter-domain paths that fulfil a common intent using different unicast tunneling mechanisms in different domains.

4.2.2. Multicast tunneling Requirements

The inter-domain solution should support the following multicast tunneling mechanisms:

All of the unicast tunneling mechanisms described in Section 4.2.1 should be supported for multicast service for the purpose of ingress replication.

SR-P2MP as defined in [I-D.voyer-pim-sr-p2mp-policy]

PIM based multicast

RSVP-P2MP and mLDP [RFC6388] based tunnels

BGP based multicast (hop-by-hop or controller-driven, for native IP, labelled, or SRv6 forwarding planes)

The inter-domain solution should support the ability to have different domains running different multicast tunneling mechanisms and should not require to leak RPF routes into IGP domains.

The solution should support inter-domain paths that fulfil a common intent using different multicast tunneling mechanisms in different domains.

4.3. Inter-domain SLA Requirements

This section discusses the end-to-end constraints that intent-based inter-domain path may have to adhere to. The requirements described in this section are applicable to the three types of AS and IGP domain partitioning described in Section 4.1.

4.3.1. Latency, Delay Variation, and Link Loss Constraints

Link delay, delay variation and link loss values can be advertised within a domain using the IGP as described in [RFC8570]. Within an IGP domain, minimum latency, minimum delay variation, and minimum link loss paths can be built using flex-algo [I-D.ietf-lsr-flex-algo]. The end-to-end low latency, low delay variation, or low link loss path requires accumulated metrics for latency, delay variation, and link loss.

The solution should allow the creation of inter-domain paths with low values of latency as calculated over the end-to-end path. It is not necessary that the solution produce the absolute minimum end-to-end latency, delay variation, or link loss path. However, the solution should provide the ability to balance scalability with optimality.

Best path selection at any intermediate border node should be allowed.

The inter-domain solution should allow advertising multiple paths end-to-end and compare the accumulated metric across all of the paths at the ingress.

4.3.2. Bandwidth Constraints

A distributed solution should support the creation of inter-domain paths using intra-domain bandwidth guaranteed paths.

A distributed solution may support optimized path placement with end-to-end bandwidth guarantees.

4.3.3. Link Inclusion/Exclusion Constraints

The links are associated with link-affinity or admin-groups. The link-affinity can be used to indicate a characteristic of a link, such as capacity, encryption, geography, etc. The inter-domain solution should support the creation of paths across different domains that satisfy link inclusion/exclusion constraints. The link constraints should also be satisfied for inter-domain links, such as those between ASBRs.

4.3.4. Node Inclusion/Exclusion Constraints

Creating an inter-domain path that includes or excludes a certain set of nodes in each domain should be supported. The inter-domain solution should be independent of the mechanisms used to achieve the node inclusion/exclusion constraints within a domain. For example, an RSVP-based domain may use link affinities to achieve node exclusion constraints, while an SR-based domain may use flex-algo, which natively supports excluding nodes.

4.3.5. Domain Inclusion/Exclusion Constraints

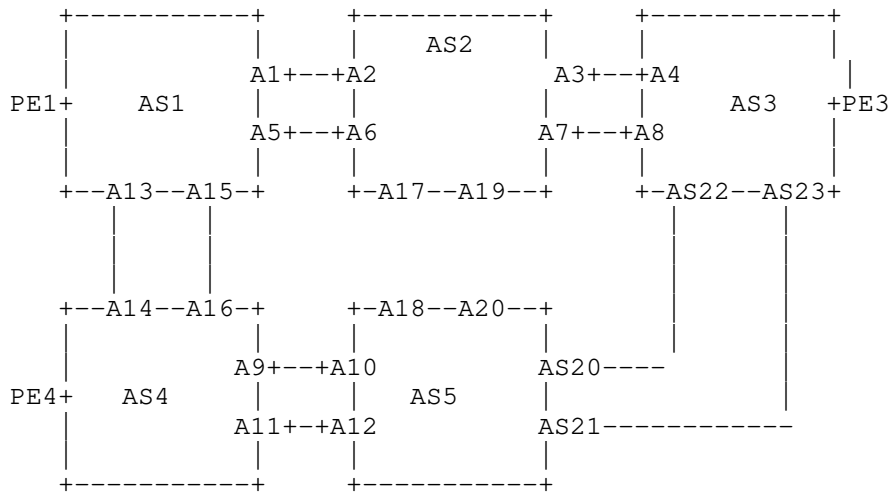


Figure 10: Multi-domain Network

Diagram Figure 10 shows a multi-domain, multi-AS network with the possibility for AS-diverse paths. The inter-domain solution should support creation of end-to-end paths that includes/excludes a certain domain entirely. For example, a network operator should be able to use the solution to create a path from PE1 to PE3 that automatically avoids passing through nodes belonging to AS2.

4.3.6. Diverse Paths

The solution should support the creation of node and link-diverse inter-domain paths.

The intra-domain portion of the end-to-end paths should make use of existing mechanisms for computing and instantiating diverse paths within a domain.

Inter-domain links (such as those connecting ASBRs) should also be taken into account for diverse inter-domain paths.

The solution should support SRLG-aware inter-domain diverse paths.

4.3.7. Constraint applicability to a subset of domains

Use cases such as data sovereignty described in Section 3.1 require that the paths with certain constraints are applicable to only a subset of domains. In domains where a constraint is not applicable, the end-to-end path should not create any state on the internal nodes.

4.3.8. Service function chaining

Support the case where the set of service functions to be applied are deployed in single domain.

Support the case where the set of service functions to be applied are deployed across multiple domains.

Support virtualized service functions as well as service functions based on physical appliances.

Support the movement of a virtualized service function from one location to another.

Support high availability for service functions.

4.4. Multicast specific requirements

Many of the requirements above are applicable to multicast traffic as well. Some requirements need to be refined with respect to multicast. Multicast also has some unique requirements not shared by unicast. These requirements will be covered in a future version of this document.

4.5. Interoperate with BGP-LU

Seamless MPLS architecture is widely deployed and BGP-LU [RFC3107] is used to connect different domains. The inter-domain solution for intent-based paths should be interoperable with BGP-LU.

4.6. Merger and Migration Requirements

4.6.1. Option A and Option B Usecases

Options A and B require additional state on border nodes, so they are typically less scalable than option C. However, options A and B can be advantageous when it is necessary to do filtering or policing on border nodes. When option A or B is deployed, the solution should still meet the SLA requirements described in Section 4.3.

4.6.2. Inter-Domain Intent Translation

In cases where two network domains previously under different administrations merge to come under a single administration, it may be preferable to use option C connectivity between the domains. The paths that fulfill the same intent may be represented using different conventions in each domain. The inter-domain solution should support efficient translation of intent from one representation to another.

4.6.3. Native Support for Best Effort Paths

The inter-domain solution for intent-based paths should also provide the ability to create end-to-end best effort paths with accumulated IGP metric across the domains. A deployment should not require two different mechanisms to be deployed for best effort and intent-based paths.

4.6.4. Interoperate with Other tunneling Mechanisms

As described in Section 4.2.1 and Section 3.6 the inter-domain solution should support one domain having one type of tunneling mechanism and another domain having another type of tunneling mechanism. The different tunneling mechanisms may completely differ in control plane and data plane operations (e.g. SRv6 and MPLS.) The inter-domain solution should provide interoperability between various tunneling mechanisms and provide the ability to create end-to-end intent-based paths.

4.7. Scalability Requirements

The inter-domain solution should be able to support up to 1 million nodes.

The inter-domain solution should facilitate the use of access nodes with low RIB/FIB and low CPU capabilities.

The inter-domain solution should facilitate the use of access nodes with low label stacking capability.

The inter-domain solution should allow for a scalable response to network events. An individual node should only need to respond to a limited subset of network events.

Service routes on the border nodes should be minimized.

Non-MPLS versions of the inter-domain solution should support summarization of prefixes in order achieve scalability.

The inter-domain solution should facilitate filtering in order to ensure the access nodes need to receive and process only the endpoint prefixes that the access node needs to send traffic to.

The inter-domain solution should minimize state on the border nodes in order to reduce label and FIB resource consumption on border nodes.

4.8. Availability Requirements

Traffic should be Fast Reroute (FRR) protected against link, node, and SRLG failures within a domain.

Traffic should be FRR protected against border node failures.

Traffic should be FRR protected against inter-domain link failures.

Traffic should be FRR protected against egress node and egress link failures.

4.9. Operations and Automation Requirements

Each domain should be independent and should not depend on the transport technology in another domain. This allows for more flexible evolution of the network.

Basic MPLS OAM mechanisms described in [RFC8029] should be supported for MPLS based solutions.

End-to-end ping and traceroute procedures should be supported.

The ability to validate the path inside each domain should be supported.

Statistics for inter-domain intent-based paths should be supported on a per path basis on the ingress and egress PE nodes as well as border nodes.

The choice of transport tunnels that make up the inter-domain path should be derived automatically from the intent that the path fulfills.

The intent defined as color in the SR-TE architecture [I-D.ietf-idr-segment-routing-te-policy] should map automatically for all controller to router protocols such as BGP-SR-TE [I-D.ietf-idr-segment-routing-te-policy], PCEP-SR [I-D.ietf-pce-segment-routing-policy-cp], and NETCONF.

The intent should be mapped automatically from flex-algo number [I-D.ietf-lsr-flex-algo].

When access devices have CPU and memory constraints, it is useful to be able to filter prefix advertisements using policies as described in Section 4.7 For large networks it is operationally a tedious and erroneous process to manage this. The inter-domain solution should facilitate filtering the advertisements automatically, based on the service prefixes it receives from end-points.

4.10. Service Mapping Requirements

The above requirements focus on the service independent aspects of inter-domain intent-based paths. In order for different services to effectively use these paths, flexible service mapping is required. The sections below summarize the requirements needed to achieve flexible service mapping.

4.10.1. Traffic service mapping

Automated steering of traffic onto transport paths based on communities carried in the service prefix advertisements should be supported.

Steering of traffic on to transport paths based on the DSCP value carried in IPv4/IPv6 packets should be supported.

Traffic steering based on EXP bits in the MPLS header should be supported.

Traffic steering based on 5-tuple packet filter should be supported. Source address, destination address, source port, destination port and protocol fields should be allowed.

All the above traffic steering mechanisms should be supported for all common types of service traffic, including L2 VPN and L3 VPN traffic and global internet traffic.

When a path that fulfills the desired intent is not available, fallback to a path that fulfills a secondary intent should be supported.

When a path that fulfills the desired intent is not available, fallback to a best-effort path should be supported.

When a path that fulfills the desired intent is not available, the option of not using a fallback path (i,e. dropping the traffic) should be supported.

4.10.2. 1 to N service mapping

The core domain is expected to have more traffic engineering constraints as compared to metros. The ability to map the services to appropriate transport tunnels at service attachment points should be supported.

4.11. Interaction with Other Approaches

This document focuses on use cases and requirements that may benefit from a distributed solution. Many of these same use cases and requirements can be addressed with centralized approaches or other distributed TE solutions. One example of a centralized approach is described in "Interconnecting Millions of Endpoints with Segment Routing" ([RFC8604]).

Distributed and centralized approaches have inherent tradeoffs. Some networks may use a single approach. Other networks may choose to use both distributed and centralized approaches to get the benefits of both. A distributed inter-domain solution should support the requirements below:

Support scenarios where some traffic uses paths created using a centralized approach, and other traffic uses paths created using the distributed solution.

Support scenarios where part of the distributed inter-domain path is created using a centralized approach.

Support scenarios where traffic uses a centralized inter-domain solution for primary traffic, and uses a distributed inter-domain solution as a backup.

The distributed solution should not have any inherent dependencies on centralized approaches.

The distributed solution should co-exist with other distributed TE solutions.

5. Backward Compatibility

6. Security Considerations

TBD

7. IANA Considerations

8. Acknowledgements

Many thanks to Kireeti Kompella, Ron Bonica, Krzysztof Szarcowitz, Srihari Sangli, Julian Lucek, Ram Santhanakrishnan, for discussions and inputs. Thanks to Colby Barth, John Scudder, Joel Halpern for review and comments.

9. Contributors

1.Kaliraj Vairavakkalai

Juniper Networks

kaliraj@juniper.net

2. Jeffrey Zhang

Juniper Networks

zzhang@juniper.net

10. References

10.1. Normative References

[I-D.hegde-rtgwg-egress-protection-sr-networks]
Hegde, S., Lin, W., and S. Peng, "Egress Protection for Segment Routing (SR) networks", draft-hegde-rtgwg-egress-protection-sr-networks-01 (work in progress), November 2020.

[I-D.ietf-idr-performance-routing]
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-03 (work in progress), December 2020.

[I-D.kaliraj-idr-bgp-classful-transport-planes]
Vairavakkalai, K., Venkataraman, N., Rajagopalan, B., Mishra, G., Khaddam, M., and X. Xu, "BGP Classful Transport Planes", draft-kaliraj-idr-bgp-classful-transport-planes-06 (work in progress), January 2021.

[I-D.zzhang-bess-bgp-multicast]
Zhang, Z., Giuliano, L., Patel, K., Wijnands, I., mishra, m., and A. Gulko, "BGP Based Multicast", draft-zzhang-bess-bgp-multicast-03 (work in progress), October 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <https://www.rfc-editor.org/info/rfc3107>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

10.2. Informative References

[I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Node Protection for SR-TE Paths", draft-hegde-spring-node-protection-for-sr-te-paths-07 (work in progress), July 2020.

[I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-07 (work in progress), March 2018.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-11 (work in progress), November 2020.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-21 (work in progress), January 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-pce-segment-routing-policy-cp]
Koldychev, M., Sivabalan, S., Barth, C., Peng, S., and H. Bidgoli, "PCEP extension to support Segment Routing Policy Candidate Paths", draft-ietf-pce-segment-routing-policy-cp-02 (work in progress), January 2021.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.saad-sr-fa-link]
Saad, T., Beeram, V., Barth, C., and S. Sivabalan, "Segment-Routing over Forwarding Adjacency Links", draft-saad-sr-fa-link-02 (work in progress), July 2020.

[I-D.voyer-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (work in progress), July 2020.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

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

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <https://www.rfc-editor.org/info/rfc4684>.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <https://www.rfc-editor.org/info/rfc5065>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC7311] Mohapatra, P., Fernando, R., Rosen, E., and J. Uttaro, "The Accumulated IGP Metric Attribute for BGP", RFC 7311, DOI 10.17487/RFC7311, August 2014, <https://www.rfc-editor.org/info/rfc7311>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <https://www.rfc-editor.org/info/rfc7510>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8604] Filsfils, C., Ed., Previdi, S., Dawra, G., Ed., Henderickx, W., and D. Cooper, "Interconnecting Millions of Endpoints with Segment Routing", RFC 8604, DOI 10.17487/RFC8604, June 2019, <https://www.rfc-editor.org/info/rfc8604>.

[RFC8679] Shen, Y., Jeganathan, M., Decraene, B., Gredler, H., Michel, C., and H. Chen, "MPLS Egress Protection Framework", RFC 8679, DOI 10.17487/RFC8679, December 2019, <https://www.rfc-editor.org/info/rfc8679>.

[TS.23.501-3GPP]
3rd Generation Partnership Project (3GPP), "System Architecture for 5G System; Stage 2, 3GPP TS 23.501 v16.4.0", March 2020.

Authors' Addresses

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks Inc.

Email: cbowers@juniper.net

Xiaohu Xu
Alibaba Inc.
Beijing
China

Email: xiaohu.xxh@alibaba-inc.com

Arkadiy Gulko
EdwardJones

Email: arkadiy.gulko@edwardjones.com

Alex Bogdanov
Google Inc.

Email: bogdanov@google.com

James Uttaro
ATT

Email: ju1738@att.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Mazen Khaddam
Cox communications

Email: mazen.khaddam@cox.com

Andrew Alston
Liquid Telecom

Email: andrew.alston@liquidtelecom.com

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com/

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 24, 2021

Z. Hu
Huawei Technologies
H. Chen
Futurewei
J. Yao
Huawei Technologies
C. Bowers
Juniper Networks
Y. Zhu
China Telecom
Y. Liu
China Mobile

February 20, 2021

SR-TE Path Midpoint Protection
draft-hu-spring-segment-routing-proxy-forwarding-13

Abstract

Segment Routing Traffic Engineering (SR-TE) supports explicit paths using segment lists containing adjacency-SIDs, node-SIDs and binding-SIDs. The current SR FRR such as TI-LFA provides fast re-route protection for the failure of a node along a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node. This document describes a mechanism for fast re-route protection against the failure of a SR-TE path after the IGP converges. It provides fast re-route protection for an adjacency segment, a node segment and a binding segment of the path.

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/.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using a segment list. SR-TE supports the creation of explicit paths using adjacency-SIDs, node-SIDs, anycast-SIDs, and binding-SIDs. A node-SID in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When the node fails, the network may no longer be able to properly forward traffic on that SR-TE path.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes an SR FRR mechanism that provides fast re-route protection for the failure of a node on a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node and drop the traffic.

To solve this problem, [I-D.ietf-spring-segment-protection-sr-te-paths] proposes that a hold timer should be configured on every router in a network. After the IGP converges on the event of a node failure, if the node-SID of the failed node becomes unreachable, the forwarding changes should not be communicated to the forwarding planes on all configured routers (including PLRs for the failed node) until the hold timer expires. This solution may not work for some cases such as some of nodes in the network not supporting this solution.

This document describes a proxy protection/forwarding mechanism, which provides more protection coverages. It considers the fast re-route protection capability of every node in the network and supports the fast re-route protection of the binding-SIDs on a failed node.

2. Proxy Forwarding

In the proxy forwarding mechanism, each neighbor of a possible failed node advertises its SR proxy forwarding capability in its network domain when it has the capability. This capability indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from the neighbors of a failed node will send traffic using the node-SID of the failed node to the nearest Proxy Forwarder after the IGP converges on the event of the failure.

Once the affected traffic reaches a Proxy Forwarder, it sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list.

For a binding segment of a possible failed node, the node advertises the information about the binding segment, including the binding SID and the list of SIDs associated with the binding SID, to its direct neighbors only. Note that the information is not advertised in the network domain.

After the node fails and the IGP converges on the failure, the traffic with the binding SID of the failed node will reach its neighbor having SR Proxy Forwarding capability. Once receiving the traffic, the neighbor swaps the binding SID with the list of SIDs associated with the binding SID and sends the traffic along the post-failure shortest path to the first node in the segment list.

3. Extensions to IGP for Proxy Forwarding

This section defines extensions to IGP for advertising the SR proxy forwarding capability of a node in a network domain and the information about each binding segment (including its binding SID and the list of SIDs associated) of a node to its direct neighbors.

3.1. Extensions to OSPF

3.1.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for all its neighboring nodes for protecting the failures of these nodes, node P advertises its SR proxy forwarding capability in its router information opaque LSA, which contains a Router Functional Capabilities TLV of the format as shown in Figure 1.

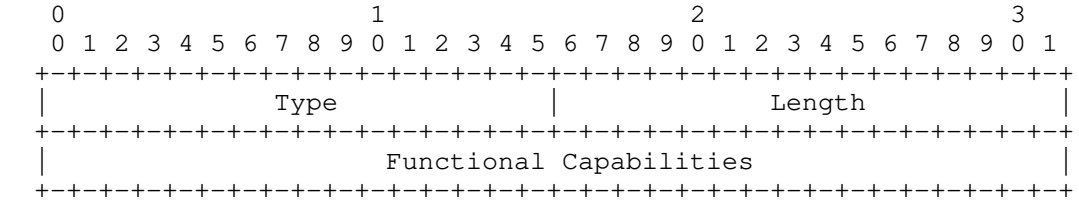


Figure 1: Router Functional Capabilities TLV

One bit (called PF bit) in the Functional Capabilities field of the TLV is used to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/ node SID of node N. Thus node X will forward the traffic targeting to the prefix/node SID of node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forward the traffic towards its final destination without going through node N. After node N fails, node X will keep the FIB entry to the proxy prefix/node SID of node N for a given period of time.

Note that the behaviors of normal IP forwarding and routing convergences in a network are not changed at all by the SR proxy forwarding. For example, the next hop used by BGP is an IP address (or prefix). The IGP and BGP converge in normal ways for changes in the network. The packet with its IP destination to this next hop is forwarded according to the IP forwarding table (FIB) derived from IGP and BGP routes.

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

A new TLV, called Proxy Node SIDs TLV, is defined for node P to advertise the node SIDs of some of its neighboring nodes. It has the format as shown in Figure 2.

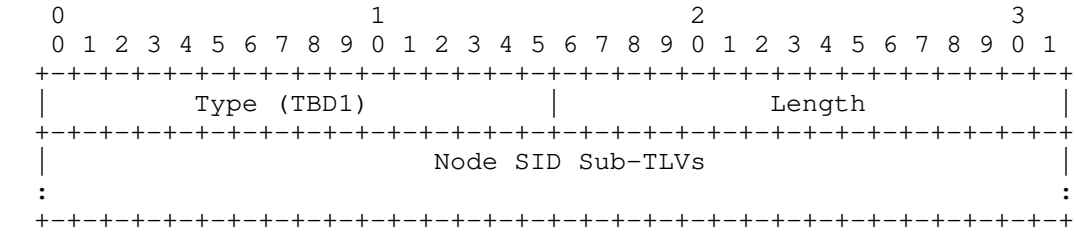


Figure 2: OSPF Proxy Node SIDs TLV

The Type (TBD1) is to be assigned by IANA. The TLV contains a number of Node SID Sub-TLVs. The Length is the total size of the Node SID Sub-TLVs included in the TLV. A Node SID Sub-TLV is the Prefix SID Sub-TLV defined in [RFC8665].

A proxy forwarding node P originates an Extended Prefix Opaque LSA containing this new TLV. The TLV includes the Node SID Sub-TLVs for the node SIDs of some of P's neighboring nodes. For each of some of P's neighboring nodes, the Node SID Sub-TLV for its prefix/node SID is included the TLV. This prefix/node SID is called a proxy prefix/ node SID.

A proxy forwarding node will originate an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The format of the LSA is shown in Figure 3.

For a proxy forwarding node P, having a number of neighboring nodes, P originates and maintains an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The TLV contains the Prefix/Node SID Sub-TLV for each of some of the neighboring nodes after node P creates the corresponding proxy forwarding entries for protecting the failure of some of the neighboring nodes.

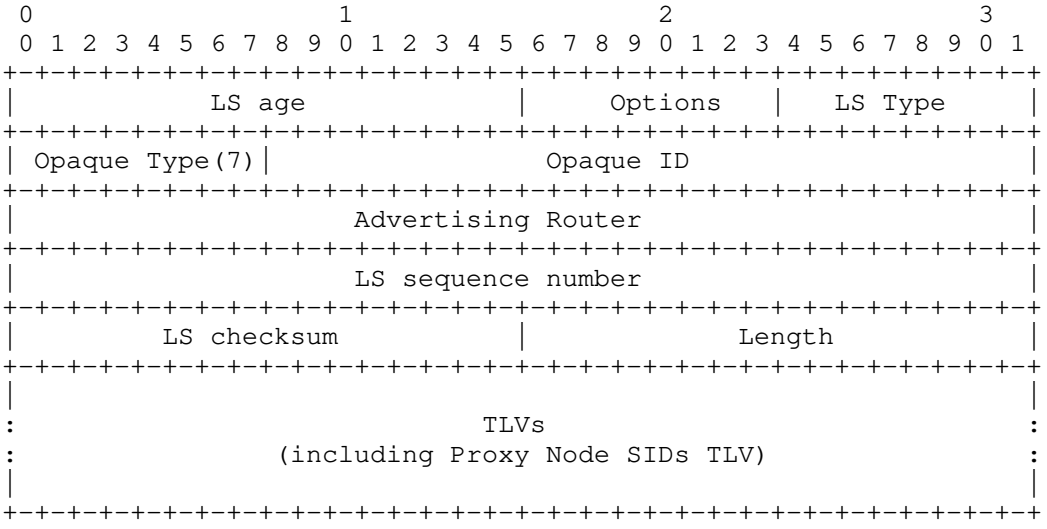


Figure 3: OSPFv2 Extended Prefix Opaque LSA

When an neighboring node fails, P maintains the LSA with the TLV containing the Prefix/Node SID Sub-TLV for the neighboring node for a given period of time. After the given period of time, the Prefix/ Node SID Sub-TLV for the neighboring node is removed from the TLV in the LSA and then after a given time the corresponding proxy forwarding entries for protecting the failure of the neighboring node is removed.

For a node X in the network, it learns the prefix/node SID of node N and the proxy prefix/node SID of node N. The former is originated and advertised by node N, and the latter is originated and advertised by the proxy forwarding node P of node N. Note that the proxy Prefix/Node SID Sub-TLV for node N does not contain a prefix of node N, and the prefix is the prefix associated with the prefix/node SID of node N originated by node N.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/ node SID of node N. Thus node X will forward the traffic targeting to node N to node P when node N fails, and node P will do a proxy forwarding for node N and forward the traffic towards its destination without going through node N.

3.1.2. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of segments, node A advertises an LSA containing the binding (i.e., the binding SID and the list of the segments). The LSA is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LSA is a opaque LSA of LS type 9 (i.e., a link local scope LSA).

A binding segment is represented by binding segment TLV of the format as shown in Figure 4.

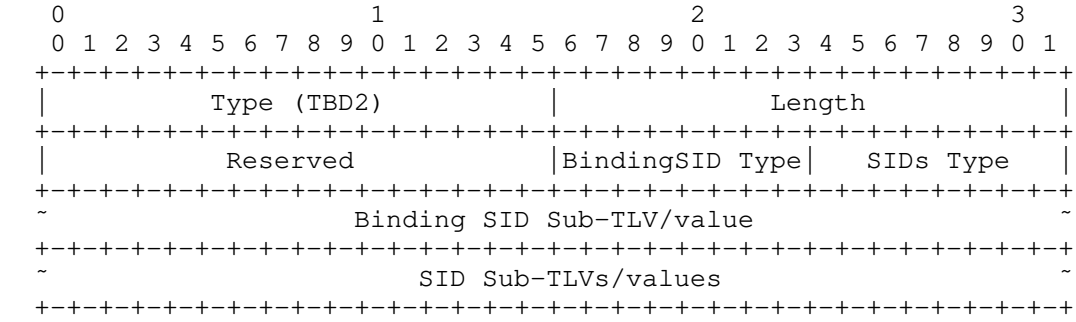


Figure 4: OSPF Binding Segment TLV

It comprises a binding SID and a list of segments (SIDs). The fields of this TLV are defined as follows:

Type: 2 octets, its value (TBD2) is to be assigned by IANA.

Length: 2 octets, its value is (4 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

o BT = 0: The binding SID is represented by a Sub-TLV (i.e., Binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [RFC8665]. BT != 0 indicates that the binding SID is represented by a value.

o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

SIDs Type (ST): 1 octet indicates whether the list of segments (SIDs) are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [RFC8665]. ST != 0 indicates that the SIDs are represented by values.

o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

The opaque LSA of LS Type 9 containing the binding segment (i.e., the binding SID and the list of the segments) has the format as shown in Figure 5. It may have Opaque Type of x (the exact type is to be assigned by IANA) for Binding Segment Opaque LSA.

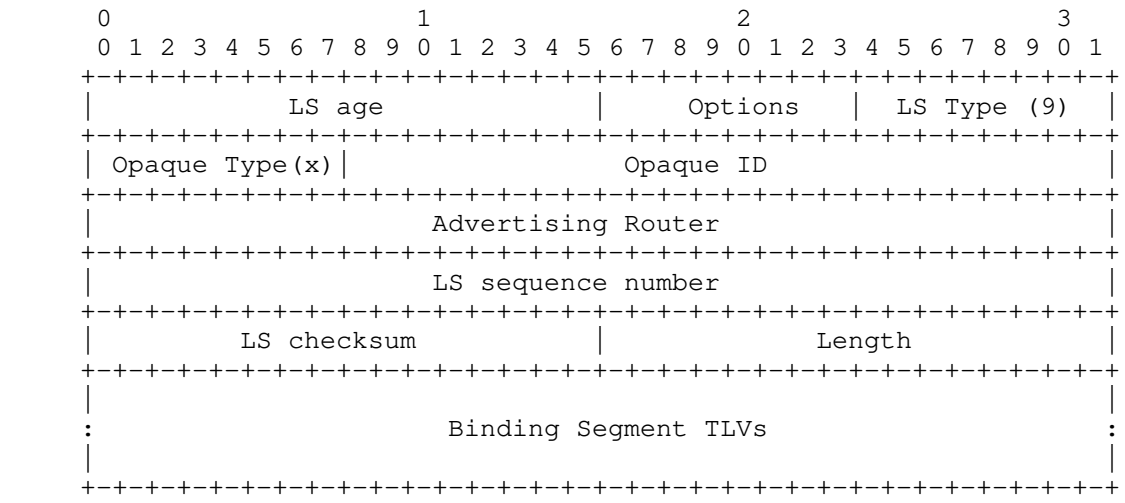


Figure 5: OSPFv2 Binding Segment Opaque LSA

For every binding on a node A, the LSA originated by A contains a binding segment TLV for it.

For node A running OSPFv3, it originates a link-local scoping LSA of a new LSA function code (TBD3) containing binding segment TLVs for the bindings on it. The format of the LSA is illustrated in Figure 6.

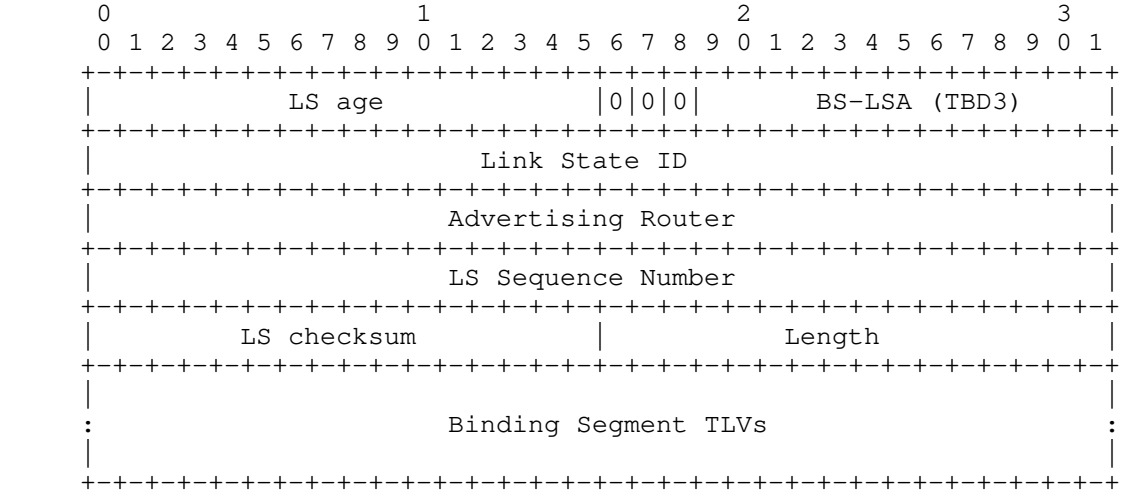


Figure 6: OSPFv3 Binding Segment Opaque LSA

The U-bit is set to 0, and the scope is set to 00 for link-local scoping.

3.2. Extensions to IS-IS

3.2.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for its neighboring nodes for protecting the failures of them, node P advertises its SR proxy forwarding capability in its LSP, which contains a Router Capability TLV of Type 242 including a SR capabilities sub-TLV of sub-Type 2.

One bit (called PF bit as shown in Figure 7) in the Flags field of the SR capabilities sub-TLV is defined to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

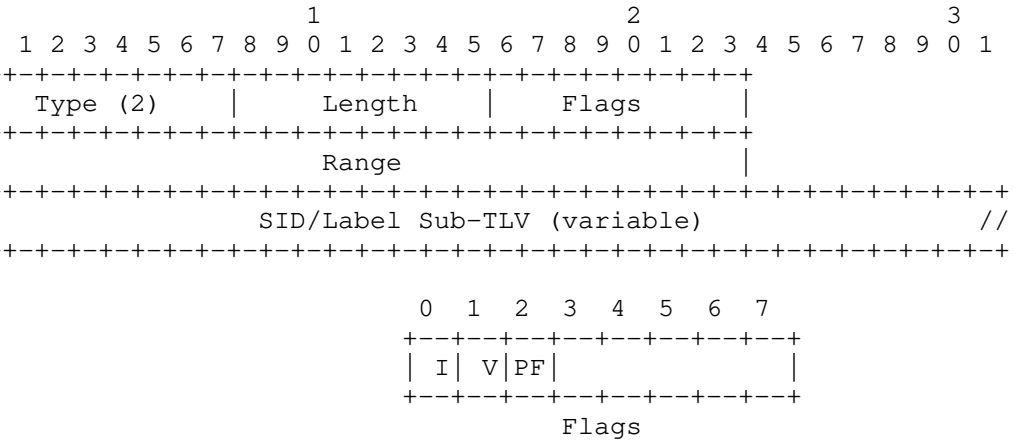


Figure 7: SR Capabilities sub-TLV

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

The IS-IS SID/Label Binding TLV (suggested value 149) is defined in [RFC8667]. A Proxy Forwarder uses the SID/Label Binding TLV to advertise the node SID of its neighboring node. The Flags field of the SID/Label Binding TLV is extended to include a P flag as shown in Figure 8. The prefix/node SID in prefix/node SID Sub-TLV included in

SID/Label Binding TLV is identified as a proxy forwarding prefix/node SID.

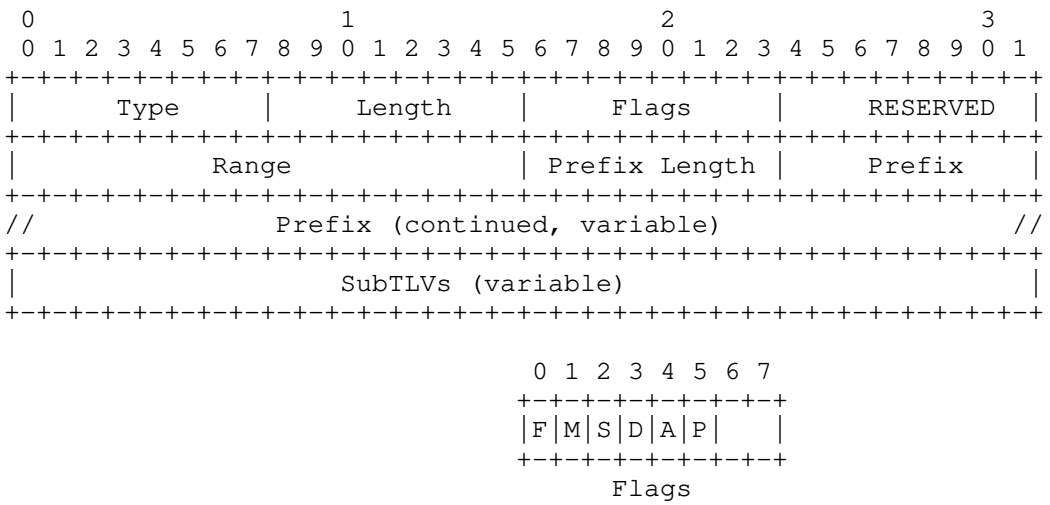


Figure 8: SID/Label Binding TLV

Where:

P-Flag: Proxy forwarding flag. If set, this prefix/node SID is advertised by the proxy node. This TLV is used to announce that the node has the ability to proxy forward the prefix/node SID.

When the P-flag is set in the SID/Label Binding TLV, the following usage rules apply.

The Range, Prefix Length and Prefix field are not used. They should be set to zero on transmission and ignored on receipt.

SID/Label Binding TLV contains a number of prefix/node SID Sub-TLVs. The TLV advertised by a proxy forwarding node P contains prefix/node SID Sub-TLVs for the node SIDs of P's neighbor nodes. Each of the Sub-TLVs is a prefix/node SID Sub-TLV defined in [RFC8667]. From the SID in a prefix/node SID Sub-TLV advertised by the Proxy Forwarding node, its prefix can be obtained through matching corresponding prefix/node SID advertised by the neighbor/protected node using TLV-135 (or 235, 236, or 237) together with the prefix/node SID Sub-TLV.

3.2.2. Advertising Binding Segment

[I-D.ietf-spring-segment-routing-policy] has defined the usage of binding-SID. For supporting binding SID proxy forwarding, a new IS-IS TLV, called Binding Segment TLV, is defined. It contains a binding SID and a list of segments (SIDs). This TLV may be advertised in IS-IS Hello (IIH) PDUs, LSPs, or in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. Its format is shown in Figure 9.

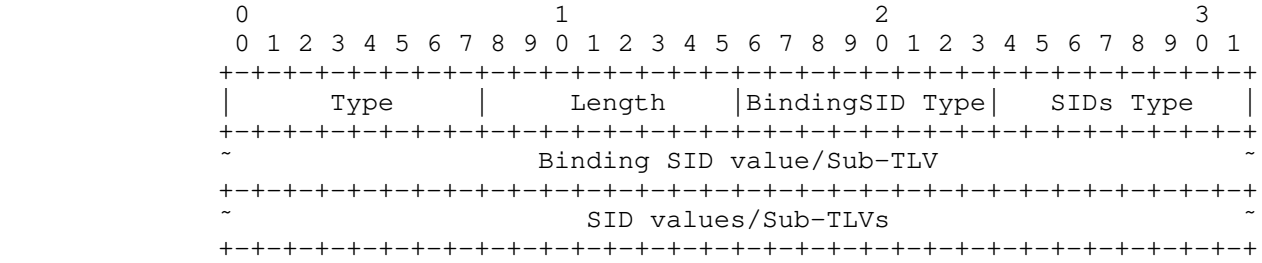


Figure 9: IS-IS Binding Segment TLV

The fields of this TLV are defined as follows:

Type: 1 octet Suggested value 152 (to be assigned by IANA)

Length: 1 octet (2 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

o BT = 0: The binding SID is represented by a Sub-TLV (i.e., binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [RFC8667]. BT != 0 indicates that the binding SID is represented by a value.

o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

SIDs Type (ST): 1 octet indicates whether the SIDs are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

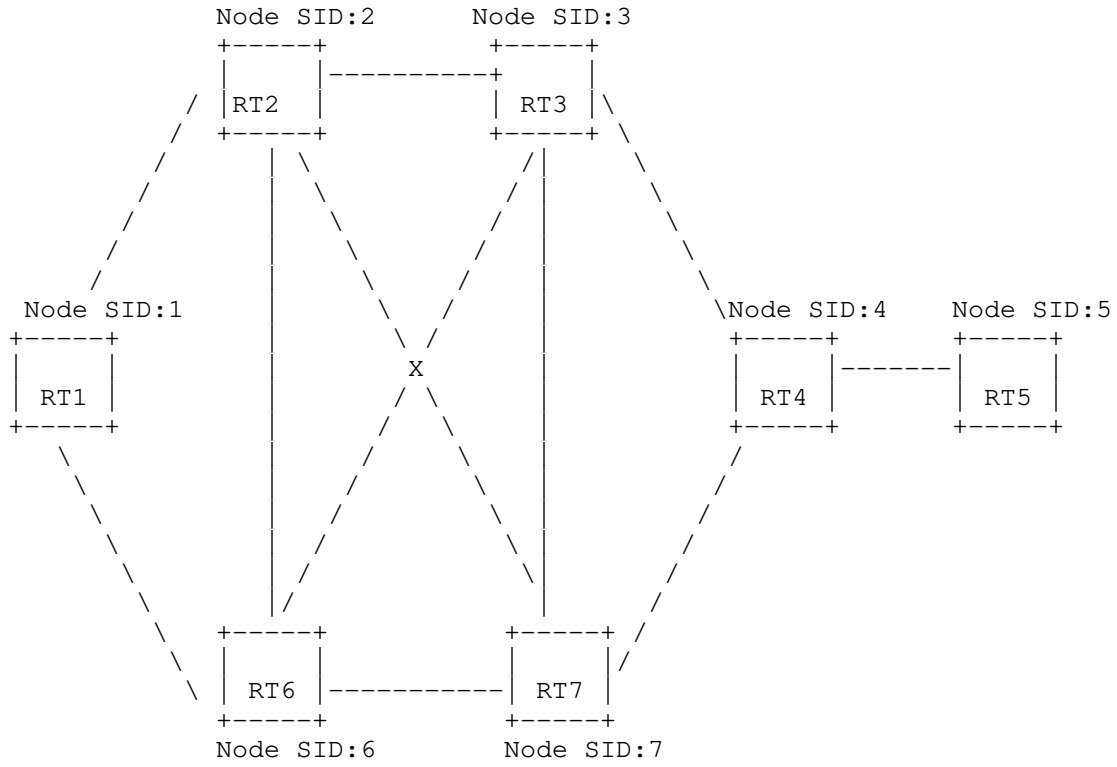
o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [RFC8667]. ST != 0 indicates that the SIDs are represented by values.

o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

4. Building Proxy Forwarding Table

Figure 10 is used to illustrate the SR proxy forwarding approach. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Three label stacks are shown in the figure. Label Stack 1 uses only adjacency-SIDs and represents the path RT1->RT2->RT3->RT4->RT5. Label Stack 2 uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5. Label Stack 3 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 3, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.



Node SRGB	Adj-SID	Label Stack 1	Label Stack 2	Label Stack 3
RT1:[1000-1999]	RT1->RT2:10012	10012	1003	1003
RT2:[2000-2999]	RT2->RT3:20023	20023	3004	100
RT3:[3000-3999]	RT3->RT6:30036	30034	4005	100 is binding SID to {30034,40045}
RT4:[4000=4999]	RT3->RT7:30037	40045		
RT5:[5000-5999]	RT3->RT4:30034			
RT6:[6000-6999]	RT7->RT4:70074			
RT7:[7000-7999]	RT4->RT5:40045			

Figure 10: Topology of SR-TE Path

4.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do a SR proxy forwarding for all its neighboring nodes, it must advertise this capability. If the PLR can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, for example, RT3, then it uses proxy Node SIDs TLV to advertise the prefix-SID learned from RT3. The TLV contains the Sub-TLV/value for the prefix/node SID of RT3 as a proxy SID. When RT3 fails, RT2 needs to maintain the Sub-TLV/value for a period of time. When the proxy forwarding table corresponding to the fault node is deleted (see section 3.2), the Sub-TLV/value is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node SID TLV advertised by RT3 and the proxy Node SIDs TLV advertised by RT2. When RT3 is normal, the nodes prefer prefix/node SID TLV. When the RT3 fails, the proxy prefix/node SIDs TLV advertised by RT2 is preferred.

4.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;

2: All adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.

3: The binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adj-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in the topology of Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. RT2 calculates the proxy forwarding table for RT3, as shown in Figure 11.

In-label	SRGBDiffValue	Next Label	Action	Map Label
2003	-1000	30034	Fwd to RT4	2004
		30036	Fwd to RT6	2006
		30037	Fwd to RT7	2007
		100	Swap to { 30034, 40045 }	

Figure 11: RT2's Proxy Forwarding Table for RT3

5. Node Protection for Segment List

Segment Routing Traffic Engineering supports the creation of explicit paths using adjacency-SIDs, node-SIDs, and binding-SIDs. The label stack is a combination of one or more of adjacency-SIDs, node-SIDs, and binding-SIDs. This Section shows how a proxy node uses the SR proxy forwarding mechanism to protect traffic to the destination node when the next segment of label stack is adjacency-SIDs, node-SIDs, or binding-SIDs, respectively.

5.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 1 {10012, 20023, 30034, 40045} represents SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adj-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

a. RT1 pops top adj-SID 10012, and forwards the packet to RT2;

b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The query result is 2004. RT2 uses 2004 as the incoming label to query the label forwarding table. The next hop is RT7, and the incoming label is changed to 7004.

c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT4 forwards it to RT4, where the original path is rejoined.

d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

5.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} represents SR-TE loose path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

a. RT1 swaps label 1003 to out-label 2003 to RT3.

b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure, RT2 pops label 2003.

c. RT2 uses 3004 as the in-label to lookup Proxy Forwarding table, The value of Map Label calculated based on SRGBDiffValue is 2004. and the query result is forwarding the packet to RT4.

d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.

e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.

f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

5.3. Next Segment is a Binding Segment

As shown in Figure 1, Label Stack 3 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-SID, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

a. RT1 swaps label 1003 to out-label 2003 to RT3.

b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.

c. RT2 uses Binding-SID:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.

d. RT2 swaps Binding-SID:100 to Segment list {30034, 40045}, and uses the 3034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.

e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.

f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

6. Security Considerations

The extensions to OSPF and IS-IS described in this document result in two types of behaviors in data plane when a node in a network fails. One is that for a node, which is a upstream (except for the direct upstream) node of the failed node along a SR-TE path, it continues to send the traffic to the failed node along the SR-TE path for an extended period of time. The other is that for a node, which is the direct upstream node of the failed node, it fast re-routes the traffic around the failed node to the direct downstream node of the failed node along the SR-TE path. These behaviors are internal to a network and should not cause extra security issues.

7. IANA Considerations

7.1. OSPFv2

Under Subregistry Name "OSPF Router Functional Capability Bits" within the "Open Shortest Path First v2 (OSPFv2) Parameters" [RFC7770], IANA is requested to assign one bit for Proxy Forwarding Capability as follows:

Bit number	Capability Name	Reference
31	Proxy Forwarding	This document

Under Registry Name "OSPFv2 Extended Prefix Opaque LSA TLVs" [RFC7684], IANA is requested to assign one new TLV value for OSPF Proxy Node SIDs as follows:

TLV Value	TLV Name	Reference
2	Proxy Node SIDs TLV	This document

Under Registry Name "Opaque Link-State Advertisements (LSA) Option Types" [RFC5250], IANA is requested to assign new Opaque Type registry values for Binding Segment LSA as follows:

Registry Value	Opaque Type	Reference
10	Binding Segment	This document

IANA is requested to create and maintain new registries:

- OSPFv2 Binding Segment Opaque LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	TLV Name	Definition
0	Reserved	
1	Binding Segment TLV	This Document
2-32767	Unassigned	
32768-65535	Reserved	

7.2. OSPFv3

Under Registry Name "OSPFv3 LSA Function Codes", IANA is requested to assign new registry values for Binding Segment LSA as follows:

Value	LSA Function Code Name	Reference
16	Binding Segment LSA	This document

IANA is requested to create and maintain new registries:

- OSPFv3 Binding Segment LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	TLV Name	Definition
0	Reserved	
1	Binding Segment TLV	This Document
2-32767	Unassigned	
32768-65535	Reserved	

7.3. IS-IS

Under Registration "Segment Routing Capability" in the "sub-TLVs for TLV 242" registry [RFC8667], IANA is requested to assign one bit flag for Proxy Forwarding Capability as follows:

Bit number	Capability Name	Reference
2	Proxy Forwarding (PF)	This document

Under Registration "Segment Identifier/Label Binding TLV 149" [RFC8667], IANA is requested to assign one bit P-Flag as follows:

Bit number	Flag Name	Reference
5	P-Flag	This document

Under Registry Name: IS-IS TLV Codepoints, IANA is requested to assign one new TLV value for IS-IS Binding Segment as follows:

Value	TLV Name	Reference
152	Binding Segment TLV	This Document

8. Acknowledgements

The authors would like to thank Peter Psenak, Acee Lindem, Les Ginsberg, Bruno Decraene and Jeff Tantsura for their comments to this work.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <https://www.rfc-editor.org/info/rfc7356>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

9.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", draft-ietf-spring-segment-protection-sr-te-paths-00 (work in progress), September 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-07 (work in progress), July 2019.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: huzhibo@huawei.com

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Junda Yao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: yaojunda@huawei.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: cbowers@juniper.net

Yongqing
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou 510000
China

Email: zhuyq8@chinatelecom.cn

Yisong
China Mobile
510000
China

Email: liuyisong@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 31, 2021

Z. Hu
Huawei Technologies
H. Chen
Futurewei
J. Yao
Huawei Technologies
C. Bowers
Juniper Networks
Y. Zhu
China Telecom
Y. Liu
China Mobile

April 29, 2021

SR-TE Path Midpoint Restoration
draft-hu-spring-segment-routing-proxy-forwarding-14

Abstract

Segment Routing Traffic Engineering (SR-TE) supports explicit paths using segment lists containing adjacency-SIDs, node-SIDs and binding-SIDs. The current SR FRR such as TI-LFA provides fast re-route protection for the failure of a node along a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node. This document describes a mechanism for the restoration of the routes to the failure of a SR-TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment of the path. With the restoration of the routes to the failure, the traffic is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

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.

Copyright Notice

Table of Contents

1. Introduction

Segment Routing Traffic Engineering (SR-TE) is a technology that implements traffic engineering using a segment list. SR-TE supports the creation of explicit paths using adjacency-SIDs, node-SIDs, anycast-SIDs, and binding-SIDs. A node-SID in the segment list defining an SR-TE path indicates a loose hop that the SR-TE path should pass through. When the node fails, the network may no longer be able to properly forward traffic on that SR-TE path.

[I-D.ietf-rtgwg-segment-routing-ti-lfa] describes an SR FRR mechanism that provides fast re-route protection for the failure of a node on a SR-TE path by the direct neighbor or say point of local repair (PLR) to the failure. However, once the IGP converges, the SR FRR is no longer sufficient to forward traffic of the path around the failure, since the non-neighbors of the failure will no longer have a route to the failed node and drop the traffic.

To solve this problem, [I-D.ietf-spring-segment-protection-sr-te-paths] proposes that a hold timer should be configured on every router in a network. After the IGP converges on the event of a node failure, if the node-SID of the failed node becomes unreachable, the forwarding changes should not be communicated to the forwarding planes on all configured routers (including PLRs for the failed node) until the hold timer expires. This solution may not work for some cases such as some of nodes in the network not supporting this solution.

This document describes a proxy forwarding mechanism for the restoration of the routes to the failure of a SR-TE path after the IGP converges. It provides the restoration of the routes to an adjacency segment, a node segment and a binding segment on a failed node along the SR-TE path. With the restoration of the routes to the failure, the traffic for the SR-TE path is continuously sent to the neighbor of the failure after the IGP converges. The neighbor as a PLR fast re-routes the traffic around the failure.

2. Proxy Forwarding

In the proxy forwarding mechanism, each neighbor of a possible failed node advertises its SR proxy forwarding capability in its network domain when it has the capability. This capability indicates that the neighbor (the Proxy Forwarder) will forward traffic on behalf of the failed node. A router receiving the SR Proxy Forwarding capability from the neighbors of a failed node will send traffic using the node-SID of the failed node to the nearest Proxy Forwarder after the IGP converges on the event of the failure.

Once the affected traffic reaches a Proxy Forwarder, it sends the traffic on the post-failure shortest path to the node immediately following the failed node in the segment list.

For a binding segment of a possible failed node, the node advertises the information about the binding segment, including the binding SID and the list of SIDs associated with the binding SID, to its direct neighbors only. Note that the information is not advertised in the network domain.

After the node fails and the IGP converges on the failure, the traffic with the binding SID of the failed node will reach its neighbor having SR Proxy Forwarding capability. Once receiving the traffic, the neighbor swaps the binding SID with the list of SIDs associated with the binding SID and sends the traffic along the post-failure shortest path to the first node in the segment list.

3. Extensions to IGP for Proxy Forwarding

This section defines extensions to IGP for advertising the SR proxy forwarding capability of a node in a network domain and the information about each binding segment (including its binding SID and the list of SIDs associated) of a node to its direct neighbors.

3.1. Extensions to OSPF

3.1.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for all its neighboring nodes for protecting the failures of these nodes, node P advertises its SR proxy forwarding capability in its router information opaque LSA, which contains a Router Functional Capabilities TLV of the format as shown in Figure 1.

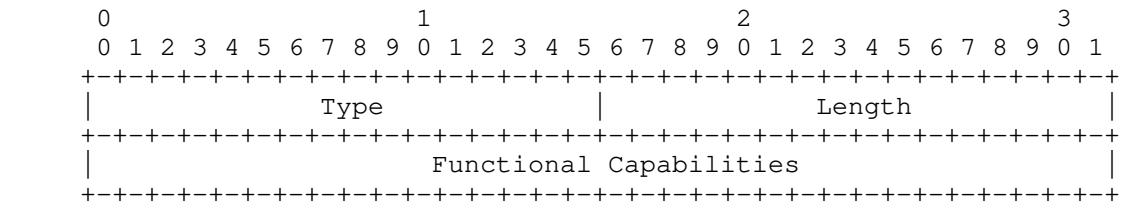


Figure 1: Router Functional Capabilities TLV

One bit (called PF bit) in the Functional Capabilities field of the TLV is used to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

For a node X in the network, it learns the prefix/node SID of node N, which is originated and advertised by node N. It creates a proxy prefix/node SID of node N for node P if node P is capable of doing SR proxy forwarding for node N. The proxy prefix/node SID of node N for node P is a copy of the prefix/node SID of node N originated by node N, but stored under (or say, associated with) node P. The route to the proxy prefix/node SID is through proxy forwarding capable nodes.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/ node SID of node N. Thus node X will forward the traffic targeting to the prefix/node SID of node N to node P when node N fails, and node P will do a SR proxy forwarding for node N and forward the traffic towards its final destination without going through node N. After node N fails, node X will keep the FIB entry to the proxy prefix/node SID of node N for a given period of time.

Note that the behaviors of normal IP forwarding and routing convergences in a network are not changed at all by the SR proxy forwarding. For example, the next hop used by BGP is an IP address (or prefix). The IGP and BGP converge in normal ways for changes in the network. The packet with its IP destination to this next hop is forwarded according to the IP forwarding table (FIB) derived from IGP and BGP routes.

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

A new TLV, called Proxy Node SIDs TLV, is defined for node P to advertise the node SIDs of some of its neighboring nodes. It has the format as shown in Figure 2.

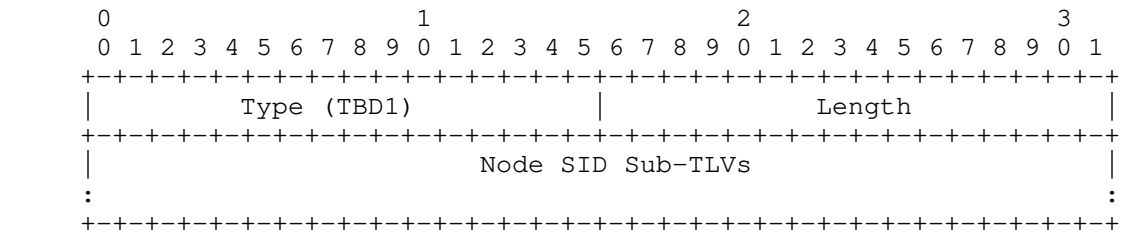


Figure 2: OSPF Proxy Node SIDs TLV

The Type (TBD1) is to be assigned by IANA. The TLV contains a number of Node SID Sub-TLVs. The Length is the total size of the Node SID Sub-TLVs included in the TLV. A Node SID Sub-TLV is the Prefix SID Sub-TLV defined in [RFC8665].

A proxy forwarding node P originates an Extended Prefix Opaque LSA containing this new TLV. The TLV includes the Node SID Sub-TLVs for the node SIDs of some of P's neighboring nodes. For each of some of P's neighboring nodes, the Node SID Sub-TLV for its prefix/node SID is included the TLV. This prefix/node SID is called a proxy prefix/ node SID.

A proxy forwarding node will originate an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The format of the LSA is shown in Figure 3.

For a proxy forwarding node P, having a number of neighboring nodes, P originates and maintains an Extended Prefix Opaque LSA, which includes a Proxy Node SIDs TLV. The TLV contains the Prefix/Node SID Sub-TLV for each of some of the neighboring nodes after node P creates the corresponding proxy forwarding entries for protecting the failure of some of the neighboring nodes.

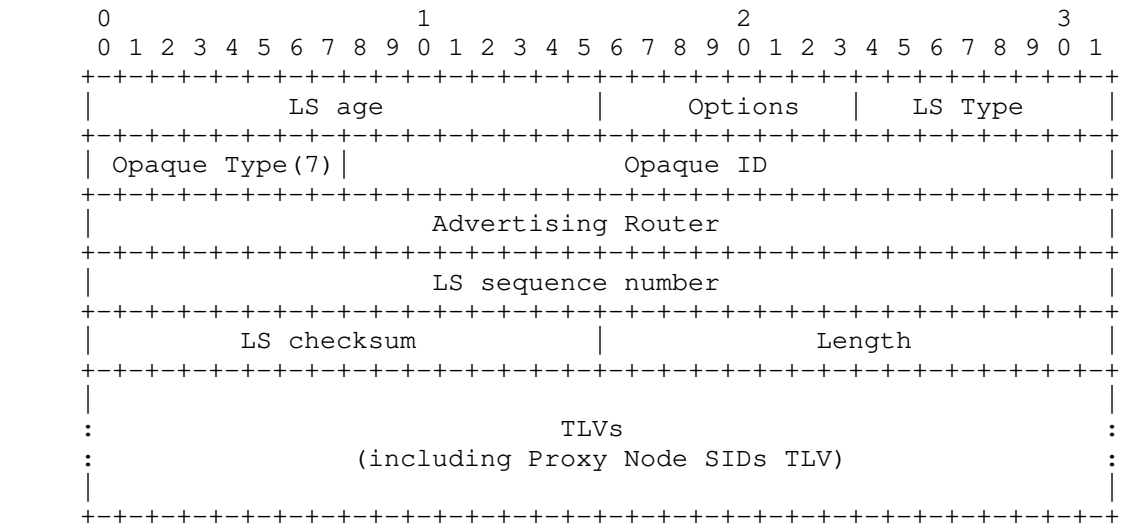


Figure 3: OSPFv2 Extended Prefix Opaque LSA

When an neighboring node fails, P maintains the LSA with the TLV containing the Prefix/Node SID Sub-TLV for the neighboring node for a given period of time. After the given period of time, the Prefix/ Node SID Sub-TLV for the neighboring node is removed from the TLV in the LSA and then after a given time the corresponding proxy forwarding entries for protecting the failure of the neighboring node is removed.

For a node X in the network, it learns the prefix/node SID of node N and the proxy prefix/node SID of node N. The former is originated and advertised by node N, and the latter is originated and advertised by the proxy forwarding node P of node N. Note that the proxy Prefix/Node SID Sub-TLV for node N does not contain a prefix of node N, and the prefix is the prefix associated with the prefix/node SID of node N originated by node N.

In normal operations, node X prefers to use the prefix/node SID of node N. When node N fails, node X prefers to use the proxy prefix/ node SID of node N. Thus node X will forward the traffic targeting to node N to node P when node N fails, and node P will do a proxy forwarding for node N and forward the traffic towards its destination without going through node N.

3.1.2. Advertising Binding Segment

For a binding segment (or binding for short) on a node A, which consists of a binding SID and a list of segments, node A advertises an LSA containing the binding (i.e., the binding SID and the list of the segments). The LSA is advertised only to each of the node A's neighboring nodes. For OSPFv2, the LSA is a opaque LSA of LS type 9 (i.e., a link local scope LSA).

A binding segment is represented by binding segment TLV of the format as shown in Figure 4.

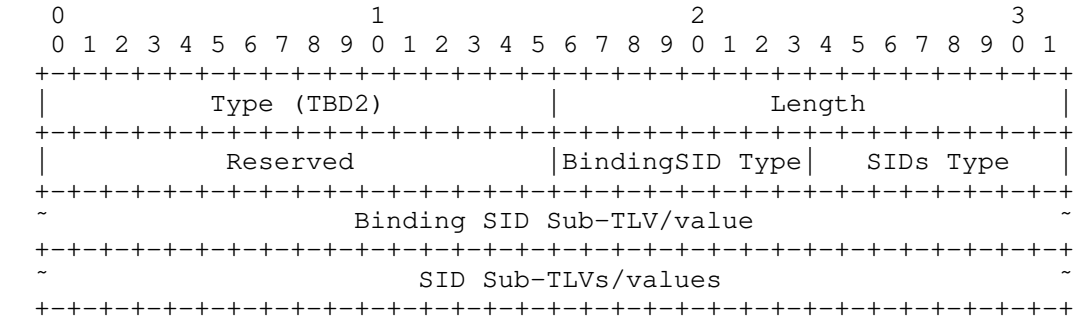


Figure 4: OSPF Binding Segment TLV

It comprises a binding SID and a list of segments (SIDs). The fields of this TLV are defined as follows:

Type: 2 octets, its value (TBD2) is to be assigned by IANA.

Length: 2 octets, its value is (4 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

o BT = 0: The binding SID is represented by a Sub-TLV (i.e., Binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [RFC8665]. BT != 0 indicates that the binding SID is represented by a value.

o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

SIDs Type (ST): 1 octet indicates whether the list of segments (SIDs) are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [RFC8665]. ST != 0 indicates that the SIDs are represented by values.

o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

The opaque LSA of LS Type 9 containing the binding segment (i.e., the binding SID and the list of the segments) has the format as shown in Figure 5. It may have Opaque Type of x (the exact type is to be assigned by IANA) for Binding Segment Opaque LSA.

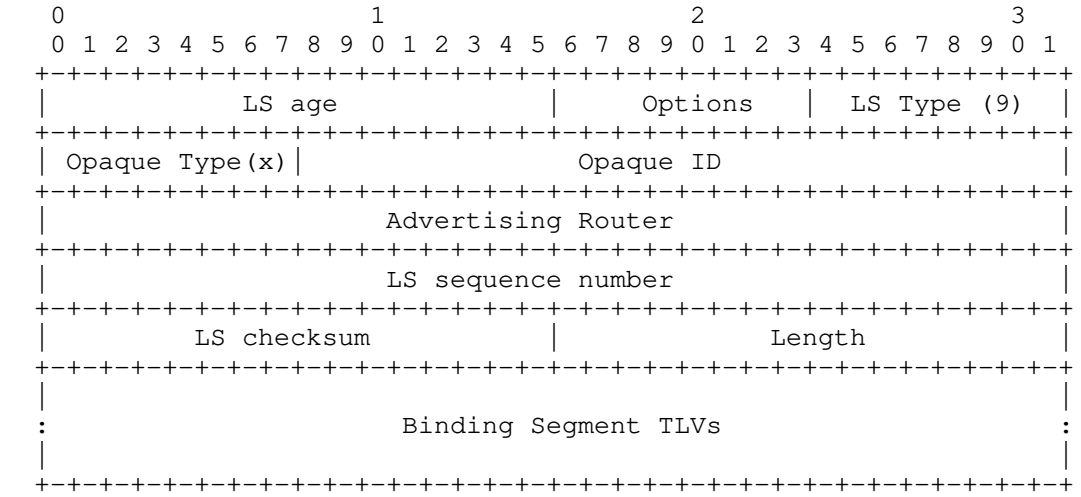


Figure 5: OSPFv2 Binding Segment Opaque LSA

For every binding on a node A, the LSA originated by A contains a binding segment TLV for it.

For node A running OSPFv3, it originates a link-local scoping LSA of a new LSA function code (TBD3) containing binding segment TLVs for the bindings on it. The format of the LSA is illustrated in Figure 6.

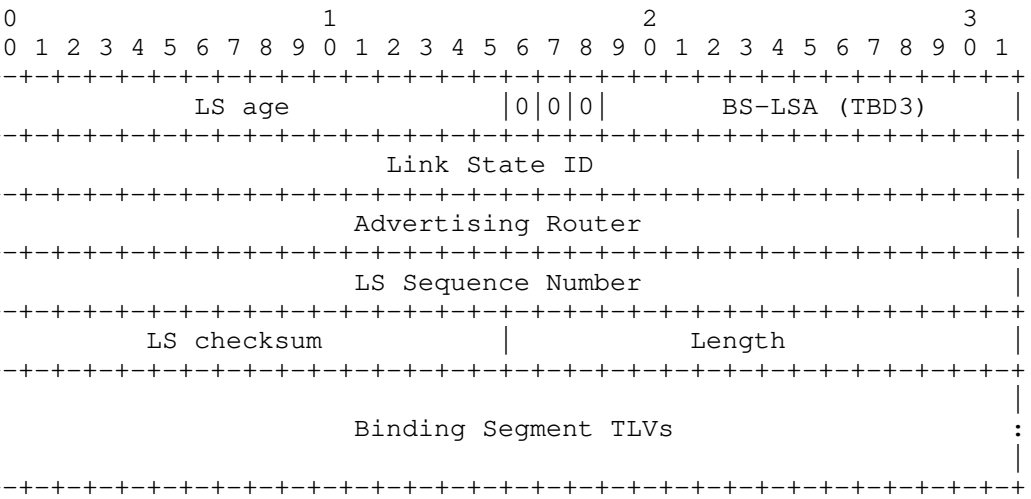


Figure 6: OSPFv3 Binding Segment Opaque LSA

The U-bit is set to 0, and the scope is set to 00 for link-local scoping.

3.2. Extensions to IS-IS

3.2.1. Advertising Proxy Forwarding

When a node P has the capability to do a SR proxy forwarding for its neighboring nodes for protecting the failures of them, node P advertises its SR proxy forwarding capability in its LSP, which contains a Router Capability TLV of Type 242 including a SR capabilities sub-TLV of sub-Type 2.

One bit (called PF bit as shown in Figure 7) in the Flags field of the SR capabilities sub-TLV is defined to indicate node P's SR proxy forwarding capability. When this bit is set to one by node P, it indicates that node P is capable of doing a SR proxy forwarding for its neighboring nodes.

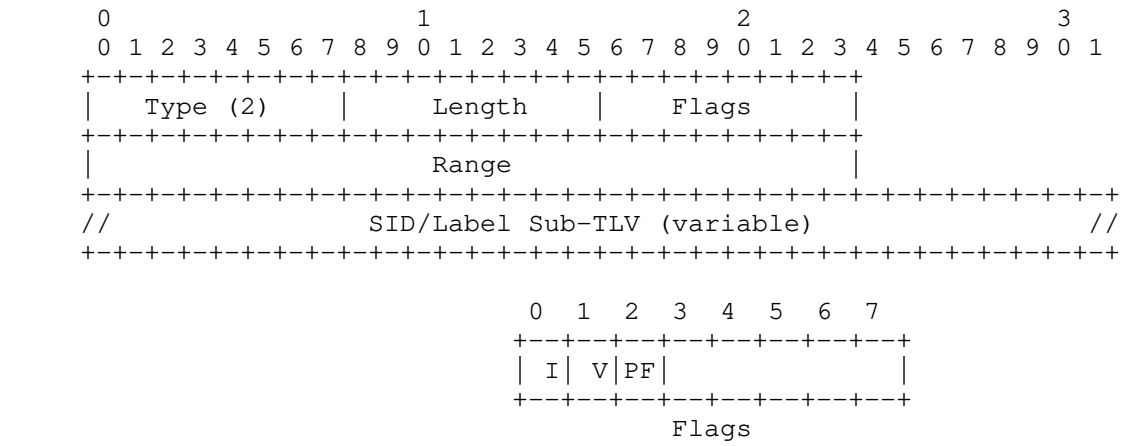


Figure 7: SR Capabilities sub-TLV

If node P can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, then it advertises the node SID of each of the nodes as a proxy node SID, indicating that it is able to do proxy forwarding for the node SID.

The IS-IS SID/Label Binding TLV (suggested value 149) is defined in [RFC8667]. A Proxy Forwarder uses the SID/Label Binding TLV to advertise the node SID of its neighboring node. The Flags field of the SID/Label Binding TLV is extended to include a P flag as shown in Figure 8. The prefix/node SID in prefix/node SID Sub-TLV included in SID/Label Binding TLV is identified as a proxy forwarding prefix/node SID.

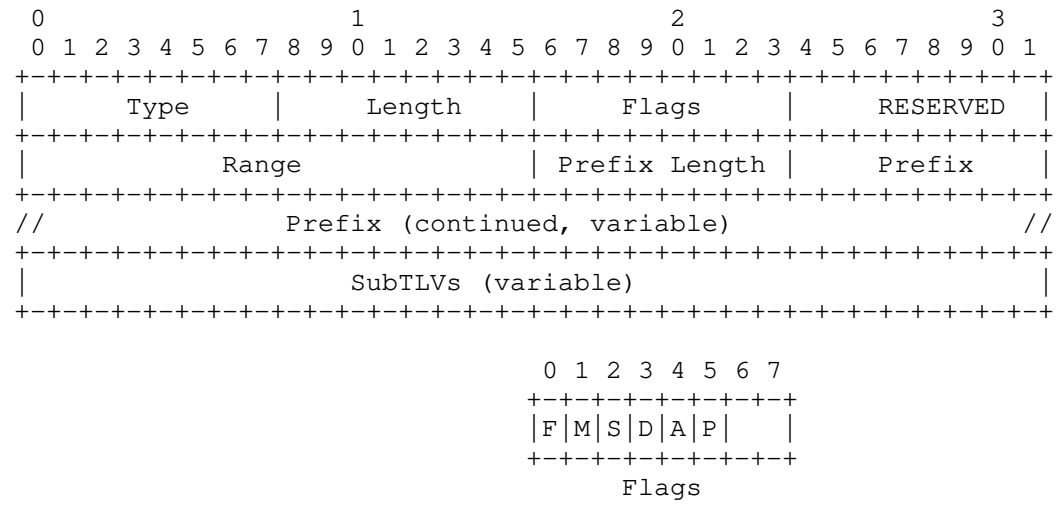


Figure 8: SID/Label Binding TLV

Where:

P-Flag: Proxy forwarding flag. If set, this prefix/node SID is advertised by the proxy node. This TLV is used to announce that the node has the ability to proxy forward the prefix/node SID.

When the P-flag is set in the SID/Label Binding TLV, the following usage rules apply.

The Range, Prefix Length and Prefix field are not used. They should be set to zero on transmission and ignored on receipt.

SID/Label Binding TLV contains a number of prefix/node SID Sub-TLVs. The TLV advertised by a proxy forwarding node P contains prefix/node SID Sub-TLVs for the node SIDs of P's neighbor nodes. Each of the Sub-TLVs is a prefix/node SID Sub-TLV defined in [RFC8667]. From the SID in a prefix/node SID Sub-TLV advertised by the Proxy Forwarding node, its prefix can be obtained through matching corresponding prefix/node SID advertised by the neighbor/protected node using TLV-135 (or 235, 236, or 237) together with the prefix/node SID Sub-TLV.

3.2.2. Advertising Binding Segment

[I-D.ietf-spring-segment-routing-policy] has defined the usage of binding-SID. For supporting binding SID proxy forwarding, a new IS-IS TLV, called Binding Segment TLV, is defined. It contains a binding SID and a list of segments (SIDs). This TLV may be

advertised in IS-IS Hello (IIH) PDUs, LSPs, or in Circuit Scoped Link State PDUs (CS-LSP) [RFC7356]. Its format is shown in Figure 9.

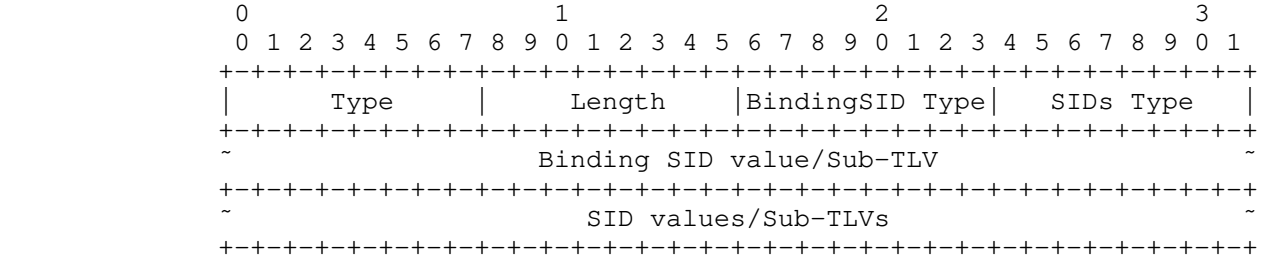


Figure 9: IS-IS Binding Segment TLV

The fields of this TLV are defined as follows:

Type: 1 octet Suggested value 152 (to be assigned by IANA)

Length: 1 octet (2 + length of Sub-TLVs/values).

Binding SID Type (BT): 1 octet indicates whether the binding SID is represented by a Sub-TLV or a value included in the TLV. For the binding SID represented by a value, it indicates the type of binding SID. The following BT values are defined:

o BT = 0: The binding SID is represented by a Sub-TLV (i.e., binding SID Sub-TLV) in the TLV. A binding SID Sub-TLV is a SID/Label Sub-TLV defined in [RFC8667]. BT != 0 indicates that the binding SID is represented by a value.

o BT = 1: The binding SID value is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o BT = 2: The binding SID value is a 32-bit SID. The length of the value is 4 octets.

SIDs Type (ST): 1 octet indicates whether the SIDs are represented by Sub-TLVs or values included in the TLV. For the SIDs represented by values, it indicates the type of SIDs. The following ST values are defined:

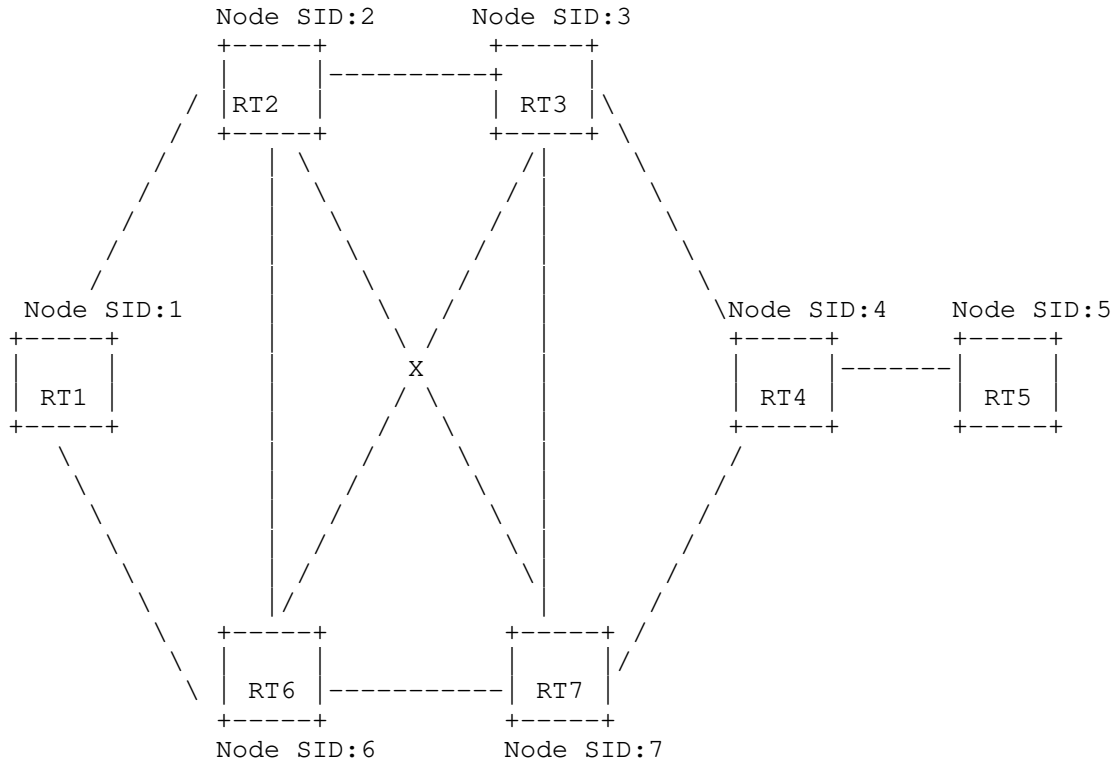
o ST = 0: The SIDs are represented by Sub-TLVs (i.e., SID Sub-TLVs) in the TLV. A SID Sub-TLV is an Adj-SID Sub-TLV, a Prefix-SID Sub-TLV or a SID/Label Sub-TLV defined in [RFC8667]. ST != 0 indicates that the SIDs are represented by values.

o ST = 1: Each of the SID values is a label, which is represented by the 20 rightmost bits. The length of the value is 3 octets.

o ST = 2: Each of the SID values is a 32-bit SID. The length of the value is 4 octets.

4. Building Proxy Forwarding Table

Figure 10 is used to illustrate the SR proxy forwarding approach. Each node N has SRGB = [N000-N999]. RT1 is an ingress node of SR domain. RT3 is a failure node. RT2 is a Point of Local Repair (PLR) node, i.e., a proxy forwarding node. Three label stacks are shown in the figure. Label Stack 1 uses only adjacency-SIDs and represents the path RT1->RT2->RT3->RT4->RT5. Label Stack 2 uses only node-SIDs and represents the ECMP-aware path RT1->RT3->RT4->RT5. Label Stack 3 uses a node-SID and a binding SID. The Binding-SID with label=100 at RT3 represents the ECMP-aware path RT3->RT4->RT5. So Label Stack 3, which consists of the node-SID for RT3 following by Binding-SID=100, represents the ECMP-aware path RT1->RT3->RT4->RT5.



Node SRGB	Adj-SID	Label Stack 1	Label Stack 2	Label Stack 3
RT1:[1000-1999]	RT1->RT2:10012	10012	1003	1003
RT2:[2000-2999]	RT2->RT3:20023	20023	3004	100
RT3:[3000-3999]	RT3->RT6:30036	30034	4005	100 is binding SID to {30034,40045}
RT4:[4000=4999]	RT3->RT7:30037	40045		
RT5:[5000-5999]	RT3->RT4:30034			
RT6:[6000-6999]	RT7->RT4:70074			
RT7:[7000-7999]	RT4->RT5:40045			

Figure 10: Topology of SR-TE Path

4.1. Advertising Proxy Forwarding

If the Point of Local Repair (PLR), for example, RT2, has the capability to do a SR proxy forwarding for all its neighboring nodes, it must advertise this capability. If the PLR can not do a SR proxy forwarding for all its neighboring nodes, but for some of them, for example, RT3, then it uses proxy Node SIDs TLV to advertise the prefix-SID learned from RT3. The TLV contains the Sub-TLV/value for the prefix/node SID of RT3 as a proxy SID. When RT3 fails, RT2 needs to maintain the Sub-TLV/value for a period of time. When the proxy forwarding table corresponding to the fault node is deleted (see section 3.2), the Sub-TLV/value is withdrawn. The nodes in the network (for example, RT1) learn the prefix/node SID TLV advertised by RT3 and the proxy Node SIDs TLV advertised by RT2. When RT3 is normal, the nodes prefer prefix/node SID TLV. When the RT3 fails, the proxy prefix/node SIDs TLV advertised by RT2 is preferred.

4.2. Building Proxy Forwarding Table

A SR proxy node P needs to build an independent proxy forwarding table for each neighbor N. The proxy forwarding table for node N contains the following information:

1: Node N's SRGB range and the difference between the SRGB start value of node P and that of node N;

2: All adjacency-SID of N and Node-SID of the node pointed to by node N's adjacency-SID.

3: The binding-SID of N and the label stack associated with the binding-SID.

Node P (PLR) uses a proxy forwarding table based on the next segment to find a node N as a backup forwarding entry to the adj-SID and Node-SID of node N. When node N fails, the proxy forwarding table needs to be maintained for a period of time, which is recommended for 30 minutes.

Node RT3 in the topology of Figure 1 is node N, and node RT2 is node P (PLR). RT2 builds the proxy forwarding table for RT3. RT2 calculates the proxy forwarding table for RT3, as shown in Figure 11.

In-label	SRGBDiffValue	Next Label	Action	Map Label
2003	-1000	30034	Fwd to RT4	2004
		30036	Fwd to RT6	2006
		30037	Fwd to RT7	2007
		100	Swap to { 30034, 40045 }	

Figure 11: RT2's Proxy Forwarding Table for RT3

5. Use of Proxy Forwarding

Segment Routing Traffic Engineering supports the creation of explicit paths using adjacency-SIDs, node-SIDs, and binding-SIDs. The label stack is a combination of one or more of adjacency-SIDs, node-SIDs, and binding-SIDs. This Section shows through example how a proxy node uses the SR proxy forwarding mechanism to forward traffic to the destination node when a node fails and the next segment of label stack is adjacency-SIDs, node-SIDs, or binding-SIDs, respectively.

5.1. Next Segment is an Adjacency Segment

As shown in Figure 1, Label Stack 1 {10012, 20023, 30034, 40045} represents SR-TE strict explicit path RT1->RT2->RT3->RT4->RT5. When RT3 fails, node RT2 acts as a PLR, and uses next adj-SID (30034) of the label stack to lookup the proxy forwarding table built by RT2 locally for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

a. RT1 pops top adj-SID 10012, and forwards the packet to RT2;

b. RT2 uses the label 20023 to identify the next hop node RT3, which has failed. RT2 pops label 20023 and queries the Proxy Forwarding Table corresponding to RT3 with label 30034. The query result is 2004. RT2 uses 2004 as the incoming label to query the label forwarding table. The next hop is RT7, and the incoming label is changed to 7004.

c. So the packet leaves RT2 out the interface to RT7 with label stack {7004, 40045}. RT4 forwards it to RT4, where the original path is rejoined.

d. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

5.2. Next Segment is a Node Segment

As shown in Figure 1, Label Stack 2 {1003, 3004, 4005} represents SR-TE loose path RT1->RT3->RT4->RT5, where 1003 is the node SID of RT3.

When the node RT3 fails, the proxy forwarding TLV advertised by the RT2 is preferred to direct the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and queries the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node RT4, which bypasses RT3. The specific steps are as follows:

a. RT1 swaps label 1003 to out-label 2003 to RT3.

b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure, RT2 pops label 2003.

c. RT2 uses 3004 as the in-label to lookup Proxy Forwarding table, The value of Map Label calculated based on SRGBDiffValue is 2004. and the query result is forwarding the packet to RT4.

d. Then RT2 queries the Routing Table to RT4, using the primary or backup path to RT4. The next hop is RT7.

e. RT2 forwards the packet to RT7. RT7 queries the local routing table to forward the packet to RT4.

f. After RT1 convergences, node SID 1003 is preferred to the proxy SID implied/advertised by RT2.

5.3. Next Segment is a Binding Segment

As shown in Figure 1, Label Stack 3 {1003, 100} represents SR-TE loose path RT1->RT3->RT4->RT5, where 100 is a Binding-SID, which represents segment list {30034, 40045}.

When the node RT3 fails, the proxy forwarding SID implied or advertised by the RT2 is preferred to forward the traffic of the RT1 to the PLR node RT2. Node RT2 acts as a PLR node and uses Binding-SID to query the proxy forwarding table locally built for RT3. The path returned is the label forwarding path to RT3's next hop node (RT4), which bypasses RT3. The specific steps are as follows:

a. RT1 swaps label 1003 to out-label 2003 to RT3.

b. RT2 receives the label forwarding packet whose top label of label stack is 2003, and searches for the local Routing Table, the behavior found is to lookup Proxy Forwarding table due to RT3 failure.

c. RT2 uses Binding-SID:100 (label 2003 has pop) as the in-label to lookup the Next Label record of the Proxy Forwarding Table, the behavior found is to swap to Segment list {30034, 40045}.

d. RT2 swaps Binding-SID:100 to Segment list {30034, 40045}, and uses the 3034 to lookup the Next Label record of the Proxy Forwarding table again. The behavior found is to forward the packet to RT4.

e. RT2 queries the Routing Table to RT4, using primary or backup path to RT4. The next hop is RT7.

f. RT2 forwards packets to RT7. RT7 queries the local routing table to forward the packet to RT4.

6. Security Considerations

The extensions to OSPF and IS-IS described in this document result in two types of behaviors in data plane when a node in a network fails. One is that for a node, which is a upstream (except for the direct upstream) node of the failed node along a SR-TE path, it continues to send the traffic to the failed node along the SR-TE path for an extended period of time. The other is that for a node, which is the direct upstream node of the failed node, it fast re-routes the traffic around the failed node to the direct downstream node of the failed node along the SR-TE path. These behaviors are internal to a network and should not cause extra security issues.

7. IANA Considerations

7.1. OSPFv2

Under Subregistry Name "OSPF Router Functional Capability Bits" within the "Open Shortest Path First v2 (OSPFv2) Parameters" [RFC7770], IANA is requested to assign one bit for Proxy Forwarding Capability as follows:

Bit number	Capability Name	Reference
31	Proxy Forwarding	This document

Under Registry Name "OSPFv2 Extended Prefix Opaque LSA TLVs" [RFC7684], IANA is requested to assign one new TLV value for OSPF Proxy Node SIDs as follows:

TLV Value	TLV Name	Reference
2	Proxy Node SIDs TLV	This document

Under Registry Name "Opaque Link-State Advertisements (LSA) Option Types" [RFC5250], IANA is requested to assign new Opaque Type registry values for Binding Segment LSA as follows:

Registry Value	Opaque Type	Reference
10	Binding Segment	This document

IANA is requested to create and maintain new registries:

- OSPFv2 Binding Segment Opaque LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	TLV Name	Definition
0	Reserved	
1	Binding Segment TLV	This Document
2-32767	Unassigned	
32768-65535	Reserved	

7.2. OSPFv3

Under Registry Name "OSPFv3 LSA Function Codes", IANA is requested to assign new registry values for Binding Segment LSA as follows:

Value	LSA Function Code Name	Reference
16	Binding Segment LSA	This document

IANA is requested to create and maintain new registries:

- OSPFv3 Binding Segment LSA TLVs

Initial values for the registry are given below. The future assignments are to be made through IETF Review [RFC5226].

Value	TLV Name	Definition
0	Reserved	
1	Binding Segment TLV	This Document
2-32767	Unassigned	
32768-65535	Reserved	

7.3. IS-IS

Under Registration "Segment Routing Capability" in the "sub-TLVs for TLV 242" registry [RFC8667], IANA is requested to assign one bit flag for Proxy Forwarding Capability as follows:

Bit number	Capability Name	Reference
2	Proxy Forwarding (PF)	This document

Under Registration "Segment Identifier/Label Binding TLV 149" [RFC8667], IANA is requested to assign one bit P-Flag as follows:

Bit number	Flag Name	Reference
5	P-Flag	This document

Under Registry Name: IS-IS TLV Codepoints, IANA is requested to assign one new TLV value for IS-IS Binding Segment as follows:

Value	TLV Name	Reference
152	Binding Segment TLV	This Document

8. Acknowledgements

The authors would like to thank Peter Psenak, Acee Lindem, Les Ginsberg, Bruno Decraene and Jeff Tantsura for their comments to this work.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <https://www.rfc-editor.org/info/rfc7356>.

[RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W., Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute Advertisement", RFC 7684, DOI 10.17487/RFC7684, November 2015, <https://www.rfc-editor.org/info/rfc7684>.

[RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <https://www.rfc-editor.org/info/rfc7770>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

9.2. Informative References

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-06 (work in progress), February 2021.

[I-D.ietf-spring-segment-protection-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Segment Protection for SR-TE Paths", draft-ietf-spring-segment-protection-sr-te-paths-00 (work in progress), September 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J., Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-07 (work in progress), July 2019.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <https://www.rfc-editor.org/info/rfc5462>.

Authors' Addresses

Zhibo Hu
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: huzhibo@huawei.com

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Junda Yao
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: yaojunda@huawei.com

Chris Bowers
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: cbowers@juniper.net

Yongqing
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou 510000
China

Email: zhuyq8@chinatelecom.cn

Yisong
China Mobile
510000
China

Email: liuyisong@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2021

D. Voyer, Ed.
Bell Canada
C. Filsfils
R. Parekh
Cisco Systems, Inc.
H. Bidgoli
Nokia
Z. Zhang
Juniper Networks

February 17, 2021

SR Replication Segment for Multi-point Service Delivery
draft-ietf-spring-sr-replication-segment-04

Abstract

This document describes the SR Replication segment for Multi-point service delivery. A SR Replication segment allows a packet to be replicated from a Replication Node to downstream nodes.

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 August 21, 2021.

Table of Contents

1. Introduction

We define a new type of segment for Segment Routing [RFC8402], called Replication segment, which allows a node (henceforth called as Replication Node) to replicate packets to a set of other nodes (called Downstream Nodes) in a Segment Routing Domain. Replication segments provide building blocks for Point-to-Multipoint Service delivery via SR Point-to-Multipoint (SR P2MP) policy. A Replication segment can replicate packet to directly connected nodes or to downstream nodes (without need for state on the transit routers). This document focuses on the Replication segment building block. The use of one or more stitched Replication segments constructed for SR P2MP Policy tree is specified in [I-D.ietf-pim-sr-p2mp-policy].

2. Replication Segment

In a Segment Routing Domain, a Replication segment is a logical construct which connects a Replication Node to a set of Downstream Nodes. A Replication segment is a local segment instantiated at a Replication node. It can be either provisioned locally on a node or programmed by a PCE. Replication segments apply equally to both SR-MPLS and SRv6 instantiations of Segment Routing.

A Replication segment is identified by the tuple <Replication-ID, Node-ID>, where:

- Replication-ID: An identifier for a Replication segment that is unique in context of the Replication Node.

- Node-ID: The address of the Replication Node that the Replication segment is for. Note that the root of a multi-point service is also a Replication Node.

In simplest case, Replication-ID can be a 32-bit number, but it can be extended or modified as required based on specific use of a Replication segment. When the PCE signals a Replication segment to its node, the <Replication-ID, Node-ID> tuple identifies the segment. Examples of such signaling and extension are described in [I-D.ietf-pim-sr-p2mp-policy].

A Replication segment includes the following elements:

- Replication SID: The Segment Identifier of a Replication segment. This is a SR-MPLS label or a SRv6 SID [RFC8402].

- Downstream Nodes: Set of nodes in Segment Routing domain to which a packet is replicated by the Replication segment.

- Replication State: See below.

The Downstream Nodes and Replication State of a Replication segment can change over time, depending on the network state and leaf nodes of a multi-point service that the segment is part of.

Replication SID identifies the Replication segment in the forwarding plane. At a Replication node, the Replication SID is the equivalent of Binding SID [I-D.ietf-spring-segment-routing-policy] of a Segment Routing Policy.

Replication State is a list of replication branches to the Downstream Nodes. In this document, each branch is abstracted to a <Downstream Node, Downstream Replication SID> tuple.

In a branch tuple, <Downstream Node> represents the reachability from the Replication Node to the Downstream Node. In its simplest form, this MAY be specified as an interface or nexthop if downstream node is adjacent to the Replication Node. The reachability may be specified in terms of Flex-Algo path (including the default algo) [I-D.ietf-lsr-flex-algo], or specified by an SR explicit path represented either by a SID-list (of one or more SIDs) or by a Segment Routing Policy [I-D.ietf-spring-segment-routing-policy].

A packet is steered into a Replication segment at a Replication Node in two ways:

- When the Active Segment [RFC8402] is a locally instantiated Replication SID

- By the root of a multi-point service based on local configuration outside the scope of this document.

In either case, the packet is replicated to each Downstream node in the associated Replication state.

If a Downstream Node is an egress (aka leaf) of the multi-point service, i.e. no further replication is needed, then that leaf node's Replication segment will not have any Replication State and the operation is NEXT. At an egress node, the Replication SID MAY be used to identify that portion of the multi-point service. Notice that the segment on the leaf node is still referred to as a Replication segment for the purpose of generalization.

A node can be a bud node, i.e. it is a Replication Node and a leaf node of a multi-point service at the same time [I-D.ietf-pim-sr-p2mp-policy].

There MUST not be any topological SID after a Replication SID in a packet. Otherwise, the behavior at Downstream nodes of a Replication segment is undefined and outside the scope of this document.

2.1. SR-MPLS data plane

When the Active Segment is a Replication SID, the processing results in a POP operation and lookup of the associated Replication state. For each replication in the Replication state, the operation is a PUSH of the downstream Replication SID and an optional segment list on to the packet which is forwarded to the Downstream node. For leaf nodes the inner packet is forwarded as per local configuration.

When the root of a multi-point service steers a packet to a Replication segment, it results in a replication to each Downstream

node in the associated replication state. The operation is a PUSH of the replication SID and an optional segment list on to the packet which is forwarded to the downstream node.

2.2. SRv6 data plane

The "Endpoint with replication" behavior (End.Replicate for short) replicates a packet and forwards the packet according to a Replication state.

When processing a packet destined to a local Replication-SID, the packet is replicated to Downstream nodes in the associated Replication state. For replication, the outer header is re-used, and the Downstream Replication SID is written into the outer IPv6 header destination address.If required, an optional segment list is used to encapsulate the replicated packet via H.Encaps. For a leaf node, the packet is decapsulated and the inner packet is forwarded as per local configuration.

When the root of a multi-point service steers a packet into a Replication segment, for each replication, H.Encaps is used to encapsulate the packet with the segment list to the Downstream node .

An End.Replicate SID MUST only appear as the ultimate SID in a SID-list. An implementation that receives a packet destined to a locally instantiated End.Replicate SID that is not the ultimate segment SHOULD reply with ICMP Parameter Problem error (Erroneous header field encountered) and discard the packet.

3. Use Cases

In the simplest use case, a single Replication segment includes the root node of a multi-point service and the egress/leaf nodes of the service as all the Downstream Nodes. This achieves Ingress Replication [RFC7988] that has been widely used for MVPN [RFC6513] and EVPN [RFC7432] BUM (Broadcast, Unknown and Multicast) traffic.

Replication segments can also be used as building blocks for replication trees when Replication segments on the root, intermediate Replication Nodes and leaf nodes are stitched together to achieve efficient replication. That is specified in [I-D.ietf-pim-sr-p2mp-policy].

4. IANA Considerations

This document requires registration of End.Replicate behavior in "SRv6 Endpoint Behaviors" sub-registry of "Segment Routing Parameters" top-level registry.

Value	Hex	Endpoint behavior	Reference
TBD	TBD	End.Replicate	[This.ID]

Table 1: IETF - SRv6 Endpoint Behaviors

5. Security Considerations

There are no additional security risks introduced by this design.

6. Acknowledgements

The authors would like to acknowledge Siva Sivabalan, Mike Koldychev, Vishnu Pavan Beeram, Alexander Vainshtein, Bruno Decraene and Joel Halpern for their valuable inputs.

7. Contributors

Clayton Hassen
Bell Canada
Vancouver
Canada

Email: clayton.hassen@bell.ca

Kurtis Gillis
Bell Canada
Halifax
Canada

Email: kurtis.gillis@bell.ca

Arvind Venkateswaran
Cisco Systems, Inc.
San Jose
US

Email: arvvenka@cisco.com

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Swadesh Agrawal

Cisco Systems, Inc.
San Jose
US

Email: swaagraw@cisco.com

Jayant Kotalwar
Nokia
Mountain View
US

Email: jayant.kotalwar@nokia.com

Tanmoy Kundu
Nokia
Mountain View
US

Email: tanmoy.kundu@nokia.com

Andrew Stone
Nokia
Ottawa
Canada

Email: andrew.stone@nokia.com

Tarek Saad
Juniper Networks
Canada

Email:tsaad@juniper.net

8. References

8.1. Normative References

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

8.2. Informative References

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

[I-D.ietf-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-01 (work in progress), October 2020.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

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

[RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <https://www.rfc-editor.org/info/rfc7988>.

Appendix A. Illustration of a Replication Segment

This section illustrates an example of a single Replication segment. Examples showing Replication segment stitched together to form P2MP tree (based on SR P2MP policy) are in [I-D.ietf-pim-sr-p2mp-policy].

Consider the following topology:

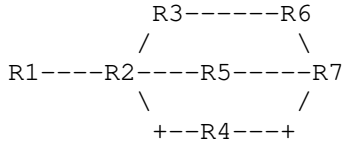


Figure 1

A.1. SR-MPLS

In this example, the Node-SID of a node Rn is N-SIDn and Adjacency-SID from node Rm to node Rn is A-SIDmn. Interface between Rm and Rn is Lmn.

Assume a Replication segment identified with R-ID at Replication Node R1 and downstream Nodes R2, R6 and R7. The Replication SID at node n is R-SIDn. A packet replicated from R1 to R7 has to traverse R4.

The Replication segment state at nodes R1, R2, R6 and R7 is shown below. Note nodes R3, R4 and R5 do not have state for the Replication segment.

Replication segment at R1:

```
Replication segment <R-ID,R1>:
 Replication SID: R-SID1
 Replication State:
   R2: <R-SID2->L12>
   R6: <N-SID6, R-SID6>
   R7: <N-SID4, A-SID47, R-SID7>
```

Replication to R2 steers packet directly to R2 on interface L12. Replication to R6, using N-SID6, steers packet via IGP shortest path to that node. Replication to R7 is steered via R4, using N-SID4 and then adjacency SID A-sID47 to R7.

Replication segment at R2:

```
Replication segment <R-ID,R2>:
 Replication SID: R-SID2
 Replication State:
   R2: <Leaf>

Replication segment at R6:

Replication segment <R-ID,R6>:
 Replication SID: R-SID6
 Replication State:
   R6: <Leaf>

Replication segment at R7:

Replication segment <R-ID,R7>:
 Replication SID: R-SID7
 Replication State:
   R7: <Leaf>
```

When a packet is steered into the Replication segment at R1:

- Since R1 is directly connected to R2, R1 performs PUSH operation with just <R-SID2> label for the replicated copy and sends it to R2 on interface L12. R2, as Leaf, performs NEXT operation, pops R-SID2 label and delivers the payload.

- R1 performs PUSH operation with <N-SID6, R-SID6> label stack for the replicated copy to R6 and sends it to R2, the nexthop on IGP shortest path to R6. R2 performs CONTINUE operation on N-SID6 and forwards it to R3. R3 is the penultimate hop for N-SID6; it performs penultimate hop popping, which corresponds to the NEXT operation and the packet is then sent to R6 with <R-SID6> in the label stack. R6, as Leaf, performs NEXT operation, pops R-SID6 label and delivers the payload.

- R1 performs PUSH operation with <N-SID4, A-SID47, R-SID7> label stack for the replicated copy to R7 and sends it to R2, the nexthop on IGP shortest path to R4. R2 is the penultimate hop for N-SID4; it performs penultimate hop popping, which corresponds to the NEXT operation and the packet is then sent to R4 with <A-SID47, R-SID1> in the label stack. R4 performs NEXT operation, pops A-SID47, and delivers packet to R7 with <R-SID7> in the label stack. R7, as Leaf, performs NEXT operation, pops R-SID7 label and delivers the payload.

A.2. SRv6

For SRv6 , we use SID allocation scheme, reproduced below, from Illustrations for SRv6 Network Programming [I-D.filsfils-spring-srv6-net-pgm-illustration]

2001:db8::/32 is an IPv6 block allocated by a RIR to the operator

2001:db8:0::/48 is dedicated to the internal address space

2001:db8:cccc::/48 is dedicated to the internal SRv6 SID space

We assume a location expressed in 64 bits and a function expressed in 16 bits

Node k has a classic IPv6 loopback address 2001:db8::k/128 which is advertised in the IGP

Node k has 2001:db8:cccc:k::/64 for its local SID space. Its SIDs will be explicitly assigned from that block

Node k advertises 2001:db8:cccc:k::/64 in its IGP

Function :1:: (function 1, for short) represents the End function with PSP support

Function :Cn:: (function Cn, for short) represents the End.X function from to Node n

Each node k has:

An explicit SID instantiation 2001:db8:cccc:k:1::/128 bound to an End function with additional support for PSP

An explicit SID instantiation 2001:db8:cccc:k:Cj::/128 bound to an End.X function to neighbor J with additional support for PSP

An explicit SID instantiation 2001:db8:cccc:k:Fk::/128 bound to an End.Replcate function

Assume a Replication segment identified with R-ID at Replication Node R1 and downstream Nodes R2, R6 and R7. The Replication SID at node k, bound to an End.Replcate function, is 2001:db8:cccc:k:Fk::/128. A packet replicated from R1 to R7 has to traverse R4.

The Replication segment state at nodes R1, R2, R6 and R7 is shown below. Note nodes R3, R4 and R5 do not have state for the Replication segment.

```
   Replication segment at R1:

   Replication segment <R-ID,R1>:
    Replication SID: 2001:db8:cccc:1:F1::0
    Replication State:
      R2: <2001:db8:cccc:2:F2::0->L12>
      R6: <2001:db8:cccc:6:F6::0>
      R7: <2001:db8:cccc:4:C7::0, 2001:db8:cccc:7:F7::0>
```

Replication to R2 steers packet directly to R2 on interface L12. Replication to R6, using 2001:db8:cccc:6:F6::0, steers packet via IGP shortest path to that node. Replication to R7 is steered via R4, using End.X SID 2001:db8:cccc:4:C7::0 at R4 to R7.

```
   Replication segment at R2:

   Replication segment <R-ID,R2>:
    Replication SID: 2001:db8:cccc:2:F2::0
    Replication State:
      R2: <Leaf>

   Replication segment at R6:

   Replication segment <R-ID,R6>:
    Replication SID: 2001:db8:cccc:6:F6::0
    Replication State:
      R6: <Leaf>

   Replication segment at R7:

   Replication segment <R-ID,R7>:
    Replication SID: 2001:db8:cccc:7:F7::0
    Replication State:
      R7: <Leaf>
```

When a packet, (A,B2), is steered into the Replication segment at R1:

- Since R1 is directly connected to R2, R1 creates encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:2:F2::0) (A, B2), and sends it to R2 on interface L12. R2, as Leaf, removes outer IPv6 header and delivers the payload.

- R1 creates encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:6:F6::0) (A, B2) then forwards the resulting packet on the shortest path to 2001:db8:cccc:6::/64. R2 and R3 forward the packet using 2001:db8:cccc:6::/64. R6, as Leaf, removes outer IPv6 header and delivers the payload.

- R1 creates encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:4:C7::0) (2001:db8:cccc:7:F7::0; SL=1) (A, B2) and sends it to R2, the nexthop on IGP shortest path to 2001:db8:cccc:4::/64. R2 forwards packet to R4 using 2001:db8:cccc:4::/64. R4 executes End.X function on 2001:db8:cccc:4:C7::0, performs PSP action, removes SRH and sends resulting packet (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) to R7. R7, as Leaf, removes outer IPv6 header and delivers the payload.

Authors' Addresses

Daniel Voyer (editor)
Bell Canada
Montreal
CA

Email: daniel.voyer@bell.ca

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Rishabh Parekh
Cisco Systems, Inc.
San Jose
US

Email: riparekh@cisco.com

Hooman Bidgoli
Nokia
Ottawa
CA

Email: hooman.bidgoli@nokia.com

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net

SPRING
Internet-Draft
Intended status: Standards Track
Expires: June 18, 2021

H. Song
Futurewei Technologies
T. Pan
BUPT

December 15, 2020

SRv6 In-situ Active Measurement
draft-song-spring-siam-00

Abstract

This draft describes an in-band active measurement method for SRv6. A probing packet contains an SRH with a flag bit set. The IOAM header and data are encapsulated in UDP payload. The probing packet originates from a segment source node and terminates at a configured segment endpoint node. Each segment node on the path, when detecting the flag, parses the UDP header and the IOAM header, and adds data to the IOAM node data fields. Multiple applications can be supported by the method.

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/.

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 June 18, 2021.

Table of Contents

1. Introduction

To support SRv6 network operation, we need various means to collect data and measure the performance of SRv6 network. [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.

[I-D.ietf-ippm-ioam-data] supports extensible data collection for user traffic. It is beneficial for SRv6 network monitor and measurement. [I-D.ali-spring-ioam-srv6] proposes to encapsulate IOAM in SRH TLV. However, IOAM's overhead may cause packet fragmentation and its processing may affect the packet forwarding throughput. Moreover, due to the extension header limitations asserted by [RFC8200], it is not easy to come up with a scheme to encapsulate the IOAM header and data in other locations in SRv6 user packets.

Fortunately, the forwarding behavior in SRv6 networks is determined by the SRH. The IOAM header and data do not need to be added to user packets. Instead, they can be encapsulated in an independent packet. As long as this packet has the same SRH as the user packet, the data collected can faithfully reflect the user packet's forwarding experience, so the result is similar to that by applying IOAM on SRv6 user packets. This approach retains the benefits of in-situ measurement but avoids the aforementioned issues.

The IOAM header and data processing can be done in slow path, without worrying about delaying the user traffic. Because of this, the potential limitation of the forwarding hardware's header processing capability (e.g., the hearder parsing depth) is no longer an issue.

This SR-based active measurement approach also supports some other 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.

The approach is built on existing protocol components with limited extra requirements.

2. In-situ Active Measurement 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.

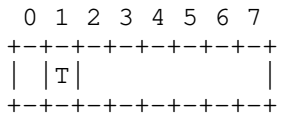


Figure 1: A Hierarchical Edge Network

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 further verify this packet is an active probing packet and the UDP payload encapsulates the IOAM header and data as

specified in [I-D.ietf-ippm-ioam-data]. The source UDP port can be used as sequence number to track the probing packets on a specific SR path.

The complete active probing packet format is shown in Figure 2.

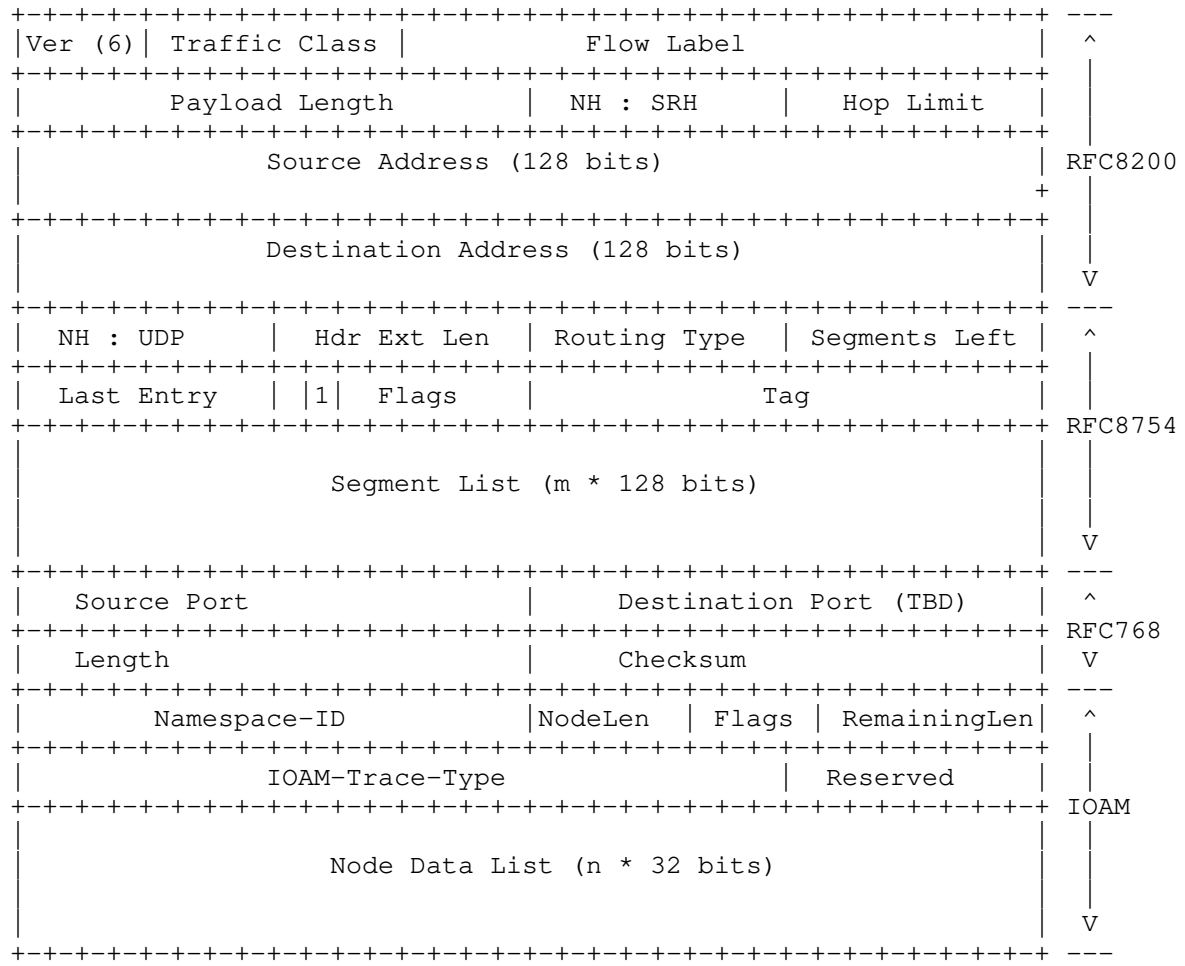


Figure 2: The active probing packet format

3. Network Operation

The SR source node constructs the probing packets. 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, 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 and analyzed according to configuration.

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.

4. Applications

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

- As described in Section 1, this is an easy way to apply IOAM in SRv6. 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 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.

- To collect the network wide telemetry data and gain global visibility within a SRv6 domain, we can apply the algorithm described in [I-D.tian-bupt-inwt-mechanism-policy] to calculate

the optimal SR paths, and construct probing packets on these paths.

5. Probing Packet Type Extension

The same scheme is also suitable for other types of probing packets. For example, The probing packets can carry IOAM E2E option header and data, IOAM DEX option header, and other telemetry headers and data. It is easy to use different reserved UDP port numbers to differentiate the payload types.

6. Security Considerations

7. IANA Considerations

An SRH Flag bit 'T'. The bit position TBD

A optional UDP destination port number indicating IOAM payload (TBD)

8. Acknowledgments

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

9.2. Informative References

[I-D.ali-spring-ioam-srv6]
Ali, Z., Gandhi, R., Filsfils, C., Brockners, F., Nainar, N., Pignataro, C., Li, C., Chen, M., and G. Dawra, "Segment Routing Header encapsulation for In-situ OAM Data", draft-ali-spring-ioam-srv6-03 (work in progress), November 2020.

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.tian-bupt-inwt-mechanism-policy]
Pan, T., Gao, M., Song, E., Bian, Z., and X. Lin, "In-band Network-Wide Telemetry", draft-tian-bupt-inwt-mechanism-policy-01 (work in progress), October 2020.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

Authors' Addresses

Haoyu Song
Futurewei Technologies
Santa Clara
USA

Email: haoyu.song@futurewei.com

Tian Pan
BUPT
Beijing
China

Email: pan@bupt.edu.cn

SPRING
Internet-Draft
Intended status: Informational
Expires: August 23, 2021

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

February 19, 2021

Compressed SRv6 SID List Analysis
draft-srcompdt-spring-compression-analysis-00

Abstract

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

Status of This Memo

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 August 23, 2021.

Table of Contents

1. Introduction

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

- CSID - [I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc] - Describes two new SRv6 SIDs, a combination of SIDs from [I-D.filsfils-spring-net-pgm-extension-srv6-usid] and [I-D.cl-spring-generalized-srv6-for-cmpr]
- CRH - [I-D.bonica-6man-comp-rtg-hdr] - Requires two new routing header types and a label mapping technique.

- o VSID - [I-D.decraene-spring-srv6-vlsid] - Defines a set of SID behaviors to access smaller SIDs within the SR header.
- o UID - [I-D.mirsky-6man-unified-id-sr] - Extends the SRH to carry MPLS labels or IPv4 addresses.

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

2. SRv6 Compression Requirements

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

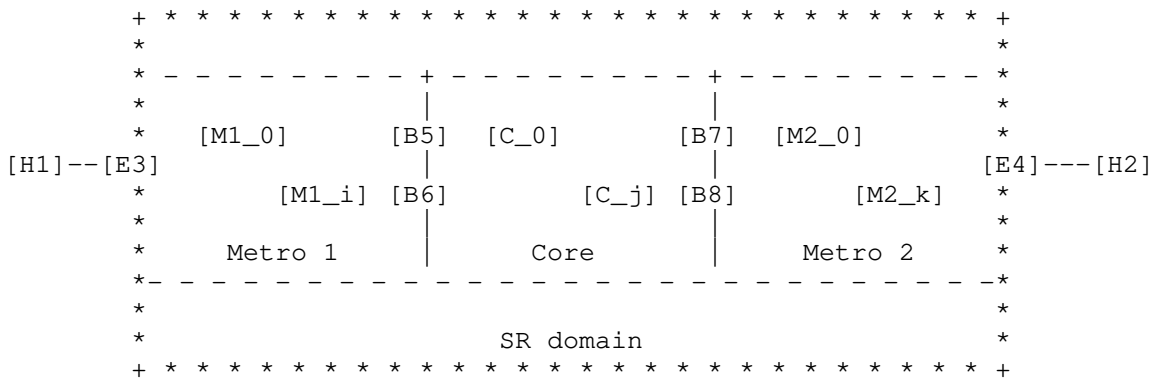


Figure 1: Sample SR Domain

- o H1 and H2 are hosts outside the SR domain
- o E3 and E4 are SR domain edge routers
- o Metro 1, Core and Metro 2 are sub-domains with independent IGP instances
- o B5 and B6 are border routers between the Metro 1 and Core
- o B7 and B8 are border routers between the Metro 2 and Core
- o M1_1..M1_i are routers in Metro 1
- o C_1..C_j are routers in Core
- o M2_1..M2_k are routers in Metro 2

2.1. Encapsulation Header Size

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

Encapsulation header size is evaluated against multiple reference scenarios.

2.2. Forwarding Efficiency

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

2.2.1. Headers Parsed (PRS)

This section records and summarizes differences in header parsing for different SID types.

- Segment lists may contain transport, adjacency, service, binding or VPN segments.

16-bit	CSID	CRH	VSID	UIDSR

Table 1: Headers Parsed, 16-bit SIDs

32-bit	CSID	CRH	VSID	UIDSR

Table 2: Headers Parsed, 32-bit SIDs

Conclusion:

2.2.2. Lookups Performed (LKU)

Some proposals require a different number of lookups per packet, depending on the SID type and segment list.

A strict TE path is considered with a 1D(1..15T).V segment list, where each transport segment is an adjacency segment.

16-bit and 32-bit	CSID	CRH	VSID	UIDSR
D.LKU(1D(1..15T).V)				

Table 3: Lookups, Strict TE Paths

Conclusion:

A loose TE path consists of a combination of prefix and adjacency segments

16-bit and 32-bit	CSID	CRH	VSID	UIDSR

Table 4: Lookups, Loose TE Paths

Conclusion:

2.3. State Efficiency

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

State efficiency is analyzed in a single sub-domain of the SR domain, where three parameters are considered:

- N: the number of nodes in the sub-domain
- I: the number of IGP algorithms [I-D.ietf-lsr-flex-algo] configured
- A: the number of local adjacency SIDs

For a core sub-domain with 1000 nodes, two IGP algorithms, and 100 adjacencies per node:

- N=1000, I=2, A=100

16-bit and 32-bit	CSID	CRH	VSID	UIDSR
S(N1000,I2,A100)				

Table 5: Forwarding State

Conclusion:

3. SRv6 Specific Requirements

3.1. SRv6 Based

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

This section records the use of SRv6 standards for compression.

	CSID	CRH	VSID	UIDSR
U.RFC8402				
U.RFC8754				
U.PGM				
U.IGP				
U.BGP				
U.POL				
U.BLS				
U.SVC				
U.ALG				
U.OAM				

Table 6: SRv6 Based

Conclusion:

3.2. Functional Requirements

3.2.1. SRv6 Functionality

A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement ensures no SRv6 functionality is lost. It is particularly important to understand

how a proposal, as evaluated in section "SRv6 Based", provides this functionality.

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

Draft reference Abbreviations
IDNETPGM: [I-D.ietf-spring-srv6-network-programming]
IDSRPOL: [I-D.ietf-spring-segment-routing-policy]
IDEXT: [I-D.ietf-lsr-isis-srv6-extensions]
IDBGPSVC: [I-D.ietf-bess-srv6-services]
IDBGPLS: [I-D.ietf-idr-bgpls-srv6-ext]
IDSVCP: [I-D.ietf-spring-sr-service-programming]
IDOAM: [I-D.ietf-6man-spring-srv6-oam]
IDFLEXALG: [I-D.ietf-lsr-flex-algo]
IDTILFA: [I-D.ietf-rtgwg-segment-routing-ti-lfa]

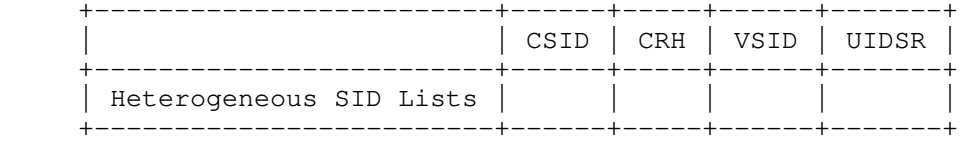
	CSID	CRH	VSID	UIDSR
F.SID				
F.Scope				
F.PFX				
F.ADJ				
F.BIND				
F.PEER				
F.SVC				
F.ALG				
F.TILFA				
F.SEC				
F.IGP				
F.BGP				
F.POL				
F.BLS				
F.SFC				
F.PING				

Table 7: SRv6 Functionality

Conclusion:

3.2.2. Heterogeneous SID Lists

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



Conclusion:

3.2.3. SID List Length

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

Conclusion:

3.2.4. SID Summarization

The solution MUST be compatible with segment summarization.

In inter sub-domain deployments with summarization:

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

Without summarization, border router SIDs must be leaked:

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

	CSID	CRH	VSID	UIDSR
SID Summarization				

Conclusion:

3.3. Operational Requirements

3.3.1. Lossless Compression

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

	CSID	CRH	VSID	UIDSR
Lossless Compression				

Conclusion:

3.4. Scalability Requirements

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

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

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

	CSID	CRH	VSID	UIDSR
Adjacency Segment Scale 65000 Prefix Segment Scale 1000000 Service Scale 1000000				

Table 8: Scale Requirements

Conclusion:

4. Protocol Design Requirements

4.1. SRv6 Base Coexistance

The compression proposal MUST support deployment in existing SRv6 networks.

	CSID	CRH	VSID	UIDSR
SRv6 Base Coexistence				

Conclusion:

5. Security Requirements

5.1. Security Mechanismns

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

	CSID	CRH	VSID	UIDSR
Security Mechanisms				

Conclusion:

5.2. SR Domain Protection

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

	CSID	CRH	VSID	UIDSR
SR Domain Protection				

Conclusion:

6. Conclusions

Encapsulation Header Size

-

Forwarding Efficiency

-

State Efficiency

-

SRv6 Based

-

SRv6 Functionality

-

Heterogeneous SID lists

-

SID List Length

-

SID Summarization

-

Operational Requirements

-

Protocol Design Requirements

-

Scalability Requirements

-

Protocol Design Requirements

-

Security Requirements

-

7. Normative References

[I-D.bonica-6man-comp-rtg-hdr]
Bonica, R., Kamite, Y., Alston, A., Henriques, D., and L. Jalil, "The IPv6 Compact Routing Header (CRH)", draft-bonica-6man-comp-rtg-hdr-24 (work in progress), January 2021.

[I-D.cl-spring-generalized-srv6-for-cmpr]
Cheng, W., Li, Z., Li, C., Clad, F., Aihua, L., Xie, C., Liu, Y., and S. Zadok, "Generalized SRv6 Network Programming for SRv6 Compression", draft-cl-spring-generalized-srv6-for-cmpr-02 (work in progress), November 2020.

[I-D.decraene-spring-srv6-vlsid]
Decraene, B., Raszuk, R., Li, Z., and C. Li, "SRv6 vSID: Network Programming extension for variable length SIDs", draft-decraene-spring-srv6-vlsid-04 (work in progress), September 2020.

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

[I-D.filsfilscheng-spring-srv6-srh-comp-sl-enc]
Cheng, W., Filsfils, C., Li, Z., Cai, D., Voyer, D., Clad, F., Zadok, S., Guichard, J., and L. Aihua, "Compressed SRv6 Segment List Encoding in SRH", draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-02 (work in progress), November 2020.

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-05 (work in progress), November 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-11 (work in progress), October 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.mirsky-6man-unified-id-sr]
Cheng, W., Mirsky, G., Peng, S., Aihua, L., and G. Mishra, "Unified Identifier in IPv6 Segment Routing Networks", draft-mirsky-6man-unified-id-sr-08 (work in progress), January 2021.

[I-D.srcompdt-spring-compression-requirement]
Cheng, W., "Compressed SRv6 SID List Requirements", draft-srcompdt-spring-compression-requirement-03 (work in progress), January 2021.

Authors' Addresses

Ron Bonica
Juniper

Email: rbonica@juniper.net

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Darren Dukes
Cisco Systems

Email: ddukes@cisco.com

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

SPRING
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

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

February 22, 2021

Compressed SRv6 SID List Requirements
draft-srcompdt-spring-compression-requirement-04

Abstract

This document specifies requirements for solutions to compress SRv6 SID lists.

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 August 26, 2021.

Table of Contents

1. Introduction

The SPRING working group defined SRv6, with [RFC8402] describing how the Segment Routing (SR) architecture is instantiated on two data-planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). SRv6 uses a routing header called the SR Header (SRH) [RFC8754] and defines SRv6 SID behaviors and a registry for identifying them in [I-D.ietf-spring-srv6-network-programming]. SRv6 is a proposed standard and is deployed today.

The SPRING working group has observed that some use cases, such as strict path TE, may require long SRv6 SID lists. There are several proposed methods to reduce the resulting SRv6 encapsulation size by compressing the SID list.

The SPRING working group formed a design team to define requirements for, and analyze proposals to, compress SRv6 SID lists.

It is a goal of the design team to identify solutions to SRv6 SID list compression that are based on the SRv6 standards. As such, this document provides requirements for SRv6 SID list compression solutions that utilize the existing SRv6 data plane and control plane.

It is also a goal of the design team to consider proposals that are not based on the SRv6 data plane and control plane. As such, this document includes requirements to evaluate whether a compression proposal provides all the functionality of SRv6 (section "SRv6 Functionality") in addition to satisfying compression specific requirements.

For each requirement, a description, rationale and metrics are described.

The design team will produce a separate document to analyze the proposals.

This document is a draft; additional requirements are under review, additional requirements will be added, and current requirements may change. Appendix A contains a subset of requirements without unanimous consensus. Additional requirements without unanimous consensus are not in the appendix.

2. Conventions used in this document

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.

2.2. Terminology

SR: Segment Routing

SRH: Segment Routing Header

MPLS: Multiprotocol Label Switching

SR-MPLS: Segment Routing over MPLS data plane

SID: Segment Identifier

SRv6: Segment Routing over IPv6

SRv6 SID List: A list of SRv6 SIDs

Compression proposal: A proposal to compress SRv6 SID lists

SRv6 base: SRv6 as defined in [RFC8402], [RFC8754], [I-D.ietf-spring-srv6-network-programming]

SID numbering space: may be implemented as

- a single IGP instance
- a single IGP level or area
- two or more autonomous systems that coordinate SID numbering space
- two or more IGP instances that coordinate SID numbering space

SRv6 Encapsulation Header: The IPv6 header, and any extension headers preceding a payload, used to implement a SRv6 base or compression proposal.

3. SRv6 SID List Compression Requirements

3.1. Dataplane Efficiency and Performance Requirements

3.1.1. Encapsulation Header Size

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

Rationale: A smaller SRv6 encapsulation results in better MTU efficiency.

Metric: Compression is the ratio of the IPv6 encapsulation size of SRv6 as defined in [RFC8402], [RFC8754], [I-D.ietf-spring-srv6-network-programming] vs the IPv6 encapsulation size of a given proposal. The encapsulation savings of a compression proposal vs the SRv6 base is a useful measurement to compare proposals.

The encapsulation metric (E) records the number of bytes required for a proposal to encapsulate a packet given a specific segment list.

- E(proposal, segment list).

The encapsulation savings(ES)records the encapsulation savings for a proposal to encapsulate a packet given a specific segment list.

- ES(proposal, segment list) = 1 - E(proposal, segment list)/E(SRv6 base, segment list).

3.1.2. Forwarding Efficiency

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

Rationale: Efficiency in bits on the wire and processing efficiency are both important. Optimizing one at the expense of the other may lead to significant performance impact.

Metric: The data plane efficiency metric (D) records the data plane forwarding efficiency of the proposed solution. Two metrics are used and recorded at each segment endpoint:

- D.PRS(segment list): number of headers parsed during processing of the segment list, starting from and including the IPv6 header.
- D.LKU(segment list): number of FIB lookups during processing of the segment list. The type of lookup is also recorded as longest prefix match (LPM) or exact match (EM)

3.1.3. State Efficiency

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

Rationale: Additional state increases the complexity of the control plane and data plane. It can also result in an increase in memory usage.

Metric: The state efficiency metric (S) records the amount of additional forwarding state required by the proposed solution.

- S(node parameters): the number of additional forwarding states that need to be stored at a node, given a set of node parameters consisting of the number of nodes in the network, number of local interfaces, number of adjacencies. The forwarding state is counted as entries required in a Forwarding Information Base (FIB) at a node.

4. SRv6 Specific Requirements

4.1. SRv6 Based

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

Rationale: A compression proposal built on existing IETF standards is preferable to creating new standards with equivalent functionality and performance.

Metric: The utilization metric (U) records whether a proposal utilizes the SRv6 specifications.

Utilization is recorded in a table, with a column per proposal and rows consisting of the following metrics:

- U.RFC8402: utilizes [RFC8402].
- U.RFC8754: utilizes [RFC8754].
- U.PGM: utilizes [I-D.ietf-spring-srv6-network-programming].
- U.IGP: utilizes [I-D.ietf-lsr-isis-srv6-extensions].
- U.BGP: utilizes [I-D.ietf-bess-srv6-services].
- U.POL: utilizes [I-D.ietf-spring-segment-routing-policy].
- U.BLS: utilizes [I-D.ietf-idr-bgpls-srv6-ext].
- U.SVC: utilizes [I-D.ietf-spring-sr-service-programming].
- U.OAM: utilizes [I-D.ietf-6man-spring-srv6-oam].
- U.ALG: utilizes [I-D.ietf-lsr-flex-algo].

Each cell contains "yes" for utilizes, or "no" for does not utilize.

4.2. Functional Requirements

4.2.1. SRv6 Functionality

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

Rationale: Operators require SRv6 functionality. Evaluating the extent to which a proposal supports SRv6 functionality is important for operators and implementors to understand the impact on network operations.

Metric: The Functionality metric (F) records whether a proposal supports SRv6 functionality and how the functionality is provided.

Functionality is recorded in a table with columns for each proposal and rows consisting of the following metrics:

- F.SID: Supports SRv6 SID functionality as described in [RFC8402]
- F.SCOPE: Supports globally and locally scoped SID functionality as described in [RFC8402]
- F.PFX: Supports prefix SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.ADJ: Supports adjacency SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.BIND: Supports binding SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.PEER: Supports BGP peering SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.SVC: Supports L3 and L2 VPN service SID functionality as described in [I-D.ietf-spring-srv6-network-programming]
- F.ALG: Supports flexible algorithms functionality as described in [I-D.ietf-lsr-flex-algo]
- F.TILFA: Supports TI-LFA functionality as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- F.SEC: Supports securing an SR domain with ingress filtering as functionally defined in [RFC8754]
- F.IGP: Supports distributing topological SIDs and behaviors via ISIS as functionally described in [I-D.ietf-lsr-isis-srv6-extensions]
- F.BGP: Supports BGP VPNs as functionally described in [I-D.ietf-bess-srv6-services]

- o F.POL: Supports SR policies and steering traffic over those policies as funcitonally described in [I-D.ietf-spring-segment-routing-policy]
- o F.BLS: Supports Link State distribution via BGP as functionally described in [I-D.ietf-idr-bgpls-srv6-ext]
- o F.SFC: Supports stateless service programming as functionally described in [I-D.ietf-spring-sr-service-programming]
- o F.PING: Supports pinging a SID to verify the SID is implemented as functionally described in [I-D.ietf-6man-spring-srv6-oam]

Each cell contains the specification name documenting the functionality.

4.2.2. Heterogeneous SID lists

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

Rationale: Support of SID lists with compressed and non-compressed SIDs reduces encapsulation size when not all SRv6 nodes deploy the compression proposal or 128-bit SIDs are required.

Metric: A compliant compression proposal supports both:

- o classic 128-bit SRv6 SIDs in the IPv6 Destination Address field
- o segment lists (i.e., paths) with both compressed and 128-bit SRv6 SIDs.

4.2.3. SID list length

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

Rationale: Strict TE paths require SID list lengths proportional to the diameter of the SR domain.

Metric: The compression proposal must be able to steer a packet through an SR path that contains up to sixteen segments.

4.2.4. SID summarization

Description: The solution MUST be compatible with segment summarization.

Rationale: Summarization of segments is a key benefit of SRv6 vs SR MPLS. In interdomain deployments, any node can reach any other node via a single prefix segment. Without summarization, border router SIDs must be leaked, and an additional global prefix segment is required for each domain border to be traversed.

Metric: A solution supports summarization when segments can be summarized for advertisement into other IGP domains or levels.

4.3. Operational Requirements

4.3.1. Lossless Compression

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

Rationale: In SRv6, we can represent a path to meet certain objectives. A compression proposal needs to support the objectives with the same path.

Metric: Information present in the pre-compression segment list MUST also be present in the post-compression SID list.

4.3.2. Preservation of non-routing information

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

Rationale: SRv6 ingress nodes encode non-routing information in the IPv6 header chain. This information can be encoded in the following fields:

- Hop Count
- DSCP bits
- ECN bits
- Flow label
- HBH Options Extension header
- Fragment Extension header
- Authentication Extension header
- Encrypted Security Payload Extension header
- Destination Options Extension header

Some of these fields are mutable (e.g., Hop Count) while others are immutable (e.g., Fragment Extension Header).

Some of these fields contain information that is required by every node along a packet's delivery path (e.g., Hop Count). Others contain information that is required only by the packet's ultimate destination (e.g., Fragment Extension Header).

Therefore, the compression mechanism MUST NOT prevent this information from being delivered, in an IPv6 header chain, to any node that needs it.

Metric: The SR source node encapsulates its payload (e.g.., Ethernet, IP, TCP) in an IPv6 header. The SRv6 header contains both routing and non-routing information. The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node.

4.4. Scalability Requirements

4.4.1. Adjacency segment scale

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

Rationale: Typically, network operators deploy networks with tens or hundreds of adjacency segments per node, but some network operators may deploy networks that use more adjacency segments per node.

Metric: A proposal that allows 65000 adjacency segments per node satisfies this requirement.

4.4.2. Prefix segment scale

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

Rationale: Typically, network operators deploy networks with thousands of prefix segments per SID numbering space, but some network operators may deploy networks that use more prefix segments per SID numbering space.

Metric: A proposal that allows 1 million prefix segments per SID numbering space satisfies this requirement.

4.4.3. Service Scale

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

Rationale: Typically, network operators deploy networks with tens to hundreds of thousands of services per node, but some network operators may deploy networks that use more services per node.

Metric: A proposal that allows 1 million services per node satisfies this requirement.

4.4.4. Compression Levels

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

Rationale: The compression proposal will be deployed in networks of varying size with SID numbering spaces of varying size. Network and service scale can directly impact SID length and the ability of a proposal to compress the SID list.

Metric: A compression proposal that supports relatively better compression for smaller SID numbering spaces and service scale satisfies this requirement.

5. Protocol Design Requirements

5.1. SRv6 Base Coexistence

Description: The compression proposal MUST support deployment in existing SRv6 networks.

Rationale: SRv6 is deployed today. A compression proposal that interoperates well with SRv6, as deployed, will reduce the overhead and simplify operations. For Network operators who would migrate to compressed SRv6 SID lists, the migration is expected to gradually occur over a period of time as they upgrade networks, domains, device families and software instances.

Metric: A compliant compression proposal provides the following

- Supports simultaneous deployment at a node with current SRv6 SIDs.
- Supports simultaneous deployment at a node with current SRv6 control plane.
- Supports simultaneous operation of current SRv6 paths with compressed paths.
- Supports the behaviors in [I-D.ietf-spring-srv6-network-programming].
- Does not require removal of existing IPv6 address planning.

6. Security Requirements

6.1. Security Mechanisms

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

Rationale: It is important to identify security issues and how to address them in any specification.

Metric: A compression solution that does not introduce unresolved security issues meets this requirement.

6.2. SR Domain Protection

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

Rationale: The unauthorized use of SIDs within the SR domain by nodes outside the domain can disrupt an operators' network.

Metric: A compliant solution describes how access to SIDs within the SR domain is denied to nodes outside the SR domain.

7. IANA Considerations

This document has no requests to IANA.

8. Security Considerations

TBD

9. Contributors

The following individuals contributed to this draft

Sanders Steffann, SJM Steffann Consultancy, sander@steffann.nl

10. Normative References

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-05 (work in progress), November 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-11 (work in progress), October 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

Authors' Addresses

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

Ron Bonica
Juniper

Email: rbonica@juniper.net

Darren Dukes
Cisco Systems

Email: ddukes@cisco.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

SPRING
Internet-Draft
Intended status: Informational
Expires: September 30, 2021

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

March 29, 2021

Compressed SRv6 SID List Requirements
draft-srcompdt-spring-compression-requirement-06

Abstract

This document specifies requirements for solutions to compress SRv6 SID lists.

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 30, 2021.

Table of Contents

1. Introduction

The SPRING working group defined SRv6, with [RFC8402] describing how the Segment Routing (SR) architecture is instantiated on two data-planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). SRv6 uses a routing header called the SR Header (SRH) [RFC8754] and defines SRv6 SID behaviors and a registry for identifying them in [I-D.ietf-spring-srv6-network-programming]. SRv6 is a proposed standard and is deployed today.

The SPRING working group has observed that some use cases, such as strict path TE, may require long SRv6 SID lists. There are several proposed methods to reduce the resulting SRv6 encapsulation size by compressing the SID list.

The SPRING working group formed a design team to define requirements for, and analyze proposals to, compress SRv6 SID lists.

It is a goal of the design team to identify solutions to SRv6 SID list compression that are based on the SRv6 standards. As such, this document provides requirements for SRv6 SID list compression solutions that utilize the existing SRv6 data plane and control plane.

It is also a goal of the design team to consider proposals that are not based on the SRv6 data plane and control plane. As such, this document includes requirements to evaluate whether a compression proposal provides all the functionality of SRv6 (section "SRv6 Functionality") in addition to satisfying compression specific requirements.

For each requirement, a description, rationale and metrics are described.

The design team will produce a separate document to analyze the proposals.

This document is a draft; additional requirements are under review, additional requirements will be added, and current requirements may change. Appendix A contains a subset of requirements without unanimous consensus. Additional requirements without unanimous consensus are not in the appendix.

2. Conventions used in this document

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.

2.2. Terminology

SR: Segment Routing

SRH: Segment Routing Header

MPLS: Multiprotocol Label Switching

SR-MPLS: Segment Routing over MPLS data plane

SID: Segment Identifier

SRv6: Segment Routing over IPv6

SRv6 SID List: A list of SRv6 SIDs

Compression proposal: A proposal to compress SRv6 SID lists

SRv6 base: SRv6 as defined in [RFC8402], [RFC8754], [I-D.ietf-spring-srv6-network-programming]

SID numbering space: may be implemented as

- a single IGP instance
- a single IGP level or area
- two or more autonomous systems that coordinate SID numbering space
- two or more IGP instances that coordinate SID numbering space

SRv6 Encapsulation Header: The IPv6 header, and any extension headers preceding a payload, used to implement a SRv6 base or compression proposal.

3. SRv6 SID List Compression Requirements

3.1. Dataplane Efficiency and Performance Requirements

3.1.1. Encapsulation Header Size

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

Rationale: A smaller SRv6 encapsulation results in better MTU efficiency.

Metric: Compression is the ratio of the IPv6 encapsulation size of SRv6 as defined in [RFC8402], [RFC8754], [I-D.ietf-spring-srv6-network-programming] vs the IPv6 encapsulation size of a given proposal. The encapsulation savings of a compression proposal vs the SRv6 base is a useful measurement to compare proposals.

The encapsulation metric (E) records the number of bytes required for a proposal to encapsulate a packet given a specific segment list.

- E(proposal, segment list).

The encapsulation savings(ES)records the encapsulation savings for a proposal to encapsulate a packet given a specific segment list.

- ES(proposal, segment list) = 1 - E(proposal, segment list)/E(SRv6 base, segment list).

3.1.2. Forwarding Efficiency

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

Rationale: Efficiency in bits on the wire and processing efficiency are both important. Optimizing one at the expense of the other may lead to significant performance impact.

Metric: The data plane efficiency metric (D) records the data plane forwarding efficiency of the proposed solution. Two metrics are used and recorded at each segment endpoint:

- D.PRS(segment list): number of headers parsed during processing of the segment list, starting from and including the IPv6 header.
- D.LKU(segment list): number of FIB lookups during processing of the segment list. The type of lookup is also recorded as longest prefix match (LPM) or exact match (EM)

3.1.3. State Efficiency

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

Rationale: Additional state increases the complexity of the control plane and data plane. It can also result in an increase in memory usage.

Metric: The state efficiency metric (S) records the amount of additional forwarding state required by the proposed solution.

- S(node parameters): the number of additional forwarding states that need to be stored at a node, given a set of node parameters consisting of the number of nodes in the network, number of local interfaces, number of adjacencies. The forwarding state is counted as entries required in a Forwarding Information Base (FIB) at a node.

4. SRv6 Specific Requirements

4.1. SRv6 Based

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

Rationale: A compression proposal built on existing IETF standards is preferable to creating new standards with equivalent functionality and performance.

Metric: The utilization metric (U) records whether a proposal utilizes the SRv6 specifications.

Utilization is recorded in a table, with a column per proposal and rows consisting of the following metrics:

- U.RFC8402: utilizes [RFC8402].
- U.RFC8754: utilizes [RFC8754].
- U.PGM: utilizes [I-D.ietf-spring-srv6-network-programming].
- U.IGP: utilizes [I-D.ietf-lsr-isis-srv6-extensions].
- U.BGP: utilizes [I-D.ietf-bess-srv6-services].
- U.POL: utilizes [I-D.ietf-spring-segment-routing-policy].
- U.BLS: utilizes [I-D.ietf-idr-bgpls-srv6-ext].
- U.SVC: utilizes [I-D.ietf-spring-sr-service-programming].
- U.OAM: utilizes [I-D.ietf-6man-spring-srv6-oam].
- U.ALG: utilizes [I-D.ietf-lsr-flex-algo].

Each cell contains "yes" for utilizes, or "no" for does not utilize.

4.2. Functional Requirements

4.2.1. SRv6 Functionality

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

Rationale: Operators require SRv6 functionality. Evaluating the extent to which a proposal supports SRv6 functionality is important for operators and implementors to understand the impact on network operations.

Metric: The Functionality metric (F) records whether a proposal supports SRv6 functionality and how the functionality is provided.

Functionality is recorded in a table with columns for each proposal and rows consisting of the following metrics:

- F.SID: Supports SRv6 SID functionality as described in [RFC8402]
- F.SCOPE: Supports globally and locally scoped SID functionality as described in [RFC8402]
- F.PFX: Supports prefix SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.ADJ: Supports adjacency SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.BIND: Supports binding SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.PEER: Supports BGP peering SID functionality as described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.SVC: Supports L3 and L2 VPN service SID functionality as described in [I-D.ietf-spring-srv6-network-programming]
- F.ALG: Supports flexible algorithms functionality as described in [I-D.ietf-lsr-flex-algo]
- F.TILFA: Supports TI-LFA functionality as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- F.SEC: Supports securing an SR domain with ingress filtering as functionally defined in [RFC8754]
- F.IGP: Supports distributing topological SIDs and behaviors via ISIS as functionally described in [I-D.ietf-lsr-isis-srv6-extensions]
- F.BGP: Supports BGP VPNs as functionally described in [I-D.ietf-bess-srv6-services]

- o F.POL: Supports SR policies and steering traffic over those policies as funcitonally described in [I-D.ietf-spring-segment-routing-policy]
- o F.BLS: Supports Link State distribution via BGP as functionally described in [I-D.ietf-idr-bgpls-srv6-ext]
- o F.SFC: Supports stateless service programming as functionally described in [I-D.ietf-spring-sr-service-programming]
- o F.PING: Supports pinging a SID to verify the SID is implemented as functionally described in [I-D.ietf-6man-spring-srv6-oam]

Each cell contains the specification name documenting the functionality.

4.2.2. Heterogeneous SID lists

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

Rationale: Support of SID lists with compressed and non-compressed SIDs reduces encapsulation size when not all SRv6 nodes deploy the compression proposal or 128-bit SIDs are required.

Metric: A compliant compression proposal supports both:

- o classic 128-bit SRv6 SIDs in the IPv6 Destination Address field
- o segment lists (i.e., paths) with both compressed and 128-bit SRv6 SIDs.

4.2.3. SID list length

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

Rationale: Strict TE paths require SID list lengths proportional to the diameter of the SR domain.

Metric: The compression proposal must be able to steer a packet through an SR path that contains up to sixteen segments.

4.2.4. SID summarization

Description: The solution MUST be compatible with segment summarization.

Rationale: Summarization of segments is a key benefit of SRv6 vs SR MPLS. In interdomain deployments, any node can reach any other node via a single prefix segment. Without summarization, border router SIDs must be leaked, and an additional global prefix segment is required for each domain border to be traversed.

Metric: A solution supports summarization when segments can be summarized for advertisement into other IGP domains or levels.

4.3. Operational Requirements

4.3.1. Lossless Compression

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

Rationale: In SRv6, we can represent a path to meet certain objectives. A compression proposal needs to support the objectives with the same path.

Metric: Information present in the pre-compression segment list MUST also be present in the post-compression SID list.

4.3.2. Preservation of non-routing information

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

Rationale: SRv6 ingress nodes encode non-routing information in the IPv6 header chain. This information can be encoded in the following fields:

- Hop Count
- DSCP bits
- ECN bits
- Flow label
- HBH Options Extension header
- Fragment Extension header
- Authentication Extension header
- Encrypted Security Payload Extension header
- Destination Options Extension header

Some of these fields are mutable (e.g., Hop Count) while others are immutable (e.g., Fragment Extension Header).

Some of these fields contain information that is required by every node along a packet's delivery path (e.g., Hop Count). Others contain information that is required only by the packet's ultimate destination (e.g., Fragment Extension Header).

Therefore, the compression mechanism MUST NOT prevent this information from being delivered, in an IPv6 header chain, to any node that needs it.

Metric: The SR source node encapsulates its payload (e.g.., Ethernet, IP, TCP) in an IPv6 header. The SRv6 header contains both routing and non-routing information. The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node.

4.3.3. Address Planning

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

Rationale: The address planning of the network may vary based on the addressing scheme of the operator, so the solution MUST support a flexible addressing scheme. Operators need to deploy the solution based on their own address planning.

Metric: The compression proposal supports locators drawn from different prefixes with the solutions analysis indicating efficiency.

4.4. Scalability Requirements

4.4.1. Adjacency segment scale

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

Rationale: Typically, network operators deploy networks with tens or hundreds of adjacency segments per node, but some network operators may deploy networks that use more adjacency segments per node.

Metric: A proposal that allows 65000 adjacency segments per node satisfies this requirement.

4.4.2. Prefix segment scale

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

Rationale: Typically, network operators deploy networks with thousands of prefix segments per SID numbering space, but some network operators may deploy networks that use more prefix segments per SID numbering space.

Metric: A proposal that allows 1 million prefix segments per SID numbering space satisfies this requirement.

4.4.3. Service Scale

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

Rationale: Typically, network operators deploy networks with tens to hundreds of thousands of services per node, but some network operators may deploy networks that use more services per node.

Metric: A proposal that allows 1 million services per node satisfies this requirement.

4.4.4. Compression Levels

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

Rationale: The compression proposal will be deployed in networks of varying size with SID numbering spaces of varying size. Network and service scale can directly impact SID length and the ability of a proposal to compress the SID list.

Metric: A compression proposal that supports relatively better compression for smaller SID numbering spaces and service scale satisfies this requirement.

5. Protocol Design Requirements

5.1. SRv6 Base Coexistence

Description: The compression proposal MUST support deployment in existing SRv6 networks.

Rationale: SRv6 is deployed today. A compression proposal that interoperates well with SRv6, as deployed, will reduce the overhead

and simplify operations. For Network operators who would migrate to compressed SRv6 SID lists, the migration is expected to gradually occur over a period of time as they upgrade networks, domains, device families and software instances.

Metric: A compliant compression proposal provides the following

- Supports simultaneous deployment at a node with current SRv6 SIDs.
- Supports simultaneous deployment at a node with current SRv6 control plane.
- Supports simultaneous operation of current SRv6 paths with compressed paths.
- Supports the behaviors in [I-D.ietf-spring-srv6-network-programming].
- Does not require removal of existing IPv6 address planning.

6. Security Requirements

6.1. Security Mechanisms

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

Rationale: It is important to identify security issues and how to address them in any specification.

Metric: A compression solution that does not introduce unresolved security issues meets this requirement.

6.2. SR Domain Protection

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

Rationale: The unauthorized use of SIDs within the SR domain by nodes outside the domain can disrupt an operators' network.

Metric: A compliant solution describes how access to SIDs within the SR domain is denied to nodes outside the SR domain.

7. IANA Considerations

This document has no requests to IANA.

8. Security Considerations

TBD

9. Contributors

The following individuals contributed to this draft

Sanders Steffann, SJM Steffann Consultancy, sander@steffann.nl

10. Normative References

[I-D.ietf-6man-spring-srv6-oam]
Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-08 (work in progress), October 2020.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay services", draft-ietf-bess-srv6-services-05 (work in progress), November 2020.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-05 (work in progress), November 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-lsr-isis-srv6-extensions]
Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-11 (work in progress), October 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-05 (work in progress), November 2020.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

Appendix A. Proposed Requirements

This appendix contains requirements that the design team discussed but could not be agreed upon.

A.1. IPv6 Based

Description: The compression mechanism requires every node along the packet's delivery path to be IPv6-capable. It MUST not require any

node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

Rational: According to RFC 8402, SRv6 is an instantiation of the SR Architecture over the IPv6 data plane.

Metric: A compliant solution requires every node along the packet's delivery path to be IPv6-capable. It does not not require any node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

A.2. Point to Multipoint

Description: The compression mechanism SHOULD support point-to-multipoint SR paths.

Rationale: Many VPN services require point-to-multipoint SR paths.

Metric: A compliant proposal can encode a multicast address in the ultimate segment of the segment list.

A.3. Parsability

Description: The compression mechanism MUST be parsable. That is, the node that consumes the compressed SID list must be able to decode the active and next segment. Parsing information MAY be conveyed in either the forwarding or control plane.

Rationale: Failure to parse the compressed SID list leads to undesired behaviors.

Metric: In the nominal case the producer and consumer of the SID list agree on the active segment and next segment. In forseeable failure modes it is possible to determine why the producer and consumer don't agree.

Authors' Addresses

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Chongfeng Xie
China Telecom

Email: xiechf@chinatelecom.cn

Ron Bonica
Juniper

Email: rbonica@juniper.net

Darren Dukes
Cisco Systems

Email: ddukes@cisco.com

Cheng Li
Huawei

Email: c.l@huawei.com

Peng Shaofu
ZTE

Email: peng.shaofu@zte.com.cn

Wim Henderickx
Nokia

Email: wim.henderickx@nokia.com

DetNet Working Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

Y(J) Stein
RAD

February 22, 2021

Segment Routed Time Sensitive Networking
draft-stein-srtsn-00

Abstract

Routers perform two distinct user-plane functionalities, namely forwarding (where the packet should be sent) and scheduling (when the packet should be sent). One forwarding paradigm is segment routing, in which forwarding instructions are encoded in the packet in a stack data structure, rather than programmed into the routers. Time Sensitive Networking and Deterministic Networking provide several mechanisms for scheduling under the assumption that routers are time synchronized. The most effective mechanisms for delay minimization involve per-flow resource allocation.

SRTSN is a unified approach to forwarding and scheduling that uses a single stack data structure. Each stack entry consists of a forwarding portion (e.g., IP addresses or suffixes) and a scheduling portion (deadline by which the packet must exit the router). SRTSN thus fully implements network programming for time sensitive flows, by prescribing to each router both to-where and by-when each packet should be sent.

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 August 26, 2021.

1. Introduction

Packet Switched Networks (PSNs) use statistical multiplexing to fully exploit link data rate. On the other hand, statistical multiplexing in general leads to end-to-end propagation latencies significantly higher than the minimum physically possible, due to packets needing to reside in queues waiting for their turn to be transmitted.

Recently Time Sensitive Networking (TSN) and Deterministic Networking (DetNet) technologies have been developed to reduce this queueing latency for time sensitive packets [RFC8557]. Novel TSN mechanisms are predicated on the time synchronization of all forwarding elements (Ethernet switches, MPLS Label Switched Routers, or IP routers, to be called here simply routers). Once routers agree on time to high accuracy, it is theoretically possible to arrange for time sensitive packets to experience "green waves", that is, never to wait in queues. For example, scheduling timeslots for particular flows eliminates packet interference, but eliminates the statistical multiplexing advantage of PSNs. In addition, the scheduling calculation and programming of the network to follow this calculation doesn't scale well to large networks.

Segment Routing (SR) technologies provide a scalable method of network programming, but until now has not been applied to scheduling. The SR instructions are contained within a packet in the form of a first-in first-out stack dictating the forwarding decisions of successive routers. Segment routing may be used to choose a path sufficiently short to be capable of providing sufficiently low end-to-end latency but does not influence the queueing of individual packets in each router along that path.

2. Forwarding and Scheduling

Routers (recall that by routers we mean any packet forwarding device) perform two distinct functions on incoming packets, namely forwarding and scheduling. By forwarding we mean obtaining the incoming packet, inspecting the packet's headers, deciding on an output port, and for QoS routing a specific output queue belonging to this output port, based on the header information and a forwarding information base, optionally editing the packet (e.g., decrementing the TTL field or performing a stack operation on a MPLS label), and placing the packet into the selected output queue.

Scheduling consists of selecting which output queue and which packet from that output queue will be the next packet to be physically transmitted over the output port. In simple terms one can think of forwarding and scheduling as "which output port" and "which packet" decisions, respectively; that is, forwarding decides to which output port to send each packet, and scheduling decides which packet to send next.

Segment routing (as well as connection-oriented mechanisms) slightly simplify the meaning of forwarding to deciding "where" to send the incoming packet, while TSN slightly simplifies the meaning of scheduling to deciding "when" to send the outgoing packet.

Routers optionally perform a third user plane operation, namely per output port and/or per flow traffic conditioning. By conditioning we mean policing (discarding packets based on a token bucket algorithm), shaping (delaying packets), (W)RED, etc. Since we will only be interested in per-packet per router behavior we will neglect conditioning, which is either per router (not distinguishing between packets) or per flow (the same for all routers along the path).

As aforementioned, forwarding decisions always select an output port, but when there are QoS criteria additionally decide on an output queue belonging to that port. The use of multiple queues per output port is to aid the scheduling, which then becomes a matter of selecting an output queue and always taking the packet at the end of the queue (the packet that has waited the longest). For example, the simplest nontrivial scheduling algorithm is "strict priority". In strict priority packets are assigned to queues according to their priority (as indicated by Priority Code Point or DiffServ Code Point field). The strict priority scheduler always first checks the queue with highest priority; if there is a packet waiting there it is selected for transmission, if not the next highest priority queue is examined and so on. Undesirably strict priority may never reach packets in low priority queues (Best Effort packets), so alternative

algorithms, e.g., Weighted Fair Queueing, are used to select from priority queues more fairly.

TSN is required for networks transporting time sensitive traffic, that is, packets that are required to be delivered to their final destination by a given time. In the following we will call the time a packet is sent by the end user application (or the time it enters a specific network) the "birth time", the required delivery time to the end-user application (or the time it exists a specific network) the "final deadline" and the difference between these two times (i.e., the maximally allowed end-to-end propagation time though the network) the "delay budget".

Unlike strict priority or WFQ algorithms, TSN scheduling algorithms may directly utilize the current time of day. For example, in the TSN scheduling algorithm known as time-aware scheduling (gating), each output queue is controlled by a timed gate. At every time only certain output queues have their gates "open" and can have their packets scheduled, while packets are not scheduled from queues with "closed" gates. By appropriately timing the opening and closing of gates of all routers throughout the network, packets in time sensitive flows may be able to traverse their end-to-end path without ever needlessly waiting in output queues. In fact, time-aware gating may be able to provide a guaranteed upper bound for end-to-end delay.

However, time-aware scheduling suffers from two major disadvantages. First, opening the gates of only certain queues for a given time duration, results in this time duration being reserved even if there are very few or even no packets in the corresponding queues. This is precisely the undesirable characteristic of Time Division Multiplexing networks that led to their replacement by Packet Switched Networks. Minimizing time durations increases efficiency, but at the cost of obliging a time sensitive packet that just missed its gate to wait until the next gate opening, endangering its conforming to the delay budget.

In order to avoid such problems, one needs to know a priori the birth times of all time sensitive packets, the lengths of all links between routers, and the loading of all routers. Based on this input one can calculate optimal gating schedules for all routers in the network and distribute this information to all the routers. This calculation is computationally expensive and updating all the routers is communicationally expensive. Moreover, admitting a new time-sensitive flow requires recalculation of all the gating schedules and updating all the routers. This recalculation and communications load is practical only for small networks and a relatively small numbers of flows.

3. Stack-based Methods for Latency Control

One can envision mechanisms for reducing end-to-end propagation latency in a network with time-synchronized routers that do not suffer from the disadvantages of time sensitive scheduling. One such mechanism would be to insert the packet's birth time (time created by end-user application or time entering the network) into the packet's headers. Each router along the way could use this birth time by prioritizing packets with earlier birth times, a policy known as Longest in System (LIS). These times are directly comparable, due to our assuming the synchronization of all routers in the network. This mechanism may indeed lower the propagation delay, but at each router the decision is sub-optimal since a packet that has been in the network longer but that has a longer application delay budget will be sent before a later packet with a tighter delay budget.

An improved mechanism would insert into the packet headers the desired final deadline, i.e., the birth time plus the delay budget. Each router along the way could use this final destination time by prioritizing packets with earlier deadlines, a policy known as Earliest Deadline First (EDF). This mechanism may indeed lower the propagation delay, but at each router the decision is sub-optimal since a packet that has been in the network longer but is close to its destination will be transmitted before a later packet which still has a long way to travel.

A better solution to the problem involves precalculating individual "local" deadlines for each router, and each router prioritizing packets according to its own local deadline. As an example, a packet sent at time 10:11:12.000 with delay budget of 32 milliseconds (i.e., final deadline time of 10:11:12.032) and that needs to traverse three routers might have in its packet headers three local deadlines, 10:11:12:010, 10:11:12.020, and 10:11:12.030. The first router employs EDF using the first local deadline, the second router similarly using the second local deadline, and the ultimate router using the last local deadline.

The most efficient data structure for inserting local deadlines into the headers is a "stack", similar to that used in Segment Routing to carry forwarding instructions. The number of deadline values in the stack equals the number of routers the packet needs to traverse in the network, and each deadline value corresponds to a specific router. The Top-of-Stack (ToS) corresponds to the first router's deadline while the Bottom-of-Stack (BoS) refers to the last's. All local deadlines in the stack are later or equal to the current time (upon which all routers agree), and times closer to the ToS are always earlier or equal to times closer to the BoS.

The stack may be dynamic (as is the forwarding instruction stack in SR-MPLS) or static with an index (as is the forwarding instruction stack in SRv6).

For private networks it is possible for the stack to be inserted by the user equipment that is the source of the packet, in which case the top of stack local deadline corresponds to the first router to be encountered by the packet. However, in such a case the user equipment must also by time synchronized for its time values to be directly compatible. In an improved strategy the stack is inserted into the packet by the ingress router, and thus its deadlines are in concert with time in the network. In such case the first deadline will not explicitly appear in the stack and the initial ToS corresponds to the second router in the network to be traversed by the packet. In either case each router in turn pops from the stack the ToS local deadline and uses that local deadline in its scheduling (e.g., employing EDF).

Since the ingress router inserts the deadline stack into the packet headers, no other router needs to be aware of the requirements of the time sensitive flows. Hence admitting a new flow only requires updating the information base of the ingress router. In an efficient implementation the ingress router's information base has deadline offset vectors for each time sensitive flow. Upon receipt of a packet from user equipment, the ingress router first determines if the packet belongs to a time sensitive flow. If so, it adds the current time to the deadline offset vector belonging to the flow and inserts it as a stack into the packet headers.

An explicit example is depicted in Figure 1. Here packets of a specific time sensitive flow are required to be received by the remote user equipment within 200 microseconds of being transmitted by the source user equipment. The packets traverse a wireless link with delay 2 microseconds to reach the router R1 (the ingress router). They then travel to router R2 over an optical fiber experiencing a propagation delay of 18 microseconds, from there to router R3 experiencing an additional 38 microseconds of fiber delay, from there to router R4 (the egress router) experiencing 16 microseconds of fiber delay. Finally, they travel over a final wireless link taking again 2 microseconds.

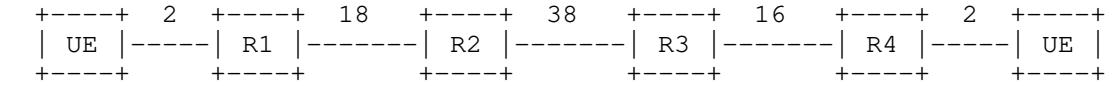


Figure 1: Example with propagation latencies

We conclude that the total constant physical propagation time is 2+18+38+16+2=76 microseconds. Moreover, assume that we know that in each router there is an additional constant time of 1 microsecond to receive the packet at the line rate and 5 microseconds to process the packet, that is, 6 microseconds per router or 24 microseconds for all four routers. We have thus reached the conclusion that the minimal time to traverse the network is 76+24=100 microseconds

Since our delay budget is 200 microseconds, we have spare time of 200-100=100 microseconds for the packets to wait in output queues. If we have no further information, we can divide this spare 100 microseconds equally among the 4 routers, i.e., 25 microseconds per router. Thus, the packet arrives at the first router after 2 microseconds, is received and processed after 2+6=8 microseconds, and is assigned a local deadline to exit the first router of 8+25=33 microseconds. The worst case times of arrival and transmission at each point along the path are depicted in Figure 2. Note that in general it may be optimal to divide the spare time in unequal fashion.

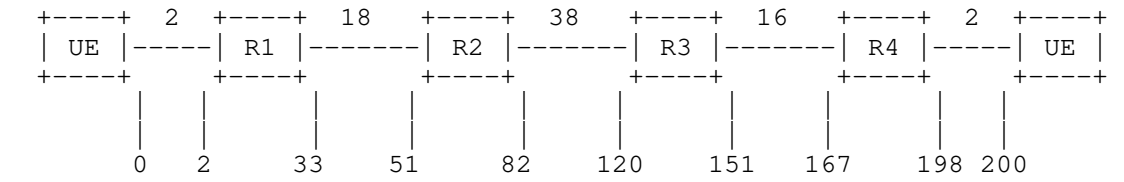


Figure 2: Example with worst case times

Assuming that the packet left router 1 the full 33 microseconds after its transmission, it will arrive at router 2 after an additional 18 microseconds, that is, after 51 microseconds. After the mandatory 6 microseconds of reception and processing and the 25 microseconds allocated for queueing, we reach the local deadline to exit router 2 by 82 microseconds. Similarly, the local deadline to exit router 3 is 151 microseconds, and the deadline to exit router 4 is 198 microseconds. After the final 2 microseconds consumed by the wireless link the packet will arrive at its destination after 200 microseconds as required

Based on these worst case times the ingress router can now build the deadline offset vector (33, 82, 151, 198) referenced to the time the packet left the source user equipment, or referenced to the time the packet arrives at the ingress router of (31, 80, 149, 196).

Now assume that a packet was transmitted at time T and hence arrives at the ingress router at time T + 2 microseconds. The ingress router R1, observing the deadline offset vector referenced to this time,

knows that the packet must be released no more than 31 microseconds later, i.e., by T + 33 microseconds. It furthermore inserts a local deadline stack [T+82, T+151, T+198] into the packet headers.

The second router R2 receives the packet with the local deadline stack, pops the ToS revealing that it must ensure that the packet exits by T + 82 microseconds. It properly prioritizes and sends the packet with the new stack [T+151, T+198]. Router R3 pops deadline T+151, and sends the packet with local deadline stack containing a single entry [T+198]. The final router pops this final local deadline and ensures that the packet is transmitted before that time The local deadline stacks are depicted in Figure 3.

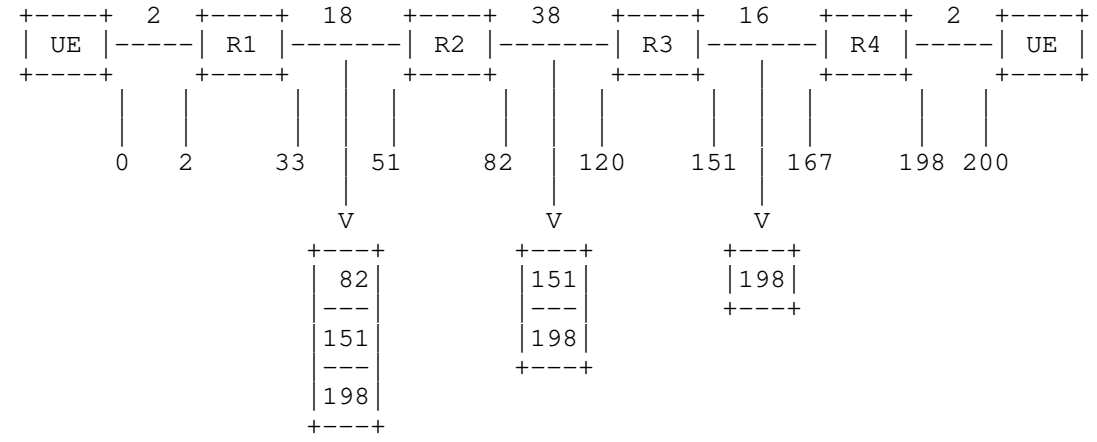


Figure 3: Example with local deadline stacks

The precise mechanism just described is by no means the only way to compute local deadlines. Furthermore, combining time-aware scheduling at the ingress router only with EDF at all the other routers can provide "green waves" with provable upper bounds to delay. However, optimizing such a scheme at scale is a challenge. A randomized algorithm for computation of the deadline offset vector is described in [AndrewsZhang].

4. The Time Sensitive Router

While a stack is the ideal data structure to hold the local deadlines in the packet, different data structures are used to hold the time sensitive packets (or their descriptors) in the routers. The standard data structure used in routers is the queue which, being a first in first out memory, is suitable for a policy of first-to-arrive first-to-exit, and not for EDF or other stack-based time sensitive mechanisms. More suitable data structures are sorted

lists, search trees, and priority heaps. While such data structures are novel in this context, efficient hardware implementations exist.

If all the time sensitive flows are of the same priority, then a single such data structure may be used for all time sensitive flows. If there are time sensitive flows of differing priorities, then a separate such data structure is required for each level of priority corresponding to a time sensitive flow, while the conventional queue data structure may be used for priority levels corresponding to flows that are not time sensitive.

For example, assume two different priorities of time sensitive flows and a lower priority for Best Effort traffic that is not time sensitive. If applying strict priority the scheduler would first check if the data structure for the highest priority contains any packets. If yes, it transmits the packet with the earliest local deadline. If not, it checks the data structure for the second priority. If it contains any packets it transmits the packet with the earliest deadline. If not, it checks the Best Effort queue. If this queue is nonempty it transmits the next packet in the queue, i.e., the packet that has waited in this queue the longest.

Separate prioritization and EDF is not necessarily the optimal strategy. An alternative (which we call Liberal EDF, or LEDF) would be for the scheduler to define a worst case (i.e., maximal) packet transmission time MAXTT (for example, the time taken for a 1500 byte packet to be transmitted at the output port's line rate). Instead of checking whether the data structure for the highest priority contains any packets at all, LEDF checks whether its earliest packet's local deadline is earlier than MAXTT from the current time. If it is, it is transmitted; if it is not the next priority is checked, knowing that even were a maximal size packet to be transmitted the scheduler will still be able to return to the higher priority packet before its local deadline.

5. Segment Routed Time Sensitive Networking

Since Segment Routing and the TSN mechanism just described both utilize stack data structures it is advantageous to combine their information into a single unified SRTSN stack. Each entry in this stack contain two subentries, the forwarding instruction (e.g., the address of the next router or the label specifying the next link) and a scheduling instruction (the local deadline).

Each SRTSN stack entry fully prescribes the forwarding and scheduling behavior of the corresponding router, both to-where and by-when the packet should be sent. The insertion of a stack into packets thus fully implements network programming for time sensitive flows.

For example, Figure 4 depicts the previous example but with the unified SRTSN stacks. Ingress router R1 inserts a SRTSN stack with three entries into the packet received. In this example the forwarding sub-entry contains the identifier or address of the next router, except for the Bottom of Stack entry that contains a special BoS code (e.g., identifier zero). The ToS entry thus contains the address of router R3 and the time by which the packet must exit router R2, namely T + 82 microseconds. Router R2 pops this ToS leaving a SRTSN stack with 2 entries. Router R3 pops the new ToS instructing it to forward the packet to router R4 by time T + 151 microseconds, leaving a stack with a single entry. Router R4 pops the ToS and sees that it has reached bottom of stack. It then forwards the packet according to the usual rules of the network (for example, according to the IP address in the IP header) by local deadline T + 198 microseconds.

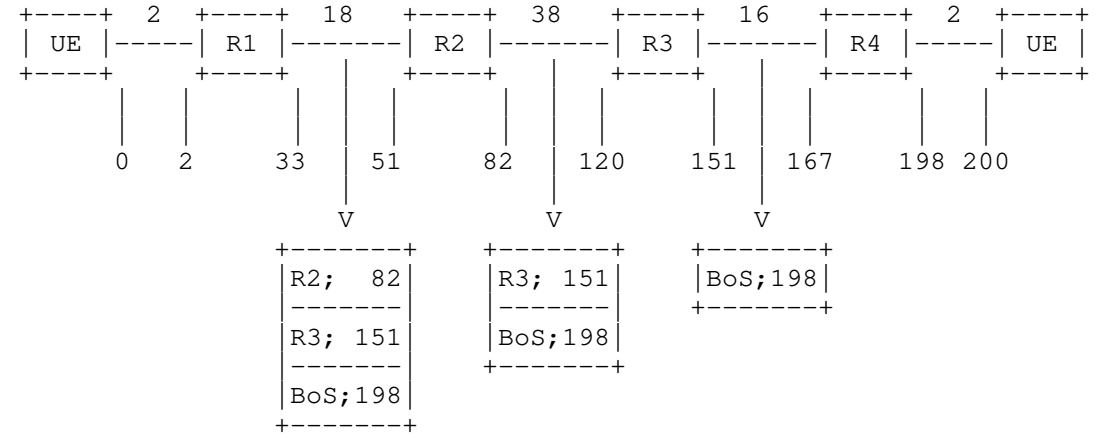


Figure 4: Example with combined SRTSN stacks

6. Stack Entry Format

A number of different time formats are in common use in networking applications and can be used to encode the local deadlines. The longest commonly utilized format is 80-bit PTP-80 timestamp defined in IEEE 1588v2 Precision Time Protocol [IEEE1588]. There are two common 64-bit time representations: the NTP-64 timestamp defined in [RFC5905] (32 bits for whole seconds and 32 bits for fractional seconds); and the PTP-64 timestamp (32 bits for whole seconds and 32 bits for nanoseconds). Finally, there is the NTP-32 timestamp (16 bits of whole seconds and 16 bits of fractional seconds) that is often insufficient due to its low resolution (15 microseconds).

However, we needn't be constrained by these common formats, since our wraparound requirements are minimal. As long as we have no ambiguity in times during the flight of a packet, which is usually much less than a second, the timestamp is acceptable. Thus, we can readily use a nonstandard 32-bit timestamp format with say 12 bits of seconds (wraparound over 1 hour) and 20 bits for microseconds, or say 8 bits for whole seconds (wraparound over 4 minutes) and 24 bits of tenths of microseconds.

For the forwarding sub-entry we could adopt like SR-MPLS standard 32-bit MPLS labels (which contain a 20-bit label and BoS bit), and thus SRTSN stack entries could be 64-bits in size comprising a 32-bit MPLS label and the aforementioned nonstandard 32-bit timestamp. Alternatively, an SR-TSN stack entry could be 96 bits in length comprising a 32-bit MPLS label and either of the standardized 64-bit timestamps.

For IPv4 networks one could employ a 32-bit IPv4 address in place of the MPLS label. Thus, using the nonstandard 32-bit timestamp the entire stack entry could be 64 bits. For dynamic stack implementations a BoS bit would have to be included.

SRv6 uses 128-bit IPv6 addresses (in addition to a 64-bit header and possibly options), and so 160-bit or 192-bit unified entries are directly derivable. However, when the routers involved are in the same network, address suffixes suffice to uniquely determine the next router.

7. Control Plane

In the above discussion we assumed that the ingress router knows the deadline offset vector for each time sensitive flow. This vector may be calculated by a centralized management system and sent to the ingress router, or may be calculated by the ingress router itself.

In the former case there is central SRTSN orchestrator, which may be based on a Network Management System, or on an SDN controller, or on a Path Computation Element server. The SRTSN orchestrator needs to be know the propagation delays for all the links in the network, which may be determined using time domain reflectometry, or via one-way delay measurement OAM, or retrieved from a network planning system. The orchestrator may additionally know basic parameters of the routers, including minimal residence time, data rate of the ports, etc. When a time sensitive path needs to be set up, the SRTSN orchestrator is given the source and destination and the delay budget. It first determines feasibility by finding the end-to-end delay of the shortest path (shortest being defined in terms of latency, not hop count). It then selects a path (usually, but not

necessarily, the shortest one) and calculates the deadline offset vector. The forwarding instructions and offset vector (as well as any other required flow-based information, such as data rate or drop precedence) are then sent to the ingress router. As in segment routing, no other router in the network needs to be informed.

In the latter case the ingress router is given the destination and the delay budget. It sends a setup message to the destination as in RSVP-TE, however, in this case arrival and departure timestamps are recorded for every router along the way. The egress router returns the router addresses and timestamps. This process may be repeated several times and minimum gating applied to approximate the link propagation times. Assuming that the path's delay does not exceed the delay budget, the path and deadline offset vector may then be determined.

The method of [AndrewsZhang] uses randomization in order to avoid the need for centralized coordination of flows entering the network at different ingress routers. However, this advantage comes at the expense of much higher achievable delay budgets.

8. Security Considerations

SRTSN concentrates the entire network programming semantics into a single stack, and thus tampering with this stack would have devastating consequences. Since each stack entry must be readable by the corresponding router, encrypting the stack would necessitate key distribution between the ingress router and every router along the path.

A simpler mechanism would be for the ingress router to sign the stack with a public key known to all routers in the network, and to append this signature to the stack. If the signature is not present or is incorrect the packet should be discarded.

9. IANA Considerations

This document requires no IANA actions.

10. Informative References

[AndrewsZhang]
Andrews, M. and L. Zhang, "Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule", Journal of Algorithms 52 57-81, 2003.

[IEEE1588] IEEE, "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE 1588-2008, DOI 10.1109/IEEESTD.2008.4579760, 2008.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010.

[RFC8557] Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", RFC 8557, DOI 10.17487/RFC8557, May 2019.

Author's Address

Yaakov (J) Stein
RAD

Email: yaakov_s@rad.com

RTGWG Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

F. Yang
M. Chen
T. Zhou
Huawei Technologies
February 22, 2021

Associated Channel over IPv6
draft-yang-rtgwg-ipv6-associated-channel-00

Abstract

In this document, an associated channel is introduced to provide a control channel based on IPv6, carrying types of control and management messages. By using the associated channel, messages can be transmitted between the network nodes to provide functions like path identification, OAM, protection switchover signaling, etc., targeting to provide high quality SLA guarantee to service.

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 August 26, 2021.

Table of Contents

1. Introduction

IPv6 is becoming widely accepted to provide the connectivity in many new emerging scenarios, including Cloud Network convergence, Cloud-Cloud interconnection, 5G vertical industries, Internet of Things, as well as the legacy networks migrating towards SR over IPv6. However, IP packet is locally lookup, and forwarded hop by hop without aware of the forwarding path. Path segment over SRv6 [I-D.ietf-spring-srv6-path-segment] provides a good solution to identify an SR path over IPv6, but can only be applicable in source routing paradigm.

To identify an IPv6 forwarding path, further to better control and manage the path, this document introduces an associated channel based on IPv6, intenting to create a control channel for the control and management usages. By using the associated channel, messages can be transmitted between the network nodes to provide functions like path identification, OAM, protection switchover signaling, etc., targeting to provide high quality SLA guarantee to service.

This document also defines a TLV format for the associated channel and how it can be encapsulated in IPv6 packet, and the potential applicability in IPv6 networks. Applications of associated channel in IPv6 shall be specified in different documents and thus are out of scope of this document.

2. Terminology

OAM: Operations, Administration, and Maintenance

SLA: Service Level Agreement

ACH: Associated CHannel

3. Associated Channel

An associated channel provides a control channel that carries at least one or more types of control and management messages. The type of message is not limited to any specific usage. The associated channel is specified by two parts of information, including the identification of associated channel and the carried message.

3.1. Identification of Associated Channel

The identification of associated channel indicates the path where the packets of associated channel are transmitted on. This identification also indicates the same path of the service forwarding path which the associated channel is associated to.

3.2. ACH TLV to Carry Message

An Associated CHannel (ACH) TLV is designed to carry the message of an associated channel. ACH TLV has the following format:

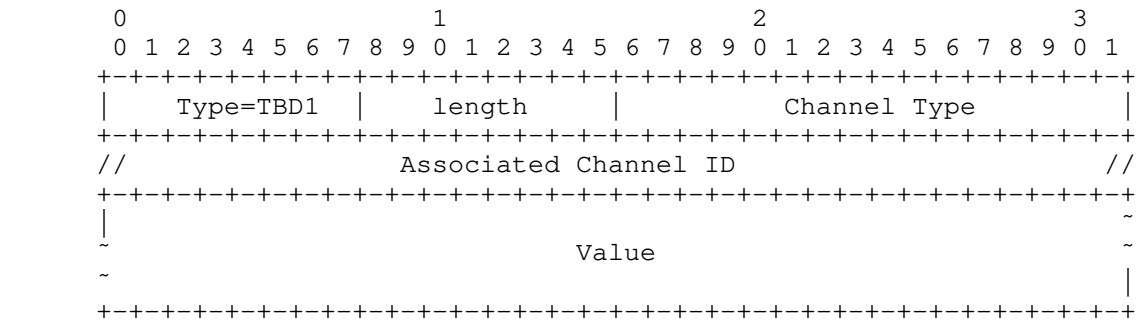


Figure 1 ACH TLV Format

Type: 8 bits, indicates it is an associated channel (ACH) TLV, and request a value assigned by IANA. The uniform type of TLV generalizes the applicability of ACH TLV to support various types of messages.

Length: 8 bits, defines the length of Value field in bytes.

Channel Type: is a 16-bit-length fixed portion as a part of Value field. It indicates the specific type of messages carried in associated channel. Note that a new ACH TLV Channel Type Registry would be requested to IANA. In the later documents which specify application protocols of associated channel, MUST also specify the applicable Channel Type field value assigned by IANA.

Associated Channel ID: indicates the identification of associated channel. The length is TBD.

Value: is a variable part of Value field. It specifies the messages indicated by Channel Type and carried in associated channel. Note that the Value field of ACH TLV MAY contain sub-TLVs to provide additional context information to ACH TLV.

3.3. Encapsulation of ACH TLV in IPv6

In the context of IPv6, ACH TLV can be encapsulated in different types of IPv6 extension header or even IPv6 payload. Note that, no matter which way ACH TLV is applied, there is no semantic change to IPv6 extension headers. Moreover, ACH TLV can be carried either with user data in an in-situ way, or in a independent synthetic packet.

3.3.1. Encapsulated in IPv6 Destination Options Header

ACH TLV can be encapsulated in IPv6 Destination Options Header as the TLV-encoded options. Figure 2 gives an example of an ACH TLV encapsulated in IPv6 Destination Options Header.

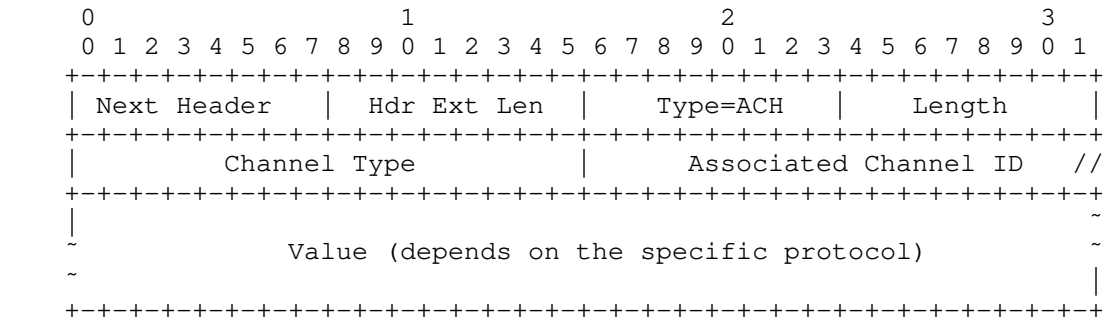


Figure 2 ACH TLV in IPv6 Destination Options Header

According to the note 1 and note 3 described in section 4.1 of[RFC8200], ACH TLV encapsulated in IPv6 Destination Options Header can provide two semantics of associated channel. When only IPv6 Destination Options Header exists or IPv6 Destination Options Header exists after the Routing Header, an end to end associated channel is provided to transmit the messages between two endpoints. When both IPv6 Destination Options Header and Routing Header exist, and IPv6 Destination Options Header exists before the Routing Header, an associated channel is provided at network nodes of the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header.

3.3.2. Encapsulated in IPv6 Hop-by-Hop Options Header

ACH TLV can be encapsulated in IPv6 Hop-by-Hop Options Header as the TLV-encoded options. Same option type numbering space is used for both Hop-by-Hop Options header and Destination Options header. Similarly, the ACH TLV in IPv6 Hop-by-Hop Options Header shares the same encapsulation shown in Figure 2.

When it is encapsulated in IPv6 Hop-by-Hop Options Header, it provides an associated channel at every node along the forwarding path.

3.3.3. Encapsulated in IPv6 Segment Routing Header

ACH TLV can be encapsulated in IPv6 Segment Routing Header, as SRH optional TLV. Figure 3 gives an example of an ACH TLV encapsulated in IPv6 Segment Routing Header.

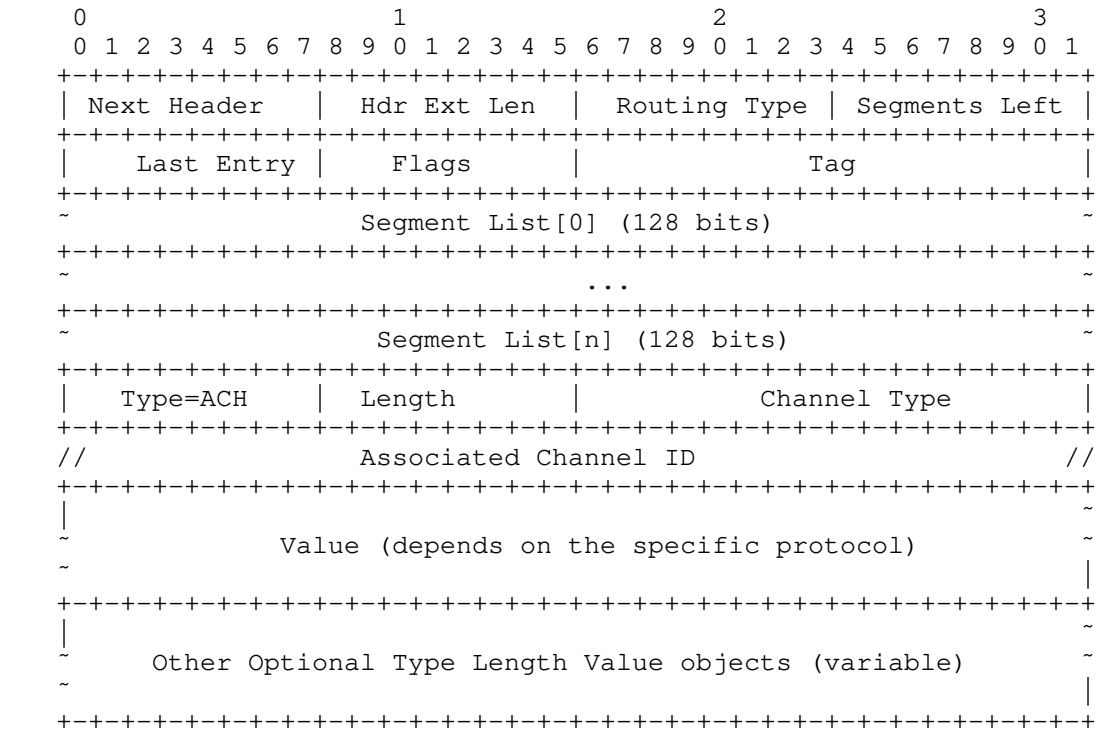


Figure 3 ACH TLV in IPv6 Segment Routing Header

When ACH TLV is encapsulated in IPv6 Segment Routing Header, it provides an associated channel at every SRv6 endpoints along the path.

3.3.4. Encapsulated in Payload

ACH TLV can also be encapsulated in the payload of an IPv6 packet. The term of payload here means the octets after the IPv6 header and extension headers. A synthetic packet is created with the payload of messages. and transmitted in an associated channel. The synthetic packet can use the same routing information with service data whose associated channel is associated to. For example, synthetic packet can encapsulate the same segment list as the one used in IPv6 SRH of service data.

3.4. Processing of ACH TLV

Take the ACH TLV encapsulated in Segment Routing Header as an example. At headend, ACH TLV is encapsulated with control and management messages in Segment Routing Header. When midpoint or tail-end receives an SRv6 packet with ACH TLV, it recognizes the ACH TLV, check the Channel Type field to interpret the protocol, and continue with processing of messages. The processing of message is not limited, for example READ or/and WRITE. It should depend on the specification of protocols used in the associated channel.

4. Applicability

4.1. Path Identification

In a native IPv6 network, packets is transmitted hop by hop, there is no way to identify an IPv6 forwarding path. The path needs to be identified when OAM or protection switchover is applied to the path.

4.2. OAM

OAM includes the a group of functions such as connectivity verification, fault indication and detection, and performance measurement of loss and delay etc. For example, BFD defines a generic control packet format that can be encapsulated in different data planes to provide low-overhead and short-duration failure detection function. The format can also be encapsulated in ACH TLV as the option TLV of Destination Options Header, to provide the same connectivity verification and fault detect functions without introducing upper layer protocols. Another example is to encapsulate PDU formats of Ethernet OAM [ITU-T G.8013] in Value field of ACH TLV to provide a set of OAM functions. By using ACH TLV to carry OAM messages in associated channel, different OAM functions can be easily integrated. The OAM functions can be performed in either end-to-end or hop-by-hop mode. For example, signal degrade happens on the intermediate node could be discovered and further indicated in associated channel to monitor the path status.

4.3. Assist to Protection Switchover

Linear protection [RFC6378] provides a very flexible protection mechanism in a mesh network because it can operate between any pair of endpoints. ACH TLV can be used to transmit the protection state control messages on an IPv6 forwarding path to provide the function of bidirectional protection switchover.

5. IANA Considerations

- This document requests IANA to assign a codepoint of Destination Options and Hop-by-Hop Options.
- This document requests IANA to assign a codepoint of Segment Routing Header TLVs to indicate ACH TLV.
- This document request IANA to create a new IANA-managed registry of ACH Channel Type to identify the usage of associated channel.

6. Security Considerations

TBD

7. Acknowledgements

TBD

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[I-D.ietf-spring-srv6-path-segment] Li, C., Cheng, W., Chen, M., Dhody, D., and R. Gandhi, "Path Segment for SRv6 (Segment Routing in IPv6)", draft-ietf-spring-srv6-path-segment-00 (work in progress), November 2020.

[RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <https://www.rfc-editor.org/info/rfc6378>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

Authors' Addresses

Fan Yang
Huawei Technologies
Beijing
China

Email: shirley.yangfan@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
Beijing
China

Email: mach.chen@huawei.com

Tianran Zhou
Huawei Technologies
Beijing
China

Email: zhoutianran@huawei.com