Abstract

This document proposes a method to support connection-oriented path in the SRv6 network. Two related SRv6 Functions need to be supported on each node along the connection-oriented 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/.

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 January 6, 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
SRv6 Network Programming concept is introduced in [RFC8986] and [I-D.filsfils-spring-srv6-net-pgm-illustration], which enables a data plane based network programming mechanism that goes beyond mere packet routing.

According to [RFC8986], an SRv6 SID is defined as the format of LOC:FUNCT:ARG, where the LOC stands for a locator, the FUNCT stands for a function, and the ARG is optional and stands for the arguments of the function. The locator is usually used to route the packet to the node who generates the SID. The basic functions of SRv6 are End (related to a node) and End.X (related to a link/adjacency), and many other functions are also defined, including some VPN related ones and some binding SIDs. In addition, it is said that even a local VM or container which can apply any complex processing on the packet can be defined as a function. The functions may or may not include arguments.

Based on SRv6, a node in the network can initiate a SID list <SID1, SID2, SID3> for a flow, so that a packet of the flow would be routed to the first node where the function1 related to SID1 would be implemented, then be routed to the second node where function2 related to SID2 would be implemented, and trigger similar operations according to SID3.

In fact, both MPLS and SRv6 are some kind of languages that support network programming. By using a label to represent a VPN instance, MPLS provides a good support to the VPN services in the network. SRv6 now shows a more powerful capability in network programming.
Perhaps in future, a lot of new network characteristics would be
developed based on SRv6; meanwhile, some old network characteristics
may also be realized by using SRv6 in order to integrate network
protocols, and simplify the network. This document gives an example
of the later.

Traditional MPLS transport is not source routing based, but is label
switching based. In MPLS networks, we can establish a label
switching path for a specific flow. It looks like a connection-
oriented path. If using the current SRv6 mechanism, we need to
initiate a SID list <SID1, SID2, SID3, ...> that includes every node
along the path, which is inconvenience. This document proposes a new
SRv6 mechanism to support the connection-oriented path by defining
two new functions on the node.

The motivation to support the connection-oriented path in SRv6 is
that sometimes a strict hop-by-hop TE path is needed in the network,
such as a DetNet path defined in RFC 8655 [RFC8655]. In one
realization of DetNet, each node along the path need allocate
specific resources to the critical traffic, and a fixed path must be
used. In future, the network may evolve to a pure SRv6 network
without MPLS. In this situation, SRv6 should support some old
network characteristics, such as the connection-oriented
characteristic mentioned in this document.

2. Data Plane for Connection-oriented Path

Data plane for a connection-oriented path in SRv6 is easy to design.
We just need to define a new End.XCopd function, which is similar to
END.X (binding to a cross-connected adjacency in Layer-3), but
includes a label argument. The function needs to be supported on
each node along the path, and it is used to support the connection-
oriented path on the data plane.

When receiving a packet with an End.XCopd SID S as the Destination
Address (DA), the node will match the SID in "My SID Table" to ensure
that S is generated by itself, and also check whether the label is
valid. If all checks are ok, the node should be able to obtain the
outgoing SID S2 in the "My SID Table". The node should replace the
DA with the outgoing SID S2, and forward the packet to the layer-3
adjacency bound to the SID S.

The penultimate node along the path will find that the connection-
oriented path is about to terminate, so that it will do normal End.X
operations, i.e., decrement SL, update the IPv6 DA with SRH[SL], and
forward the packet to the layer-3 adjacency bound to the SID.
Figure 1 shows an example of label switching in SRv6. It is assumed that each NodeX has a locator as AX. Node 1 generates a packet to Node 5 with an SRH header: <A1::End.XCopd:ARG1, A5::End.DT4> and an <SA, DA> pair: <A1::, A1::End.XCopd:ARG1>. And it is assumed that A1::End.XCopd:ARG1 can match a "switching table" entry: incoming SID A1::End.XCopd:ARG1, outgoing SID A2::End.XCopd:ARG2, and an interface binding to this End.XCopd:ARG1 function. Hence, after the process of "label switching", the Node 1 sends out a packet with an SRH header: <A1::End.XCopd:ARG1, A5::End.DT4> and an <SA, DA> pair: <A1::, A2::End.XCopd:ARG2>.

We assume that the Node 2 has a switching table entry: incoming SID A2::End.XCopd:ARG2, outgoing SID A3::End.XCopd:ARG3, and an interface binding to that End.XCopd:ARG2 function, so that the packet will be sent to Node 3, and then Node 4.

We also assume that the Node 4 has a switching table entry: incoming SID A4::End.XCopd:ARG4, outgoing SID A5::End.XCopd:0003, and an interface binding to that End.XCopd function. When the label "0003" appears, it means the node is the penultimate node. The Node 4 will do normal End.X operations, and sends out a packet without an SRH header, but with an <SA, DA> pair: <A1::, A5::End.DT4>.

The switching table should be established in advance on each node along the path, and it demonstrates the mapping relationship of the incoming SID and the outgoing SID. In the first hop of the path, it also demonstrates the mapping relationship of the first SID and the connection-oriented path. The way by which the label switching table is established on each node is described in the following section.

3. Control Plane for Connection-oriented Path

A PCE-based/controller-based method can surely configure each node along the path with a proper label switching table. However, this document also provides another optional mechanism for the distributed control plane. In fact, this method looks like what the RSVP-TE does.
in the MPLS network defined in RFC 3209 [RFC3209]. In other words, we can simulate some basic functions of RSVP-TE by using SRv6 network programming.

We need to define a new End.Copc function, which is used to establish and maintain the connection-oriented path in the SRv6 network. The function needs to be supported on each node along the path.

The End.Copc function also includes a label argument. Some of the label space should be reserved. In this document, we suppose that the label "0000" stands for the path establishment procedure. If a node receives a packet with an End.Copc function as the DA with a label value "0000", the node will trigger the path establishment procedure just as what the PATH message does in RSVP-TE. If a node receives a packet with an End.Copc function as the DA with a normal label value, the node will use the downstream label to establish a label switching table entry just as what the RESV message does in RSVP-TE.

However, in this way, the Head-End needs to notify each node along the path by some means, and we do not have a notification mechanism between different nodes in the data-plane network programming now. This document suggests to enable a simple notification method for the data-plane network programming if the information is not that complicated. For example, we can send a "ping" message with a specific TLV containing the necessary information. The advantage is easy to inter operate.

```
+-------------+-------------+-------------+-------------+-------------+
|   Node1     |   Node2     |   Node3     |   Node4     |   Node5     |
+-------------+-------------+-------------+-------------+-------------+
|   A1::, A2::End.Copc:0000> -->| A1::, A3::End.Copc:0000> -->|
|   A1::, A3::End.Copc:0000> -->| A1::, A4::End.Copc:0000> -->|
|   A1::, A4::End.Copc:0000> -->| A1::, A5::Copc:0000>    -->|
|   A1::, A5::Copc:0003>    -->| A1::, A3::End.Copc:0117> <---|
|   A1::, A2::End.Copc:0445> <---|
|   A1::, A1::End.Copc:0998> <<<|
```

Figure 2: <SA, DA> changes along the Connection-oriented Path on the control plane

Figure 2 shows an example of label switching path establishment in SRv6. Node 1 generates a "ping" packet with an <SA, DA> pair: <A1::, A1::End.Copc:0000>. A new TLV defined for "ping" would include each End.Copc functions along the path, and the TLV is supposed to be in
the payload of the "ping" message. And it is assumed that A1::End.Copc:0000 can match the "My SID Table", and the DA is replaced by A2::End.Copc:0000 after the Node 1 has read the specific TLV in the payload. Then, Node 1 sends the packet to the Node2 according to the new DA. Similar operation takes place in Node2-4.

Node 5 will find it is the last SID after reading the specific TLV in the payload. It generates a label "0003", and sends back the packet, as a response packet. In this time, the "ping" packet has an <SA, DA> pair: <A1::, A4::Copc:0003>. Based on the information in the response packet, Node 4 can generate a label "0117", and establish a swapping table entry: incoming SID A4::End.XCopc:0117, outgoing SID A5::End.XCopc:0003, and an interface binding to the A4’s End.XCopc function.

Similarly, the Node 3 can generate a label "0445", and establish a swapping table entry: incoming SID A3::End.XCopc:0445, outgoing SID A4::End.XCopc:0117, and an interface binding to the A3’s End.XCopc function.

The Node 1 will find it is the first SID after reading the specific TLV in the payload of the "ping" message, and optionally, it can also generate a label "1111", and establish a swapping table entry: incoming SID A1::End.XCopc:1111, outgoing SID A2::End.XCopc:0998, and an interface binding to the A1’s End.XCopc function.

The swapping table is used in this document for the convenience of description. In fact, it should be several entries in the "My SID Table". We can also define some label for the procedure of releasing the path in future version of the draft.

4. IANA Considerations

TBD.

5. Security Considerations

TBD.

6. Acknowledgements

TBD.

7. References
7.1. Normative References


7.2. Informative References


Authors’ Addresses

Zongpeng Du
China Mobile
No.32 XuanWuMen West Street
Beijing  100053
China

Email: duzongpeng@foxmail.com
Peng Liu  
China Mobile  
No.32 XuanWuMen West Street  
Beijing  100053  
China  

Email: liupengyjy@chinamobile.com
Redundancy Policy for Redundancy Protection
draft-geng-spring-redundancy-policy-04

Abstract

This document introduces a variant of SR Policy called Redundancy Policy, in order to instruct the replication of service packets and assign more than one redundancy forwarding paths used for redundancy protection.

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 26 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Redundancy protection [I-D.ietf-spring-sr-redundancy-protection] is a generalized protection mechanism by replicating and transmitting copies of flow packets on redundancy node over multiple different and disjoint paths, and further eliminating the redundant packets at merging node. This document introduces Redundancy Policy to support redundancy protection, which is a variant of SR Policy [I-D.ietf-spring-segment-routing-policy]. Redundancy Policy instructs the replication of service packets and assigns more than one equivalent forwarding paths used for redundancy protection. Redundancy Policy applies equally to both MPLS data plane (SR-MPLS) [RFC8660] and Segment Routing with IPv6 data plane (SRv6) [RFC8986].
2. Terminology and Conventions

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.

The other terminologies used in this document are:

- Redundancy Node: the start point of redundancy protection, where the network node replicates the flow packets.
- Merging Node: the end point of redundancy protection, where the network node eliminates and ordering (optionally) the flow packets.
- Redundancy Policy: an extended SR Policy which instructs more than one redundancy forwarding paths to support packet redundant transmission.

3. Redundancy Policy

Redundancy Policy is used to enable packet replication and instantiation more than one active ordered lists of segments between redundancy node and merging node to steer the same flow through different paths in an SR domain.

3.1. Identification of Redundancy Policy

Redundancy Policy is a variant of SR Policy and also identified through the tuple <headend, color, endpoint>. Specifically, a redundancy policy is identified by <redundancy node, color, merging node>. Redundancy node is specified as IPv4/IPv6 address of headend of Redundancy Policy, which is the node to perform packet replication. Merging node is specified as IPv4/IPv6 address of endpoint of Redundancy Policy, which is the node to perform packet elimination. The value of color specifies the intent of the redundancy policy is "redundancy protection for high reliability", which indicates service packets are replicated into multiple copies and carried on different forwarding paths.

3.2. Structure of Redundancy Policy

Redundancy policy shares the basic structure and elements with SR Policy and its information model is shown in the following:
Redundancy policy POL1 <R Node= R1, Color = 1, M Node = M1>
Candidate-path CP1 <protocol-origin = 20, originator = 100:1.1.1.1, discriminator = 1>
   Flag Redundancy
   Preference 200
   SID-List1 <SID11...SID1i>
   SID-List2 <SID21...SID2j>
Candidate-path CP2 <protocol-origin = 20, originator = 100:2.2.2.2, discriminator = 2>
   Preference 100
   Weight W3, SID-List3 <SID31...SID3i>

The Redundancy Policy POL1 is identified by the tuple <redundancy node, color, merging node>, in which R1 is the redundancy node, M1 is the merging node, and Color 1 represents the intent of redundancy protection. Two candidate-paths CP1 and CP2 instruct the ordered segment lists from redundancy node to merging node. In candidate path CP1, a new attribute Flag is added to indicate the type of candidate path. When the candidate path is indicated with the flag of redundancy, the attribute Weight is not applicable to the SID-Lists and all SID Lists of the candidate path are used for redundancy forwarding. Regarding the other attributes of candidate path such as originator, preference, priority, segment-list etc, the definitions apply the same as [I-D.ietf-spring-segment-routing-policy].

3.3. Flag of a Candidate Path

Flag is an optional attribute of a candidate path, which is used to indicate the type of a candidate path is for redundancy forwarding. When the candidate path with flag of redundancy is selected as the active candidate path, this SR Policy is identified as the Redundancy Policy. Flag of a candidate path is an 8-bit bitmap. The table below specifies the current definition of Flag:

<table>
<thead>
<tr>
<th>Bitmap</th>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>R</td>
<td>Redundancy paths</td>
</tr>
<tr>
<td>1-7</td>
<td>U</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Figure 2: Flag
3.4. Behavior of Redundancy Policy

When the SR policy is identified as a redundancy policy, network node uses rules to compute and select the valid active ordered segment-lists for redundancy forwarding. The specific rules are:

* The candidate paths are selected to determine the best path of an SR policy. Preference, Protocol-Origin, and other tie-breaking rules defined in section 2.9 of [I-D.ietf-spring-segment-routing-policy] are evaluated until only one valid best path is selected.

* In a redundancy policy, the candidate path with a flag of redundancy is always selected as the best path in the first place.

* When the selected active candidate path is with a flag of redundancy, all the segment-lists of the candidate path are used as the active segment-lists for redundancy forwarding, where each active segment-list carries an entire copy of service packets.

* Weight is not applicable for the segment-lists in a candidate path with a flag of redundancy. Redundancy policy has no purpose of weighted load-balancing.

* The candidate path without a flag of redundancy in the same SR policy with the candidate paths with a flag, is considered as the backup path, which allowing provisioning of multiple path options.

Take the information model in section 3.2 as an example, preference value 200 of CP1 is higher than preference value 100 of CP2, thus CP1 is selected as the active candidate path. Because CP1 is with the flag of redundancy, both Segment-List1 and Segment-List2 are selected as the active Segment-Lists for redundancy forwarding. After service packets are replicated, each segment-list forwards each replicas of service packets. When CP1 becomes invalid and fallbacks to CP2, CP2 provides the backup path to the redundancy forwarding.

3.5. BSID and Redundancy Policy

Redundancy policy can be optionally associated with a Binding Segment. Redundancy SID defined in [I-D.ietf-spring-sr-redundancy-protection] can be the Binding SID of redundancy policy. In other words, Redundancy SID triggers the instantiation of redundancy policy in the forwarding plane on redundancy node.
3.6. Steering into a Redundancy Policy

A packet is steered into a Redundancy Policy at a redundancy node in following ways:

* Incoming packets have an active SID matching the Redundancy SID at the redundancy node.

* Per-destination Steering: incoming packets match a BGP/Service route which recurses on a Redundancy Policy.

* Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are Redundancy Policy.

* Policy-based Steering: incoming packets match a routing policy which redirects them on a Redundancy Policy.

3.7. Protocol Extensions

Similar to SR Policy, Redundancy Policy requires the control plane protocol extensions to distribute candidate paths and other information. New sub-TLVs are expected to be defined to encode new information of Redundancy Policy Candidate Paths in BGP [I-D.ietf-idr-segment-routing-te-policy] and PCEP [I-D.ietf-pce-segment-routing-policy-cp].

4. IANA Considerations

TBD

5. Security Considerations

TBD

6. References

6.1. Normative References

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

Yang, et al. Expires 26 January 2023


6.2. Informative References


Authors’ Addresses

Fan Yang
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: shirley.yangfan@huawei.com

Xuesong Geng
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: gengxuesong@huawei.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: zhoutianran@huawei.com

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

One of the goals of DetNet is to provide bounded end-to-end latency for critical flows. This document defines how to leverage Segment Routing (SR) and Segment Routing over IPv6 (SRv6) to implement bounded latency. Specifically, new SRv6 SID function is used to specify bounded latency information for a packet. When forwarding devices along the path follow the instructions carried in the packet, the bounded latency is achieved by different implementations based on bounded latency information.

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 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Deterministic Networking (DetNet) provides a capability to carry specified data flows with extremely low data loss rates and bounded latency within a network domain. DetNet is enabled by a group of technologies, such as resource allocation, service protection and explicit routes ([RFC8655]).

Segment Routing (SR) leverages the source routing paradigm. A ingress node steers a packet through an ordered list of instructions, called "segments". When SR is used over the MPLS data plane, SIDs are an MPLS label or an index into an MPLS label space (either SRGB or SRLB).

SR can also be applied over IPv6 data plane using Routing Extension Header (SRH). Besides routing, the segment of SRv6 can indicate functions which are executed locally in the node where they are defined. SRv6 network programming makes it convenient to add sophisticated operations in the network. ([RFC8402])
DetNet data plane is enhanced to facilitate DetNet transit nodes to support end-to-end bounded latency transmission. [I-D.yzz-detnet-enhanced-data-plane] introduces an unified data plane field for bounded latency, which is called Bounded Latency Information (BLI). BLI is designed to cope with a variety of queuing/scheduling/shaping mechanisms in a uniform format in the data plane.

This document describes how to implement DetNet with SR or SRv6. It can provide: 1. Source routing, which can steer the DetNet flows through the network according to an explicit route with allocated resource by segment list in SRH; 2. Network programming, which can give packet instructions in every node along the path to guarantee bounded latency. DetNet SR MPLS/SRv6 data plane extensions for enhanced DetNet are defined in this document.

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 [RFC2119].

2.1. Terminology

Terminologies for DetNet go along with the definition in [RFC8655]. Other terminologies are defined as follows:

* NH: The IPv6 next-header field.

* SID: A Segment Identifier which represents a specific segment in a segment routing domain ([RFC8402]).

* SRH: The Segment Routing Header ([RFC8754]).

2.2. Conventions

Conventions in the document are defined as follows:

* NH=SRH means that NH is 43 with routing type 4.

* 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.

* SRH[SL] represents the SID pointed by the SL field, that is the SLth SID in the Segment List.
* (SA, DA) (S3, S2, S1; SL) represents an IPv6 packet with:

IPv6 header with source and destination addresses SA and DA respectively, and next-header SRH, with SID list <S1, S2, S3> with SegmentsLeft = SL

The payload of the packet is not represented

(S3, S2, S1; SL) represents the same SID list as <S1, S2, S3>, 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

3. SRv6 for Enhanced DetNet

To guarantee the end-to-end bounded latency transmission in DetNet network, bounded latency information is required to be conveyed inband with the service data to facilitate the queuing algorithm performed on the DetNet transit nodes. When the bounded latency information is used in DetNet IP data plane or DetNet MPLS data plane, it is carried in IP/UDP or MPLS encapsulations. When the bounded latency information is used in TSN over IP/MPLS data plane, the information used in TSN networks is transparently transmitted IP/UDP or MPLS encapsulations. Note that, which queuing mechanism is used is a local choice determined by DetNet transit nodes. It is not necessary to be explicitly indicated in packets.

When an SRv6 SID is in the Destination Address field of an IPv6 header of a packet, it is routed through transit nodes in an IPv6 network as an IPv6 address. SRv6 SID consists of LOC:FUNCT:ARG, where a locator (LOC) is encoded in the L most significant bits of the SID, followed by F bits of function (FUNCT) and A bits of arguments (ARG), which is defined in ([RFC8986]).

Bounded Latency Information (BLI) is defined in [I-D.yzz-detnet-enhanced-data-plane] to guide forwarding in network device, which could be initiated in SRv6 data plane. With the characteristics of Segment Routing, the bounded latency information could be coupled with explicit path to provide latency guarantee in each node/adjacency indicated by the segment list.

Bounded Latency Information is indicated by the allocated SID at each node along the path without maintaining per-flow states at the intermediate and egress nodes. Hence, it naturally supports flow aggregation, and that allows DetNet to support large number of DetNet flows and scale to large networks.
As defined in [I-D.yzz-detnet-enhanced-data-plane], 8 or more Bounded Latency Information Types (BLI Type) are introduced to differentiate the types of BLIs, based on the required information of queuing/scheduling/shaping mechanisms to guarantee bounded latency. Bounded Latency Information Value (BLI Value) is a specified value of a specific type of BLI to provide guidance for packet processing with the meaning of a particular BLI type. The pair <BLI Type, BLI Value> information should be indicated by SRv6 data plane.

The "Endpoint with L3 cross-connect" behavior ("End.X" for short) is a variant of the End behavior. It is the SRv6 instantiation of an Adj-SID ([RFC8402]), and its main use is for traffic-engineering policies.

Two new variations of End.X SID are defined for DetNet bounded latency, which are called End.X.BL and End.X.BLI respectively, and bounded latency information can be defined as functions or arguments in the new types of SID.

Editors Notes: Another option to implement this is to define new flavors. This method will be considered when not only End.X could be combined with BLI.

3.1. End.X.BL: Forwarding the packet with bounded latency guarantee

This document defines End.X.BL, which is used to identify Bounded Latency Information for Enhanced DetNet. End.X.BL a variation of End.X.

End.X.BL SID has two meanings: 1) to identify an interface/link, just like the adjacency SID; 2) to identify the pair <BLI Type and BLI Value> information on the interface/link to guarantee bounded latency. So different End.X.BL SIDs could be allocated to the same interface/link in order to indicated different pairs <BLI Type, BLI Value>.

The SRv6 encapsulation with End.X.BL SIDs is shown as follows:
When N receives a packet destined to S and S is a local End.X.BL SID, N does the following:
When an SRH is processed {
    If (Segments Left == 0) {
        Stop processing the SRH, and proceed to process the next header in the packet, whose type is identified by the Next Header field in the routing header.
    }
    If (IPv6 Hop Limit <= 1) {
        Send an ICMP Time Exceeded message to the Source Address with Code 0 (Hop limit exceeded in transit), interrupt packet processing, and discard the packet.
    }
    max_LE = (Hdr Ext Len / 2) - 1
    If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
        Send an ICMP Parameter Problem to the Source Address with Code 0 (Erroneous header field encountered) and Pointer set to the Segments Left field, interrupt packet processing, and discard the packet.
    }
    Decrement IPv6 Hop Limit by 1
    Decrement Segments Left by 1
    Update IPv6 DA with Segment List[Segments Left]
    Submit the packet to the IPv6 module for transmission to the new destination via a L3 adjacency indicated by the End.X.BL SID
    Send the packet out using <BLI Type, BLI Value> indicated by the End.X.BL SID with the corresponding bounded latency guarantee mechanism
}

### 3.2. End.X.BLI: Forward the packet with bounded latency guarantee through BLI

The "Endpoint with forwarding the packet with bounded latency guarantee by BLI" behavior ("End.X.BLI" for short) is a variant of the End behavior.

End.X.BLI SID has two meanings: 1) to identify an interface/link, just like the adjacency SID; 2) to identify the BLI Type to guarantee bounded latency. So different End.X.BLI could be allocated to the same interface/link in order to indicated different types of BLIs. The BLI Value corresponding to the End.X.BLI SID is carried explicitly in the SRv6 packet header.

There are 3 possible options for carrying variable BLI Value associated with the End.X.BLI SID, including:
* Option1: Arguments in End.X.BLI SID
* Option2: SRH TLV for BLI used together with End.X.BLI SID
* Option3: New options in DoH before SRH together with End.X.BLI SID

3.2.1. BLI in Arguments of End.X.BLI SID

The behavior also takes an argument: "Arg.BLI". This argument provides a local BLI Value information for bounded latency guarantee. The SRH with End.X.BLI SIDs is showed as follows:

```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |   Hdr Ext Len |  Routing Type |  Segment Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Last Entry  |     Flags     |              Tag              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Location & Function                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Segment List[n]                          |
|                      which is End.X.BLI SID                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Optional TLVS                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where

* Location&Function: the most significant bits that are used for routing and function indication;
* Bounded Latency Information: the least significant bits, which is defined [I-D.yzz-detnet-enhanced-data-plane].

When N receives a packet destined to S and S is a local End.X.BLI SID, N does the following:
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
         header in the packet, whose type is identified by
         the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address
         with Code 0 (Hop limit exceeded in transit),
         interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If (((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address
         with Code 0 (Erroneous header field encountered)
         and Pointer set to the Segments Left field,
         interrupt packet processing, and discard the packet.
S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Submit the packet to the IPv6 module for transmission
         to the new destination via a L3 adjacency indicated by the
         End.X.BLI SID
S16.   Send the packet out using BLI Type indicated by the
         End.X.BLI SID and BLI Value carried in the argument
         with the corresponding bounded latency guarantee mechanism
S17.   }

3.2.2. BLI in TLV of SRH

Optional TLV defined in SRH could also be extended for BLI, which is
used together with End.X.BLI.

3.2.2.1. BLI List TLV

When all or part of the nodes/adjacencies in the explicit path
indicated by the segment list request different BLI values
corresponding to the End.X.BLI SID to guarantee bounded latency, a
BLI List TLV is defined. The SRH with End.X.BLI SIDs is showed as
follows:
The Type field is 8 bits in length, and the value is TBD1.

The Length field is 8 bits in length and its value is variable, which depends on the length of BLI list.

BLI Left: 8-bit unsigned integer. Number of BLI remaining, i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination.

BLI List[0..m]: 32-bit unsigned integer, representing the nth BLI in the BLI list.

The BLI in the BLI list corresponds to the Segment in the Segment List one by one. The length of BLI List depends on the number of End.X.BLI in the segment list.

When N receives a packet destined to S and S is a local End.X.BLI SID, N does the following:
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
         header in the packet, whose type is identified by
         the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address
         with Code 0 (Hop limit exceeded in transit),
         interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If (((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address
         with Code 0 (Erroneous header field encountered)
         and Pointer set to the Segments Left field,
         interrupt packet processing, and discard the packet.
S11. }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Submit the packet to the IPv6 module for transmission
         to the new destination via a L3 adjacency
S16.   Send the packet out using BLI Type indicated by the
         End.X.BLI SID and BLI Value carried by BLI List[BLI Left]
         in SRH TLV and BLI Left--
         with the corresponding bounded latency guarantee mechanism
S17. }

3.2.2.2. Shared BLI TLV

When all the nodes/adjacencies in the explicit path indicated by the
segment list request the same BLI value to guarantee bounded latency,
the Shared BLI TLV is defined. The SRH with End.X.BLI SIDs is showed
as follows:
The Type field is 8 bits in length, and the value is TBD2.

The Length field is 8 bits in length and its value is variable, which depends on the length of BLI list.

The Shared BLI field is 32 bits in length and corresponds to definition of BLI in [I-D.yzz-detnet-enhanced-data-plane].

When N receives a packet destined to S and S is a local End.X.BLI SID, N does the following:
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.      Stop processing the SRH, and proceed to process the next
       header in the packet, whose type is identified by
       the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.      Send an ICMP Time Exceeded message to the Source Address
       with Code 0 (Hop limit exceeded in transit),
       interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If (((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.      Send an ICMP Parameter Problem to the Source Address
       with Code 0 (Erroneous header field encountered)
       and Pointer set to the Segments Left field,
       interrupt packet processing, and discard the packet.
S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List[Segments Left]
S15.   Submit the packet to the IPv6 module for transmission
       to the new destination via a L3 adjacency
S16.   Send the packet out using BLI Type indicated by the
       End.X.BLI SID and BLI Value indicated by Shared BLI TLV
       with the corresponding bounded latency guarantee mechanism
S17. }

3.2.2.3. BLI Options in DOH before SRH

According to [RFC8200], BLI could also be defined through DOH before
SRH for the specified segment. For the case of BLI List, considering
that the location of DOH is before SRH, it is not recommended to be
defined in DoH, because it will affect the processing efficiency of
Segment in SRH. For Shared BLI TLV, it can be carried by the DOH
Option. In order for the consistency, this document recommends to
use the SRH TLV to carry both information.

4. SR MPLS for Enhanced DetNet

For SR MPLS data plane, this document defines a new segment that is
called a BLI Segment, which is used to identify Bounded Latency
Information for Enhanced DetNet just like End.BL SID. A BLI Segment
is an adjacency segment and allocated from the Segment Routing Local
Block (SRLB)[RFC8402]. BLI Segment indicateds <BLI Type, BLI Value>
of an interface/link. So different BLI segments could be allocated
to the same interface/link in order to indicated different pairs <BLI Type, BLI Value>.

Editors Notes: SR MPLS extension with meta data which is still under discussion will be defined based on the progress of MPLS DT. The possible definition of MPLS segment associated with the variable BLI values like the SRv6 End.X.BLI will be defined in the future version.

5. IANA Considerations

The following codepoints are defined in this document in Segment Routing Header TLVs registry:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD1</td>
<td>BLI List TLV</td>
<td>This document</td>
</tr>
<tr>
<td>TBD2</td>
<td>Shared BLI TLV</td>
<td>This document</td>
</tr>
</tbody>
</table>

6. Security Considerations

TBD

7. Normative References

[I-D.ietf-detnet-bounded-latency]

[I-D.yzz-detnet-enhanced-data-plane]


Authors’ Addresses

Xuesong Geng
Huawei
Email: gengxuesong@huawei.com

Zhenbin Li
Huawei
Email: lizhenbin@huawei.com

Tianran Zhou
Huawei
Email: zhoutianran@huawei.com
Problem statement for Inter-domain Intent-aware Routing using Color
draft-hr-spring-intentaware-routing-using-color-00

Abstract

This draft describes the scope, set of use-cases and requirements for a distributed routing based solution to establish end-to-end intent-aware paths spanning multi-domain packet networks. The document
focuses on BGP given its predominant use in inter-domain routing deployments, however the requirements may also apply to other solutions.

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 January 15, 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction ........................................... 4
  1.1. Objectives ........................................ 4
2. Typical large scale network deployment scenarios .......... 5
  2.1. 5G access networks .................................. 5
  2.2. WAN networks for Content distribution ................. 6
  2.3. Data Center Inter-connect Networks .................. 7
3. Use Cases for Inter-domain Intent-based Transport ........ 7
  3.1. Inter-domain Data Sovereignty ........................ 7
  3.2. Inter-domain Low-Latency Services ................... 8
  3.3. Inter-domain Service Function Chaining ................ 8
  3.4. Inter-domain Multicast Use cases ................... 9
4. Deployment use cases ..................................... 9
4.1. Network Domains under different administration .......... 9
5. Intent-Aware Routing Framework .......................... 10
  5.1. Intent .............................................. 10
  5.2. Color .............................................. 11
  5.3. Colored Service Route ................................ 11
  5.4. Intent-Aware Route using Color ....................... 11
  5.5. Service Route Automated Steering on intent-aware route using color ................................ 11
  5.6. Inter-Domain intent-aware routing using colors with SR Policy .................................. 11
  5.7. Motivation for a BGP-based intent-aware routing solution using colors ............................ 12
  5.8. BGP Intent-Aware Routing using Color ................ 12
  5.9. Architectural consistency among intent-aware routing solutions using colors ..................... 12
6. Technical Requirements .................................. 14
  6.1. Intent Requirements .................................. 14
    6.1.1. Transport Network Intent Requirements ............ 15
    6.1.2. VPN (Service Layer) Network Intent Requirements ... 23
  6.2. Traffic Steering Requirements ........................ 23
  6.3. Deployment Requirements .............................. 24
    6.3.1. Multi-domain deployment designs .................. 24
    6.3.2. Scalability Requirements ........................ 30
    6.3.3. Network Availability Requirements ............... 32
    6.3.4. BGP Protocol Requirements ........................ 33
    6.3.5. Multicast Intent Requirements ................... 35
    6.3.6. OAM Requirements ................................ 35
7. Backward Compatibility .................................. 35
8. Security Considerations .................................. 35
9. IANA Considerations ...................................... 35
10. Acknowledgements ....................................... 35
11. Contributors .......................................... 35
12. References ............................................. 36
  12.1. Normative References ................................ 36
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 extend vertically in terms of the different services and variety of intent 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. In MPLS based networks BGP-LU (RFC8277) has been widely deployed for providing reachability across multiple domains.

The evolving network requirements (e.g. 5G, native cloud) in such a multi-domain network requires the establishment of paths that span multiple domains or AS’s while maintaining specific transport characteristics or intent (e.g. bandwidth, latency). There is also a need to provide flexible, scalable, and reliable end-to-end connectivity for multiple services across the network domains.

1.1. Objectives

This document describes requirements for scalable, intent-aware reachability across multiple domains.

The base problem that it focuses on is the BGP-based delivery of an intent across several transport domains, however the requirements may also apply to other distributed solutions.

The problem space is then widened to include any intent (including Network Function Virtualization (NFV) chains and their location), any data plane and the application of intent-based routing to the Service/VPN routes.

It is intended that the requirements enable the design of technology and protocol extensions that address the widest application, while ensuring consistency and compatibility with existing deployed solutions.
2. Typical large scale network deployment scenarios

This section describes a few typical deployment scenarios that involve large-scale multi-domain network designs and use of various topology, IGP and BGP routing models. While the examples use specific types of deployments for illustration, neither the use-cases nor the network designs are limited to any particular provider deployment.

2.1. 5G access 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.

Such transport networks use multiple domains to support scalability. For this analysis, we consider a representative network design with four levels 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.

```
+-------+   +-------+   +------+   +------+
|       |   |       |   |      |   |      |
+--+ P-AGG1+---+ AGG1 +---+ ABR1 +---+ LSR1 +--> to ABR
|       |   |       |   |      |   |      |
+-------+   +-------+   +------+   +------+
ISIS L1    ISIS L2                    ISIS L2

Figure 1: 5G network
```

5G networks support a variety of service use cases that may require end-to-end network slicing. In certain cases, the end-to-end connectivity requires the ability to forward over intent-aware paths, such as paths delivering low-delay. The inter-domain routing
solution should support the establishment of end to end paths that
address specific intent requirements, as well as support multiple
such paths to address slicing requirements.

2.2. WAN networks for Content distribution

Networks built for providing delivery of content are geographically
distributed by design to provide connectivity in multiple regions and
sharing of data across regions.

As these WAN networks grow beyond several thousand nodes, they are
divided into multiple IGP domains for scale and reliability. An
illustration is provided in in Figure 2.

```
+-------+     +-------+     +-------+
|       |     |       |     |       |
|     ABR1  ABR2    ABR3   ABR4     |
|       |     |       |     |       |
|       |     |       |     |       |
PE1+   D1 +-----+  D2   +-----+   D3  +PE2
|       |     |       |     |       |
|     ABR11  ABR22  ABR33  ABR44    |
|       |     |       |     |       |
+-------+     +-------+     +-------+

|ISIS1-|      |-ISIS2-|     |-ISIS3-|
```

Figure 2: Content distribution WAN Example

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
distributed inter-domain solution should be able to create highly
constrained inter domain TE paths in a scalable manner.

Some deployments may use a controller to acquire the topologies of
multiple domains and build end-to-end constrained paths. This
approach can be scaled with hierarchical controllers. However, there
is still a risk of a loss of network connectivity to one or more
controllers, which could lead to a failure to satisfy the strict
requirements of data sovereignty. The network should be able to have
pre-established TE paths end-to-end that don’t rely on controllers,
to address these failure scenarios.
2.3. Data Center Inter-connect Networks

Distributed data centers are playing an increasingly important role in providing access to information and applications. Geographically diverse data centers are usually connected via a high speed, reliable and secure DC WAN core network.

One variation of a DCI topology is shown in Figure 3.

```
+-------+     +-------+     +-------+     +-------+
|       |     | DC WAN |     |       |
| ASBR1 ASBR2 ASBR3 ASBR4 |               |
|       |     |       |     |       |
| PE1+   DC1  +-----+  CORE +-----+  DC2  +PE2 |
| ASBR11 ASBR22 ASBR33 ASBR44 |               |
|       |     |       |     |       |
+-------+     +-------+     +-------+

|--ISIS1--|      |--ISIS2--|    |--ISIS3--|
```

Figure 3: DCI Network

In many DC WAN deployments, applications require end-to-end path diversity and end-to-end low latency paths.

Another consideration in DC WAN deployments is the choice of encapsulation technologies. Some deployments use the same tunneling mechanism within the DC and DCI networks, while other deployments use different mechanisms in each. It is important for a solution to provide flexibility in 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. They apply to all the different deployment scenarios described above.

3.1. Inter-domain Data Sovereignty
Figure 4: Multi domain Network

Figure 4 depicts an example of a WAN with multiple ASes, where each AS serves a continent. 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.

In other networks, the domain to avoid may encompass an entire AS.

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. 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.
3.4. 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.

![Diagram of inter-domain multicast use cases]

Figure 5: Multicast use cases

Figure 5 shows a simplified multi-domain network supporting multicast. Multicast sources S1 and S2 are in a different domain from the receivers R1 and R2. The solution should support establishment of intent-aware multicast distribution trees (P tunnels) across the domains and steer customer multicast streams on it. It should maintain the scaling properties of a multi-domain architecture by avoiding leaking of RPF routing state into the IGP domains.

4. Deployment use cases

4.1. Network Domains under different administration

![Diagram of networks with inconsistent intent mappings]

Figure 6: Networks with inconsistent intent mappings

In diagram Figure 5 above, AS1 and AS2 may be operating as closely coordinated but independent administrative domains, and still require
end-to-end paths across the two ASes to deliver services. This scenario could be a result of a merger. It is possible that AS1 and AS2 may have assigned different values for the same intent.

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.

An inter-domain solution should provide effective mechanisms to translate intent across domains without requiring renumbering of the intent mapping.

5. Intent-Aware Routing Framework

This section describes the basic concepts, terminologies and architectural principles that define intent-aware routing and the protocols and technologies that currently support it. The goal of this section is to establish the requirement for consistency with existing deployed solutions and describe the framework for it.

The figure below is used as reference.

```
+-----------------------------------+
|----+                         +----|
| E1 |                         | E2 |- V/v with C
|----+                         +----|
+-----------------------------------+
```

Figure 7: Intent-aware routing using color reference topology

5.1. Intent

Intent in routing may be any combination of the following behaviors:

- Topology path selection (e.g. minimize metric, avoid resource)
- NFV service insertion (e.g. service chain steering)
- Per-hop behavior (e.g. QoS for 5G slice)

An intent-aware routed path may be within a single network domain or across multiple domains.
5.2. Color

Color is a 32-bit numerical value that is associated with an intent, as defined in [I-D.ietf-spring-segment-routing-policy]

5.3. Colored Service Route

An Egress PE E2 colors a BGP service (e.g. VPN) route V/v to indicate the particular intent that E2 requests for the traffic bound to V/v. The color (C) is encoded as a BGP Color Extended community [I-D.ietf-idr-tunnel-encaps].

5.4. Intent-Aware Route using Color

(C, E2) represents a intent-aware route to E2 which satisfies the intent associated with color C.

Multiple technologies already provide intent-aware paths in solutions that are widely deployed.

- SR Policy [I-D.ietf-spring-segment-routing-policy]
- IGP Flex-Algo [I-D.ietf-lsr-flex-algo]

In the context of large-scale SR-MPLS networks, SR Policy is applicable to both intra-domain and inter-domain deployments; whereas IGP Flex-Algo is better suited to intra-domain scenarios.

5.5. Service Route Automated Steering on intent-aware route using color

An ingress PE E1 automatically steers V-destined packets onto a intent-aware path bound to (C, E2). If several such paths exist, a preference scheme is used to select the best path: E.g. IGP Flex-Algo first, then SR Policy.

5.6. Inter-Domain intent-aware routing using colors with SR Policy

If E1 and E2 are in different domains, E1 may request an SR-PCE in its domain for a path to (C, E2). The SR-PCE (or a set of them) computes the end-to-end path and installs it at E1 as an SR Policy. The end-to-end intent-aware path may seamlessly cross multiple domains.
5.7. Motivation for a BGP-based intent-aware routing solution using colors

While the following requirements may be covered with an SR Policy solution, an operator may prefer a BGP-based solution due to:

- Operational familiarity and expectation of incremental evolution from an existing Seamless-MPLS/BGP-LU inter-domain deployment [I-D.ietf-mpls-seamless-mpls]
- Expectation of higher scale with BGP
- Expectation of a familiar operational trust model between BGP domains (peering policy)

5.8. BGP Intent-Aware Routing using Color

A BGP Intent-Aware Routing solution signals intent-aware routes to reach a given destination (e.g. E2). (C, E2) represents a BGP hop-by-hop distributed route that builds an inter-domain intent-aware path to E2 for color C.

5.9. Architectural consistency among intent-aware routing solutions using colors

As seen above, multiple technologies exist that provide intent aware routing in a network. A BGP based solution must be compliant with the existing principles that apply to them.

A deployment model that provides consistency is as follows:

- Service routes are colored using BGP Color Extended-Community to request intent [I.D.ietf-spring-segment-routing-policy]
  * V/v via E, colored with C
- Colored service routes are automatically steered on an appropriate intent-aware path using color
  * V/v via E with C is steered via (E, C)
  * (E, C) provided by any intent-aware technology or protocol
- Intent-aware routes may resolve recursively via other intent-aware routes
  * (E, C) via N recursively resolves via (N, C)
Here is a brief example that illustrates these principles.

Figure 8: Inter-domain intent-aware routing using color reference topology

In the figure above, all the nodes are part of an inter-domain network under a single authority and with a consistent color-to-intent mapping:

- Color C1 is mapped to "low delay"
  * Flex-Algo FA1 is mapped to "low delay" and hence to C1 in each domain

- Color C2 is mapped to "low delay and avoid resource R"
  * Flex-Algo FA2 is mapped to "low delay and avoid resource R" and hence to C2 in each domain

E1 receives two BGP colored service routes from E2:

- V/v with BGP Color Extended community C1
- W/w with BGP Color Extended community C2

E1 has the following inter-domain intent-aware paths using color:

- (E2, C1) provided by BGP which recursively resolves via intra-domain intent-aware paths:
  * (N1, C1) provided by IGP FA1 in Domain1
  * (N2, C1) provided by SR Policy bound to color C1 in Domain2
E1 automatically steers the received colored service routes as follows:

* V/v via (E2, C1) provided by BGP intent-aware route using color
* W/w via (E2, C2) provided by SR Policy

The example illustrates the benefits provided by leveraging the architectural principles:

* Seamless co-existence of multiple intent-aware technologies, e.g. BGP and SR Policy
  * V/v is steered on BGP intent-aware path
  * W/w is steered on SR Policy intent-aware path
* Seamless and complementary interworking between different intent-aware technologies
  * V/v is steered on a BGP intent-aware path that is itself resolved within domain 2 onto an SR Policy bound to the color of V/v
* Another benefit that can be extrapolated from the example is that intent-aware routes from different technologies may serve as alternative paths for the same intent.

6. Technical Requirements

6.1. Intent Requirements

The BGP Intent-Aware routing 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

Subsequent sections elaborate on these requirements.

6.1.1. Transport Network Intent 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.

The network diagram below illustrates the reference network topology used in this section

![Network Diagram](https://example.com/network_diagram.png)

Figure 9: Transport Network Intent Requirements Reference Diagram
The following network design assumptions apply to the reference topology above, as an example:

- Independent ISIS/OSPF SR instance in each domain.
- Peering links have equal cost metric.
- Peering links have delay configured or measured as shown by "D". D=50 for cross peering links.
- The cross links between ASBRs share the same risk.
- The top parallel link between 121-211 shares same risk with the link 122-212.
- The top parallel link between 231-321 shares same risk with the link 232-322.
- VPN service is running from PE31, PE32 to PE11, PE12 via service RRs (S-RRn in figure).

Intent-aware inter-domain routing information to end point E with intent C is represented using (C,E). The notation used is a representation of the intent-aware route using color, and does not indicate a specific protocol encoding.

The following sections illustrate requirements and provide detailed examples for several intent types.

6.1.1.1. Minimization of end-to-end metric

Various metric types can be advertised within an IGP domain and minimum metric paths can be computed within IGP domain, with Flex-Algo [I-D.ietf-lsr-flex-algo] for instance.

The BGP solution should allow the establishment of inter-domain intent-aware paths with low values of a metric type, accumulated over the end-to-end path.

In the reference topology of Figure 9

* 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 end-to-end path
  * Color C1 - Minimum cost intent.
  * Intent-aware route for C1 sets up path(s) between PEs for end-to-end minimum cost.
  * 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) intent-aware route via several equal paths:
    + One such path is through FA0 to node 121, links 121-211, FA0 to 231, link 231-321, FA0 to PE31
    + Another such path is through FA0 to node 122, link 122-212, FA0 to 232, link 232-322, FA0 to PE31.
      - PE11 may load-balance among these paths
  * On PE11, VPN routes from PE31 colored with C1 are steered via (C1, PE31) intent-aware route.

- Latency Optimized End-to-end path
  * Color C2 - Minimum latency intent.
  * BGP Intent-aware route for C2 advertises path(s) between PEs for end-to-end minimum delay.
  * These paths traverse over intra-domain Flex-Algo 128 in each domain and account for the peering link delay between ASBRs.
  * Example: PE11 learns (C2, PE31) intent-aware route and best path is through FA128 to node 122, link 122-212, FA128 to 232, link 232-322, FA128 to PE31.
  * On PE11, VPN routes from PE31 colored with C2 are steered via (C2, PE31) intent-aware route.
6.1.1.2. Exclusion/inclusion of link affinity

The Intent-aware BGP routing solution should allow the establishment of inter-domain paths that satisfy link affinity inclusion/exclusion constraints. The link affinity constraints should also be satisfied for inter-domain links, such as those between ASBRs.

Using the reference topology of Figure 7 for the example below:

- Color C3 - Intent to Minimize cost metric and avoid purple links
- 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.
- BGP intent-aware route 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) intent-aware route via 2 paths.
  - First path is through FA 129 to node 121, link 121-211, FA129 to 231, link 231-321, FA129 to PE31.
  - Second path is through FA129 to node 122, link 122-212, FA129 to 232, link 232-322, FA129 to PE31.
- On PE11, VPN routes from PE31 colored with C3 are steered via (C3, PE31) intent-aware route.

6.1.1.3. Exclusion/inclusion of nodes

Support creating an inter-domain path that includes or excludes a certain set of nodes in each domain.

Mechanisms used to achieve the node inclusion/exclusion constraints within different domains should be independent.

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.
The example below describes the details for Figure 9

- Color C4 - Intent to Minimize cost metric and avoid nodes
  * Each domain has Flex Algo 129 and Flex-Algo 129 is not enabled on nodes 121, 211, 231, 321
  * Flex Algo 129 definition is set to minimum cost metric
- Intent-aware route for C4 sets up paths between PEs for minimum end-to-end cost and avoiding specific nodes.
- These paths traverse over intra domain Flex Algo 129 in each domain and accounts for peering link cost between ASBR and avoiding specific nodes.
- Example: PE11 learns (C4, PE31) intent-aware route via 1 path.
  * The path is through FA129 to node 122, link 122-212, FA129 to 232, link 232-322, FA129 to PE31.
- On PE11, VPN routes colored with C4 are steered via (C4, PE31) intent-aware route.

6.1.1.4. Diverse Paths

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.

Support creation of inter-domain diverse paths that avoid shared risk links.

The example below describes the details for Figure 8

- Color C5 and C6 - Intent to create diverse paths avoiding common node, link and shared risk
  * Each domain has SRLG aware diverse path built as below
  * Domain 1: Color C5 -> PE11, 121
  * Color C6 -> PE12, 122
* Domain 2: Color C5 -> 211,231
* Color C6 -> 212,232
* Domain 3: Color C5 -> 321,PE31
* Color C6 -> 322,PE32

* Shared risk among inter-domain links is as described in the topology description
  * Intent-aware diverse paths represented by C5 and C6 setup in each domain
  * Local policies on inter-domain links to avoid common shared risk for intent C5 and C6
  * Example: PE11 learns (C5, PE31) intent-aware route via 1 path.
  * The path is through PE11,121-211 (bottom link), 231-321 (bottom link), PE31
    * Example: PE12 learns (C6, PE32) intent-aware route via 1 path.
    * The path is through PE12,122,212, 232,322, PE32

  o On PE11, VPN routes colored with C5 are steered via (C5, PE31) intent-aware route.
  o On PE12, VPN routes colored with C6 are steered via (C6, PE32) intent-aware route.

6.1.1.5. Applicability of intent to a subset of domains

Support creation of paths with certain intents applicable to only a subset of domains.

No constraint specific state on internal nodes where intent is not applicable.

The example below describes the details for Figure 9
  o Color C7 to exclude purple links

    * Purple links exist only in domain 2
* Intra-domain Intent-aware paths in domain 2 via 211,231

* Intra-domain paths for C7 not created in Domain 1 and Domain 3

- On PE11, VPN routes colored with C7 are steered via (C7, PE31) intent-aware route.

* Intent-aware route (C7,PE31) uses best effort paths in Domain1 and Domain3

* Intent-aware route (C7,PE31) uses intra-domain intent-aware path C7 in Domain2

6.1.1.6. Exclusion/inclusion of domain

![Domain Exclusion Diagram](image)

Figure 10: Domain Exclusion Diagram

Color C4 - Avoid sending selected traffic via Domain 3

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

6.1.1.7. Virtual network function chains in local and core domains

Figure 11: Transport NFV Diagram

- 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
E11 and E21 MAY be involved in inter-domain signalling in order to send service traffic towards PEs in remote domains. Different functions may be collocated at the same network node. (For example, PE functionality and NFV attachment functionality may be collocated.)

6.1.2. VPN (Service Layer) Network Intent Requirements

This section describes requirements and reference use-cases for extending intent-aware routing to the VPN (Service) layer. Details for this section will be added in the next revision.

6.2. Traffic Steering Requirements

Traffic arriving at an ingress PE for a colored service route gets steered into an intent-aware path to the egress PE. Section 5.1.9 illustrates the automated steering mechanism, driven through Color Extended Community in the service route.

- Flexible traffic steering is required, with support for different types:
  - Per-Destination Steering: Incoming packets are steered based on the destination address of the packets
  - Per-Flow Steering: Incoming packets are steered based on the destination address of the packets and additional fields in the packet header
    - DSCP for IPv4/IPv6 packets and EXP for MPLS packets
    - 5-tuple IP flow (Source address, destination address, source port, destination port and protocol fields).
- When no path that fulfills the desired intent is available:
  - An option of ordered fallback should be supported
    - via one or more alternative intents; or via a best-effort path.
  - An option of not using a fallback path for the service route should also be supported.
  - Fallback scheme per service route should be supported
    - Fallback schemes should be decoupled from primary. For example, different service routes using same primary but different fallback schemes.
Above steering mechanisms should be supported for any service, including L2/L3 VPNs and Internet/global routing.

6.3. Deployment Requirements

The solution must support the representative deployment designs and associated deployment requirements described in the following sub-sections.

6.3.1. Multi-domain deployment designs

This section describes four different ways that multi-domain networks could be organized. This is a representation of most common deployments and not an exhaustive coverage.

6.3.1.1. Multiple IGP domains within a single AS, inter-connected at border nodes

![Diagram of Transport Multiple Domains Network Diagram]

The above diagram shows three different IGP domains, Domain1, Domain2 and Domain3 inter-connected at the ABRs 121, 122, 231, 232.
This single-AS network uses I-BGP sessions, with ABRs acting as inline route reflectors to PEs.

Note that the IGP design included here and in other models below is illustrative. In practice, there may be multiple areas/levels or multiple IGP instances.

6.3.1.2. Multiple IGP domains within a single AS, with iBGP between border nodes

![Network Diagram]

The above diagram shows a single AS1 with three different IGP domains, Domain1, Domain2, and Domain3. 121,122,212,231,321,322 are border nodes for the IGP domains and they participate in only one IGP domain.

In this design, domain inter-connect is via iBGP peering links between Area border nodes. ABRs act as inline route reflectors to PEs.
6.3.1.3. Multiple ASes inter-connected with E-BGP between border nodes

![Network Diagram]

**Figure 14: Transport Multiple Domains with eBGP Network Diagram**

The above diagram shows three different ASes (AS1, AS2 and AS3.) 121,122, 211, 212, 231,232, 321,322 are border nodes between the ASes.

In this design, domain inter-connect is via eBGP peering links between AS border nodes. The ASBR also runs I-BGP sessions with other ASBRs or RRs in the same AS.

6.3.1.4. Multiple sites with same AS connected via different core AS
Figure 15: Transport Multiple Domains with same AS Network Diagram

121, 122, 231, 232 belong to AS2 only. AS1 and AS2 domains may run multi-instance IGP or different levels/areas.

This topology uses I-BGP sessions to some clients and E-BGP sessions to other nodes. When an RR is used between PEs in AS1 and ABRs in AS2, it will have iBGP sessions to clients in same AS and e-BGP sessions to nodes in other AS.

6.3.1.5. AS Confederations

BGP confederations [RFC 5065] allows the division of a public AS into multiple sub-ASes, usually with private identifiers. The solution should support BGP based intent-aware paths within the sub-AS or across the sub-ASes of the confederation, in any of the network designs described in sections 5.4.1.1 to section 5.4.1.4.

6.3.1.6. Transport Technologies

6.3.1.6.1. Unicast transport

The solution must support the following:

- End-to-end paths crossing transport domains that use different technologies and encapsulations, such as:
* LDP-MPLS
* RSVP-TE-MPLS
* SR-MPLS
* SRv6
* SR-TE (MPLS and SRv6)
* IGP Flex-Algo (MPLS and SRv6)
* Native IPv4/IPv6 forwarding (networks without MPLS enabled)

- Note:
  * All MPLS/SR-MPLS deployments may be IPv4/IPv6 or dual-stack
  * SR-TE includes color-only and other policies as defined in [I-D.ietf-spring-segment-routing-policy]

- Interworking between domains with different encapsulations (e.g. SR-MPLS and SRv6)

- Different transport encapsulations simultaneously within a domain, for co-existence and migration

6.3.1.6.2. Multicast transport

A routing solution for end-to-end intent-aware paths should support multicast as well as unicast. This section will be updated in the next revision of the document.

6.3.1.7. Co-existence, compatibility and interworking with existing intent-aware routing solutions

The BGP intent-aware routing solution MUST be compliant with the intent-aware routing framework described in Section 5.1.9. Specifically,

- It MUST support service routes using Color Extended-Community to request intent as defined in [I-D.ietf-spring-segment-routing-policy]
- It MUST support automated steering of colored service routes on a BGP intent-aware path using color
Intent-aware routes MAY resolve recursively via other intent-aware routes provided by any solution.

6.3.1.8. Co-existence and Interworking with BGP-LU

BGP-LU [RFC3107] is widely deployed to provide inter-domain best-effort connectivity across different domains. The BGP intent-aware routing solution should support:

- Establishment of best-effort paths by using a color to represent best-effort intent, to avoid the need to deploy both technologies.
- Co-existence of inter-domain BGP-LU and BGP intent aware routing in a network.
- Support interworking of BGP-LU and BGP intent-aware network domains.

6.3.1.9. Domains with different intent granularity

All domains in a network may not support the same number and granular definition of colors. However, the maximum granularity of colors should be provided for end to end paths that are set up for steering of a colored service route, with mapping from a more granular color to a less granular color where needed.

6.3.1.10. Co-existence with alternative solutions

Section 5 describes co-existence and interworking of the BGP intent aware routing solution with other existing intent-aware solutions.

Controller based approaches or other distributed TE solutions can also address the use-cases in this document.

The intent-aware routing solution should coexist with such alternative solutions.

- It should allow traffic to use paths created by an alternative solution.
- It should allow part of the inter-domain path to be created by an alternative solution.
- The routing solution may be used to provide backup paths for a primary path created by an alternative solution, or vice versa.
6.3.2. Scalability Requirements

6.3.2.1. Scale Requirements

- Support a massive scaled transport network
  - Number of Remote PE’s: \( \geq 300k \)
  - Number of Colors \( C \): \( \geq 5 \)

- Support a scalable MPLS dataplane solution

- Constraints that need to be addressed:
  - Typical inter-domain MPLS network designs (e.g. Seamless-MPLS) build hop-by-hop stitched MPLS LSPs towards every PE in the network. For the scale above, the number of forwarding entries required to represent each remote PE for each color will exceed the 1M MPLS label space limit.
  - PE and transit nodes may be devices with low FIB capacity.
  - Additionally, they may also have constraints on packet processing (e.g. label ops, number of labels pushed)

- To address these constraints:
  - The solution must support hierarchy in the forwarding plane E.g. via a label stack or a list of segments, such that no single node needs to support a data-plane scaling in the order of \((\text{Remote PE} \times C)\)
  - The solution should minimize state on border nodes in order to reduce label and FIB resource consumption, while taking into account packet processing constraints.

- Support ability to abstract the topology and network events from remote domains - for scale, stability and faster convergence.
  - E.g. contain the control plane propagation of a failure event for an ABR within its attached upstream domain.

- Support an Emulated-PULL model for the BGP signaling

PE nodes may be devices with limited CPU and memory. The state on a PE should be restricted to transport endpoints that it needs for service steering.
BGP Signaling is natively a PUSH model.

For comparison, 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.

Emulated-PULL refers to the ability for a BGP node PE1 to "subscribe" to (C, PE2) route such that only paths for (C, PE2) are signaled to PE1.

The requirements for an Emulated-PULL solution are as follows:

- The subscription and related filtering solution must apply to any BGP node.
- For transport routes, this means
  - Ability for a node (e.g. PE/ABR/ASBR) to signal interest for routes of specific colors.
  - Ability for a node (e.g ABR/ASBR) to propagate the subscription message.
  - PEs may choose to only learn routes that they need - e.g. remote VPN endpoints (PEs/VPN ASBRs) or transit nodes (ABRs/transport ASBRs).
  - ABR/ASBRs also only learn and propagate routes for which nodes within the local domain have expressed interest.
  - The requirements for VPN routes will be updated in the future version of the document.
- Automation of the subscription/filter route
  - 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 summarize the endpoints and thus request a number of endpoints for a particular intent in a single subscription route.
6.3.2.2. Scale Analysis

This section will be updated in the future revision of the document.

6.3.3. Network Availability Requirements

- A BGP intent-aware routing 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 section 5.4.1.1:
  - Failure of intra-domain links should limit loss of connectivity (LoC) to under 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 under 50ms
  - Failure of an ABR node (e.g. 121, 231) should limit LoC to under 1sec, or under 50ms depending on the network deployment scenario.
  - Failure of a remote PE node (e.g. PE31) should limit LoC to under 1sec, or under 50ms depending on the network deployment scenario and specific service failover requirements

- In the Inter-AS Option C VPN reference topology in Section 5.4.1.3:
  - Failure of intra-domain links should limit LoC to under 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 under 50ms
  - Failure of an ASBR node (e.g. 121, 211) should limit LoC to under 1sec, or under 50ms depending on the network deployment scenario.
* Failure of a remote PE node (e.g. PE31) should limit LoC to under 1sec, or under 50ms depending on the network deployment scenario and specific service failover requirements

* Failure of an external link (e.g. 121-211) should limit LoC to under 1sec, or under 50ms depending on the network deployment scenario.

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.

6.3.4. BGP Protocol Requirements

This section summarizes the key protocol requirements that should be addressed by the intent-aware BGP routing solution. While the context for several requirements has been discussed earlier in the document, this section emphasizes aspects pertinent to the protocol design.

The solution should support the following:

o Signaling and distribution of different Intent-aware routes to reach a participating node, e.g. a PE. Intent must be indicated by the notion of a Color as defined in [I-D.ietf-spring-segment-routing-policy]

* Signal different instances of a prefix, one route per color

* Signal intent (color) associated with each route

* At any BGP hop, allow propagating the best path selected for each route, or additional paths

* Generate routes sourced from IGP-FA, SR-TE Policies, RSVP-TE and BGP-LU from a domain

o Path selection for Intent-aware routes

* Accumulation of intent specific metric at each BGP hop and compare the accumulated metric across all received paths at intermediate hops and at an ingress PE.

* Ability to load balance among multiple received paths at intermediate BGP hops and at an ingress PE

* Backup path installation for fast convergence at intermediate BGP hops and at an ingress PE
o Validation of received paths
  * Resolvability of next-hop in control plane
  * Availability of encapsulation in data plane

o Next-hop resolution for BGP Intent-aware route
  * Flexibility to use different intra-domain and inter-domain mechanisms, both intent-aware and traditional
    + IGP-FA, SR-TE, RSVP-TE, IGP, BGP-LU etc.
  * Recursive resolution over other BGP Intent-Aware routes
  * Recursive resolution via alternative color or best-effort paths when a particular intent is not available in a domain

o Flexible, efficient, extensible protocol definition
  * As an example for context, currently deployed mechanisms such as BGP-LU (RFC 8277) were designed for MPLS, hence only signal per prefix label(s) in NLRI. However, RFC9012 and RFC8669 have described extensions to BGP to signal multiple encapsulations, though in BGP attributes. The target deployments for intent-aware routing need to support additional transport as described in section 6.3.1.6.1. In addition, they also need to support a significantly higher targeted scale as described in scaling requirements.
    * Hence, the protocol definition should
      + Support efficient signaling of different transport encapsulations
      + Support efficient signaling multiple encapsulations for co-existence and migration between encapsulations
      + Accommodate efficiency of processing and future extensibility

o Separation of transport and VPN service semantics
  * Allow for different route distribution planes or processing for service vs transport routes

o Signaling across domains with different color mappings for a given intent
6.3.5. Multicast Intent Requirements

This section will be updated in the future revision of the document.

6.3.6. OAM Requirements

This section will be updated in the future revision of the document.

7. Backward Compatibility

This section will be updated in the future version of the document.

8. Security Considerations

This section will be updated in the future version of the document.

9. IANA Considerations

This section will be updated in the future version of the document.

10. Acknowledgements

The authors would especially like to thank Joel Halpern for his guidance on the collaboration work that has produced this document and feedback on many aspects of the problem statement.

We would like to thank Daniel Voyer, Robert Raszuk, Kireeti Kompella, Ron Bonica, Krzysztof Szarkowicz, Julian Lucek, Ram Santhanakrishnan, Stephane Litkowski for discussions and inputs.

We also express our appreciation to Hannes Gredler Simon Spraggs, Jose Liste and Jiri Chaloupka for discussions that have helped provide input to the problem statement.

Many thanks to Colby Barth, John Scudder, Kamran Raza, Kris Michelson, Huaimo Chen for their review and valuable suggestions.

11. Contributors

1. Kaliraj Vairavakkalai
   Juniper Networks
   kaliraj@juniper.net

2. Jeffrey Zhang
   Juniper Networks

12. References

12.1. Normative References


12.2. Informative References


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

[I-D.ietf-idr-performance-routing]

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

[I-D.ietf-lsr-flex-algo]

[I-D.ietf-lsr-flex-algo-bw-con]

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

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

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

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

[I-D.kaliraj-idr-bgp-classful-transport-planes]  

[I-D.voyer-pim-sr-p2mp-policy]  

[I-D.zhang-bess-bgp-multicast]  

[RFC3630]  

[RFC3906]  

[RFC4271]  

[RFC4272]  


Authors’ Addresses

Shraddha Hegde (Editor)
Juniper Networks Inc.
Exora Business Park
Bangalore, KA  560103
India

Email: shraddha@juniper.net
MPLS Extension Header Encodings
draft-jags-mpls-ext-hdr-00

Abstract

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

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 September 2022.
Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the
document authors. All rights reserved.

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

Table of Contents

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

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

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

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

1. Requirements

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

1. MPLS packet to carry additional data in the MPLS label stack to influence forwarding. This can be of two types:
   1a. Forwarding Instruction Flags (FIF) that does not use additional data.
   1b. Forwarding Instruction (FI) that needs additional data.
2. MPLS packet to carry additional data after the Bottom of the MPLS Label Stack.
3. Any combination of (1) and (2) in the same MPLS packet.

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

2. Conventions Used in This Document

2.1. Requirements Language

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

2.2. Terminology

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

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

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

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

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

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

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

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

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

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

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

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

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

MPLS (Multiprotocol Label Switching): Multiprotocol Label Switching.

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

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

TC (Traffic Class): Traffic Class.

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

3. Overview

Extending existing MPLS Header needs two main parts.

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

  Option 1. MEI by extending ELI/EL

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

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

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

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

Figure 1: MPLS Label Format

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

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

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

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

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

3.1. Option 1 - ELC as MPLS Extension Header Indicator

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

0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Entropy Label Indicator (7) | TC |S|      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Slice ID     |  Entropy Label        | IL |S|  ELC (SPI=1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

Figure 2: ELI/EL Packet Format

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

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

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

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

3.1.1. Advantages with ELC

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

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

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

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

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

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

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

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MEI=SPL (TBA1) | IL |S| IPI,BPI,HBI |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: New SPL as MPLS Extension Header Indicator

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

3.3. Option 3 - NPL as MPLS Extension Header Indicator

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

```
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MEI=NPL | IL |S| IPI,BPI,HBI |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

4. In-Stack MPLS Extension Header Encoding

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

```
+-----------------------------+-----------------------------+
|      Entropy Label or SPL or NPL      | IL=1|S| IPI=1         |
+-----------------------------+-----------------------------+
|  IS-FI Opcode |    In-Stack Data      |R|D|E|S| In-Stack Data |
+-----------------------------+-----------------------------+
```

Figure 5: In-Stack Extension Header Format

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

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

Label Field:

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

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

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

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

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

1. Define the Data format encoded in the MPLS extension header.
2. Define the Hop-By-Hop or Edge-To-Edge (only on the decapsulation node) processing scope.
3. The Hop-By-Hop IS-FI opcodes MUST be placed before the Edge-To-Edge IS-FI Opcodes in the MPLS Extension Header of the packet to optimize the Hop-By-Hop processing in hardware.

TC Field:

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

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

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

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

TTL Field:

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

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

5. Bottom Of Stack MPLS Extension Header Encoding

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

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

Figure 6: BOS Extension Header Format

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

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

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

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

Table 2: BOS MPLS Extension Header Format

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

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

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

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

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

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

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

```
+-----------------------------+-----------------------------+
|     Entropy Label or SPL or NPL      | IL=1|0|   IPI=1       |
+-----------------------------+-----------------------------+
|IS-FI Opcode=1 | Flags                |R|1|E|1|   Flags       |
+-----------------------------+-----------------------------+
```

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

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

Label Field:

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

TC Field:

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

TTL Field:

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

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

```
+-----------------------------+-----------------------------+
|     Entropy Label or SPL or NPL      | IL=2|0|   IPI=1       |
+-----------------------------+-----------------------------+
|IS-FI Opcode=1 | Flags                |R|1|E|1|   Flags       |
|1|   Flags                           |R|1|E|1|   Flags       |
+-----------------------------+-----------------------------+
```
More than 20 bits of data can be encoded as part of IS-FI opcode. In this specific case, the FI flags which are more than 20 bits are encoded in next 4 bytes of the MPLS header.

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

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

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

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

Figure 9: Example In-Stack Extension Header Carrying FI with the data

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

Label Field:

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

TC Field:

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

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

TTL Field:

8-bit field is used to encode the In-Stack data apart from 12-bit Label field.
Figure 10: Example In-Stack Extension Header Carrying FI with the data more than 20 bits

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

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

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

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

Figure 11: Example BOS Extension Header Carrying FI with data

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

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

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

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

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

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

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

10. Node Capability Signaling

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

11. Security Considerations

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

12. Backward Compatibility

12.1. Backward Compatibility With ELC as MEH Indicator

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

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

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

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

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

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

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

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

13. Processing In-Stack MPLS Extension Header

Encapsulating Node:

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

* SHOULD NOT change the IS-FI Opcode and the first 12 bits of the In-Stack Data for the same packet flow avoid ECMP path change.

* MAY change In-Stack data part present only in the TTL field for the same packet flow.

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

Intermediate Node:

* MUST ignore the IS-FI Opcode that are not supported.

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

* SHOULD NOT change the IS-FI Opcode and the first 12 bits of the In-Stack Data for the same packet flow.

* MAY change In-Stack data part present only in the TTL field for the same packet flow.

* MAY remove the IS-FI opcode and its corresponding data for all matching packet flow.

Decapsulating Node:
14. Processing BOS MPLS Extension Header

Encapsulating Node:

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

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

Intermediate Node:

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

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

Decapsulating Node:

* MUST remove the BOS MPLS Extension header.

15. IANA Considerations

Below are the IANA actions which this document is requesting.

15.1. IANA Considerations for Forwarding Instruction Flags

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

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

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

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

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

Table 4: In-Stack Forwarding Instruction Opcode Registry

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

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

Table 5: In-Stack Forwarding Instruction Opcode Values

15.3. IANA Considerations for BOS-FI Opcode

IANA is requested to create a new registry to assign BOS-FIOC opcode values.
Table 6: Bottom-Of-Stack Forwarding Instruction Opcode Registry

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid value</td>
<td>This document</td>
</tr>
<tr>
<td>255</td>
<td>Opcode Range Extension Beyond 255</td>
<td>This document</td>
</tr>
</tbody>
</table>

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

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

15.4. IANA Considerations for New Special Purpose Label

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

16. Appendix

16.1. Alternate approach for In-Stack Extension Header Encoding

In the above In-Stack Extension Header Encoding the Label field is used to encode the FI Opcode. So just for completeness, here is the alternate way of In-Stack Extension Header Encoding is provided.
IPI flag is set to "1" to indicate the presence of In-Stack MPLS Extension Header.

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

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

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

TC Field:
This carries data stacking bits. They are as follows:

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

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

TTL Field:
This carries In-Stack Forwarding Instruction opcode.

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

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

The TTL field in the SPL (value TBA1) is used to encode FI Flags including IPI, HBI and BPI flags defined in this document.
The FI Opcode value 3 as an example indicates encoding of Entropy Label and Slice ID as shown in the above Figure.

16.3. MPLS BOS Extension Header Example with IOAM Data Fields

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

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

17.1. Normative References


17.2. Informative References


Acknowledgments

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

Contributors

TBD

Authors’ Addresses

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

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

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

Bruno Decraene
Orange
Email: bruno.decuraene@orange.com

This document defines extensions to Border Gateway Protocol (BGP) to distribute SR policies carrying headend behavior.

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 December 10, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with
1. Introduction

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

As described in [I-D.ietf-spring-segment-routing-policy], a headend can steer a packet flow into an SR Policy in various ways, including BSID steering, per-destination steering, per-flow steering, and policy-based steering. Moreover, [I-D.jiang-idr-ts-flowspec-srv6-policy] describes a way by using BGP FlowSpec to steer packets into an SRv6 Policy.

inserts a new SRH in between the IPv6 Header and the received SRH rather than pushing a new IPv6 header, which can be applied to express scalable traffic-engineering policies across multiple domains.

The SRv6 Binding SID sub-TLV is defined in [I-D.ietf-spring-segment-routing-policy] to signal the SRv6 BSID information along with SR Policies. It enables the specified SRv6 BSID behavior to be instantiated on the headend node. However, if the packets are steering into an SR Policy in some other way than using BSID, the headend behavior is not specified during the distributing of SR Policy by BGP. The network operator has to use additional tools, like NETCONF, to signal the headend behavior.

This document defines extensions to Border Gateway Protocol (BGP) to distribute SR policies carrying headend behavior. So that the headend can be instructed to perform specific behavior when packets are steered into the SR policy without BSID.

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. Headend Behavior in SR Policy

As defined in [I-D.ietf-idr-segment-routing-te-policy], the SR policy encoding structure is as follows:
SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
  Tunnel Encaps Attribute (23)
  Tunnel Type: SR Policy
    Binding SID
    SRv6 Binding SID
    Preference
    Priority
    Policy Name
    Policy Candidate Path Name
    Explicit NULL Label Policy (ENLP)
    Segment List
      Weight
      Segment
      Segment
      ...

SR policy with headend behavior is expressed as follows:

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
  Tunnel Encaps Attribute (23)
  Tunnel Type: SR Policy
    Binding SID
    SRv6 Binding SID
    Preference
    Priority
    Policy Name
    Policy Candidate Path Name
    Explicit NULL Label Policy (ENLP)
    Headend Behavior
    L2 Headend Behavior
    Segment List
      Weight
      Segment
      Segment
      ...

2.1. Headend Behavior Sub-TLV

The Headend Behavior sub-TLV encodes the default headend behavior associated with the candidate path for L3 traffic. When the headend steers L3 packets into that SR Policy and the associated candidate path is active, the specific headend behavior should be performed by default. In the case of BSID steering, the behavior defined by the BSID overrides the default headend behavior.
The Headend Behavior sub-TLV is optional, and MUST NOT appear more than once in the SR Policy encoding.

The Headend Behavior sub-TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     |            RESERVED           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Headend Behavior       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where:

- **Type**: to be assigned by IANA.
- **Length**: 4.
- **RESERVED**: 2 octets of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **Headend Behavior**: a 2-octet value. The following values are defined.
  - * TBD: H.Encaps. A headend behavior defined in [RFC8986].
  - * TBD: H.Insert. A headend behavior defined in [I-D.filsfils-spring-srv6-net-pgm-insertion].

### 2.2. L2 Headend Behavior Sub-TLV

The L2 Headend Behavior sub-TLV encodes the default headend behavior associated with the candidate path for L2 traffic. When the headend steers L2 packets into that SR Policy and the associated candidate path is active, the specific headend behavior should be performed by default.

The L2 Headend Behavior sub-TLV is optional, and MUST NOT appear more than once in the SR Policy encoding.

The L2 Headend Behavior sub-TLV has the following format:
where:

- **Type**: to be assigned by IANA.
- **Length**: 4.
- **RESERVED**: 2 octets of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **L2 Headend Behavior**: a 2-octet value. The following values are defined.
  * TBD: H.Encaps.L2. A headend behavior defined in [RFC8986].

3. Extensions of BGP-LS

[I-D.ietf-idr-te-lsp-distribution] describes a mechanism to collect the SR policy information that is locally available in a node and advertise it into BGP-LS updates. Extensions of BGP-LS for headend behavior of SR Policy will be included in the future version of this draft.

4. Security Considerations

Procedures and protocol extensions defined in this document do not affect the security considerations discussed in [I-D.ietf-idr-segment-routing-te-policy].

5. IANA Considerations

Headend Behavior Sub-TLV (TBD)

L2 Headend Behavior Sub-TLV (TBD)
6. References

6.1. Normative References


6.2. Informative References


Authors’ Addresses

Changwang Lin
New H3C Technologies
Email: linchangwang.04414@h3c.com

Wenying Jiang
China Mobile
Email: jiangwenying@chinamobile.com

Yisong Liu
China Mobile
Email: liuyisong@chinamobile.com

Mengxiao Chen
New H3C Technologies
Email: chen.mengxiao@h3c.com

Hao Li
New H3C Technologies
Email: lihao@h3c.com
Abstract

Segment Routing is a source routing paradigm that explicitly indicates the forwarding path for packets at the ingress node. An SR Policy is a set of candidate paths, each consisting of one or more segment lists. This document defines extensions to BGP SR Policy to specify the identifier of segment list.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on September 29, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

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

However, there is no identifier for segment list in BGP SR Policy, which may cause inconvenience for other mechanisms to designate segment lists distributed by BGP.

For example, a network controller distributes SR policies to the headend nodes, and the headend nodes collect traffic forwarding statistics per segment list. When a headend node report each statistic to the controller, it needs to specify the segment list which the statistic belongs to. Due to the lack of identifier, the headend node usually reports all SIDs in the associated segment list along with the statistic, and the controller needs to distinguish the segment list by comparing the SIDs one by one. The advertisement
of all SIDs in the segment list consumes a lot of octets, and the comparison of SIDs can be complicated.

For another example, a network controller distributes SR policies using BGP, and then it uses NETCONF to set some configurations of the segment lists, which are not suitable to be carried in BGP. So the controller needs to specify the segment list which the configurations belong to. In this case, a simple identifier of segment list can also be helpful.

This document defines extensions to BGP SR Policy to specify the identifier of segment list.

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. Segment List Identifier in SR Policy

As defined in [I-D.ietf-idr-segment-routing-te-policy], the SR policy encoding structure is as follows:

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
  Tunnel Encaps Attribute (23)
    Tunnel Type: SR Policy
    Binding SID
    SRv6 Binding SID
    Preference
    Priority
    Policy Name
    Policy Candidate Path Name
    Explicit NULL Label Policy (ENLP)
    Segment List
      Weight
      Segment
      Segment
      ...
      ...

SR policy with segment list identifier is expressed as below:
SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
  Tunnel Encaps Attribute (23)
  Tunnel Type: SR Policy
  Binding SID
  SRv6 Binding SID
  Preference
  Priority
  Policy Name
  Policy Candidate Path Name
  Explicit NULL Label Policy (ENLP)
  Segment List
    Weight
    Segment List Identifier
    Segment
    Segment
    ...

The segment list identifier can be advertised using the Segment List ID sub-TLV or the Segment List Name sub-TLV, as defined in Section 2.1 and 2.2.

2.1. Segment List ID Sub-TLV

The Segment List ID sub-TLV specifies the identifier of the segment list by a 4-octet number.

The Segment List ID sub-TLV is optional and it MUST NOT appear more than once inside the Segment List sub-TLV.

The Segment List ID sub-TLV has the following format:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Length      |     Flags     |   RESERVED    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Segment List ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

where:

  o Type: TBD.

  o Length: 6.
```
2. Segment List ID Sub-TLV

The Segment List ID sub-TLV is a 4-octet identifier for the segment list. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

2.2. Segment List Name Sub-TLV

The Segment List Name sub-TLV specifies the identifier of the segment list by a symbolic name. It is optional and MUST NOT appear more than once inside the Segment List sub-TLV.

The Segment List Name sub-TLV has the following format:

```
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +---------------------------------------------+
       |     Type      |   Length      |     Flags     |   RESERVED    |
       +---------------------------------------------+
```

where:

- **Type**: TBD.
- **Length**: Variable.
- **Flags**: 1 octet of flags. None are defined at this stage. Flags SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **Reserved**: 1 octet of reserved bits. SHOULD be set to zero on transmission and MUST be ignored on receipt.
- **Segment List Name**: Symbolic name for the segment list. It SHOULD be a string of printable ASCII characters, without a NULL terminator.

3. Security Considerations

TBD
4. IANA Considerations

Segment List ID sub-TLV and Segment List Name sub-TLV (TBD)

5. References

5.1. Normative References


[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017


5.2. Informative References

Authors’ Addresses

Changwang Lin
New H3C Technologies

Email: linchangwang.04414@h3c.com

Mengxiao Chen
New H3C Technologies

Email: chen.mengxiao@h3c.com
Encapsulation of BFD for SRv6 Policy
draft-liu-bfd-srv6-policy-encap-01

Abstract

Bidirectional Forwarding Detection (BFD) mechanisms can be used for fast detection of failures in the forwarding path of SR Policy. This document describes the encapsulation of BFD for SRv6 Policy, which can be applied for both S-BFD and U-BFD. The BFD packets may be encapsulated in transport mode or tunnel mode.

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 January 11, 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents
1. Introduction

Segment Routing (SR) [RFC8402] allows a headend node to steer a packet flow along any path. Per-path states of intermediate nodes are eliminated thanks to source routing. A Segment Routing Policy (SR Policy) [I-D.ietf-spring-segment-routing-policy] is an ordered list of segments (i.e., instructions) that represent a source-routed policy. The packets steered into an SR Policy carry an ordered list of segments associated with that SR Policy. The SRv6 Policy is the instantiation of SR Policy for SR over IPv6 (SRv6) data-plane.

In order to provide end-to-end protection, the headend node need to rapidly detect any failures in the forwarding path of SR Policy, so that it could switch from the active candidate path to another backup candidate path within the same SR Policy or switch from the active SR Policy to another backup SR Policy. Bidirectional
Forwarding Detection (BFD) mechanisms [RFC5880] [RFC7880] can be used for fast detection of such failures.

As described in [I-D.ali-spring-bfd-sr-policy], the BFD mechanism to be used for monitoring SRv6 Policies is expected to be simple and lightweight, which can be setup and deleted dynamically and on-demand. S-BFD simplifies the mechanism for using BFD with a large proportion of negotiation aspects eliminated. [I-D.ali-spring-bfd-sr-policy] describes the use of S-BFD for monitoring of SR Policy paths, and specifies the S-BFD Discriminator selection, the S-BFD session initiation and the return path control related to S-BFD use for SR Policy.

Unaffiliated BFD Echo Function (U-BFD) [I-D.ietf-bfd-unaffiliated-echo] simplifies the BFD Echo Function procedure, by which the remote system does not need to support BFD or maintain any BFD session states. U-BFD may also be used to monitor SR Policies.

This document describes the encapsulation of BFD for SRv6 Policy, which can be applied for both S-BFD and U-BFD. The BFD packets may be encapsulated in transport mode or tunnel mode.

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. Encapsulation of BFD Packet for SRv6 Policy

On SRv6 data-plane, a BFD packet for SRv6 Policy carries a Segment Routing Header (SRH) [RFC8754] containing a list of SRv6 SIDs associated with that SRv6 Policy.

The BFD packets may be encapsulated in transport mode or tunnel mode. In transport mode, the SRH is inserted after the IPv6 header. In tunnel mode, an outer IPv6 header with an SRH is encapsulated, which looks like an BFD packet for plain IPv6 is steered into an SRv6 Policy.
An SRv6 Policy may consist of multiple candidate paths, and each candidate path may consist of multiple Segment-Lists. To monitor a candidate path, an implementation may setup multiple sessions for each Segment-List associated with that path. If some of the Segment-Lists fail, the forwarding will be weighted load-balancing among the other Segment-Lists. If all of the Segment-Lists fail, the candidate path is deemed to be failed. An implementation may monitor each candidate path belonging to the SRv6 Policy respectively. If the active candidate path fails, the switchover to another backup candidate path will be triggered. If all the candidate paths fails, the SRv6 Policy is deemed to be failed. How to setup sessions for the candidate paths and Segment-Lists associated with an SRv6 Policy is out of the scope of this document.

2.1. Control Packet in Transport Mode

In transport mode, the encapsulation format of BFD control packet is as follows:

```
+-------------+---------+-------------+------------+------------+
| IPv6 header |   SRH   | UDP Header  |  Payload   |
+-------------+---------+-------------+------------+------------+
```

Figure 1: Transport Mode Encapsulation

```
+-------------+---------+-------------+------------+------------+
| IPv6 header |   SRH   | IPv6 header | UDP Header |  Payload   |
+-------------+---------+-------------+------------+------------+
```

Figure 2: Tunnel Mode Encapsulation
In the SRH, the first element of the Segment List (Segment List[0]) contains the SRv6 SID or IPv6 address of the tail-end node.

If the last segment of the SRv6 Policy Segment-List does not belong to the tail-end node, an IPv6 address of tail-end should be added as Segment List[0], while Segment List[1] contains the last segment of the SRv6 Policy Segment-List. The typical scenarios are as follows:

- The last segment of the SRv6 Policy Segment-List may be an End.X SID of the penultimate hop. If it is used as Segment List[0], the final destination for the BFD packet is missing.

- The last segment of the SRv6 Policy Segment-List may be a Binding SID, for example, the application of SRv6 Policy for L3VPN service across multiple domains. If it is used as Segment List[0], according to [RFC8986], the node which instantiates the BSID will not perform the encapsulation behavior of the associated SRv6 Policy, but stop processing the SRH and proceed to process the next header in the packet.

Else, the additional tail-end IPv6 address is not necessary, and it can be omitted in order to reduce the SRH size.
After tail-end receives the control packet, it will send response packet back to the headend. The response packet is IP routed based on the IPv6 SA of the control packet from headend. Additional measures may be taken to control the forwarding path of response packet, which is out of the scope of this document.

2.2. Echo Packet in Transport Mode

In transport mode, the encapsulation format of BFD echo packet is as follows:

```
+-----------------------------------------------------------+
| IPv6 Header                                               |
| . Source IP Address = Headend IPv6 Address                 |
| . Destination IP Address = Segment List[SL]                |
| . Next-Header = SRH                                        |
| .                                                           |
+-----------------------------------------------------------+
| SRH                                                       |
| . Segment List[0] = Headend IPv6 Address                   |
| . Segment List[1]                                          |
| . Segment List[2]                                          |
| . ...                                                     |
| . Next-Header = UDP                                        |
| .                                                           |
+-----------------------------------------------------------+
| UDP Header                                                |
| .                                                           |
+-----------------------------------------------------------+
| Payload                                                   |
| .                                                           |
+-----------------------------------------------------------+
```

Figure 4: Format of Echo Packet in Transport Mode

The BFD echo packet u-turns on the tail-end node and returns to the headend node. The difference from the control packet is that the final destination of the echo packet is the headend itself. So, Segment List[0] in the SRH should contain the IPv6 address of the headend, in order to indicate the tail-end to forward the echo packet back to headend. The return path is IP routed.

In both S-BFD and U-BFD for SRv6 Policy, the echo packet may be used to control the return path. After the echo packet reaches the tail-end along the forwarding path of SRv6 Policy Segment-List, additional segments will indicate the packet to be forwarded along specific path back to the headend.
If segments of the return path is included in the SRH of echo packet and the last segment of the return path belongs to the headend, the additional headend IPv6 address is not necessary to be added as Segment List[0]. How to identify corresponding segment stack for the return paths are outside the scope of this document.

2.3. Control Packet in Tunnel Mode

In tunnel mode, the encapsulation format of BFD control packet is as follows:

```
+-----------------------------------------------+
| IPv6 Header                                  |
|   Source IP Address = Headend IPv6 Address    |
|   Destination IP Address = Segment List[SL]   |
|   Next-Header = SRH                          |
|                                              |
+-----------------------------------------------+
| SRH                                          |
|   Segment List[0] = Tail-end IPv6 Address, or |
|     Last Segment of SRv6 Policy Segment-List  |
|     Segment List[1]                          |
|     Segment List[2]                          |
|     ...                                      |
|   Next-Header = IPv6                         |
|                                              |
+-----------------------------------------------+
| IPv6 Header                                  |
|   Source IP Address = Headend IPv6 Address    |
|   Destination IP Address = Tail-end IPv6 Address |
|   Next-Header = UDP                          |
|                                              |
+-----------------------------------------------+
| UDP Header                                   |
|                                              |
+-----------------------------------------------+
| Payload                                      |
|                                              |
+-----------------------------------------------+
```

Figure 5: Format of Control Packet in Tunnel Mode

In the SRH, the first element of the Segment List (Segment List[0]) contains the SRv6 SID or IPv6 address of the tail-end node.

If the last segment of the SRv6 Policy Segment-List does not belong to the tail-end node and its function does not include decapsulation of the outer IPv6 header, an IPv6 address of tail-end should be
added as Segment List[0], while Segment List[1] contains the last segment of the SRv6 Policy Segment-List. The typical scenarios are as follows:

- The last segment of the SRv6 Policy may be an End.X SID of the penultimate hop. If it is used as Segment List[0], the penultimate hop needs to remove the outer IPv6 header with all SRH, and forwards the inner IPv6 packet to reflector. If the last segment is with Ultimate Segment Decapsulation (USD) flavor, the penultimate SR endpoint node will perform such decapsulation as defined in [RFC8986]. Otherwise, how to process the packet when the upper-layer header type is IPv6, is not clearly defined in [RFC8986]. It depends on implementation, and may not work well for BFD.

- The last segment of the SRv6 Policy may be a Binding SID, which is the same with the Binding SID case in section 2.1. Else, the additional tail-end IPv6 address is not necessary, and it can be omitted in order to reduce the SRH size.

After tail-end receives the control packet, it will send response packet back to the headend. The response packet is IP routed based on the inner IPv6 SA of the control packet from headend. Additional measures may be taken to control the forwarding path of response packet, which is out of the scope of this document.

2.4. Echo Packet in Tunnel Mode

In tunnel mode, the encapsulation format of BFD echo packet is as follows:
IPv6 Header
  . Source IP Address = Headend IPv6 Address
  . Destination IP Address = Segment List[SL]
  . Next-Header = SRH

SRH
  . Segment List[0] = Tail-end IPv6 Address, or
    Last Segment of SRv6 Policy Segment-List
  . Segment List[1]
  . Segment List[2]
  . ...
  . Next-Header = IPv6

IPv6 Header
  . Source IP Address = Headend IPv6 Address
  . Destination IP Address = Headend IPv6 Address
  . Next-Header = UDP

UDP Header

Payload

Figure 6: Format of Echo Packet in Tunnel Mode

If the last segment of the SRv6 Policy Segment-List does not belong to the tail-end node, an IPv6 address of tail-end should be added as Segment List[0], in order to guarantee that the packet can reach the tail-end.

After the tail-end receives the control packet, it decapsulates the outer IPv6 header with SRH, and then forwards the inner IPv6 packet back to the headend based on the IPv6 DA.

If additional segments of the return path are included in the SRH of echo packet, the tail-end IPv6 address should not be included in the SRH. The segment stack should guarantee that the packet can reach the tail-end and then goes back to the headend. How to identify corresponding segment stack for the return paths are outside the scope of this document.
3. Choice of Headend and Tail-end IPv6 Addresses

3.1. Control Packet

When traffics are steered into an SRv6 Policy, the headend encapsulates the received packets in an outer IPv6 header along with an SRH. The Source Address of the outer IPv6 header is an IPv6 Address of the headend itself which can be routed. It may be a local interface address of the headend used for all SRv6 Policies. Or, different source addresses may be allocated per SRv6 Policy by local configuration.

For the BFD control packet, the headend IPv6 address in the Source Address of IPv6 header may use the source address associated with the SRv6 Policy.

An SRv6 Policy is identified through the tuple <headend, color, endpoint>. The endpoint indicates the destination of the policy, and is usually specified as an IPv6 address of the tail-end node.

For the BFD control packet, the headend may choose endpoint of the SRv6 Policy to be the tail-end IPv6 address which appears in the first segment of SRH or DA of inner IPv6 header, without additional knowledge of the tail-end. However, in some scenarios, the endpoint of SRv6 Policy can be the unspecified address (:: for IPv6), and the tail-end IPv6 Address may be specified by local configuration or network controller.

3.2. Echo Packet

For the BFD echo packet, the headend IPv6 address in the Source Address of IPv6 header may use the source address associated with the SRv6 Policy, which is similar with the control packet.

Because the echo packet u-turns on the tail-end, the tail-end does not need to parse the packet or use the source address as the destination address to send back, which is different from the control packet. So, the SA of echo packet is not necessary to be routable. An implementation may use unreachable address as the headend IPv6 address in the SA, which would prevent icmpv6 messages from flooding into the headend in failure cases.

For the headend IPv6 address which appears in the first segment of SRH or DA of inner IPv6 header of the echo packet, it should be an IPv6 address belonging to the headend and can be routed. An implementation may use the source address associated with the SRv6 Policy. Or, particular addresses may be allocated per SRv6 Policy by
local configuration, in order to distinguish the BFD echo packets for different SRv6 Policies.

4. Checksum in UDP Header

The computation of Checksum in UDP header includes the Destination Address of IPv6 header.

In the encapsulation of transport mode, the IPv6 DA may change along the SRv6 forwarding path. When computing the UDP Checksum, the headend should use the first segment in the SRH as the IPv6 DA. It is consistent with the packet received by the final destination, the tail-end node for control packet or the headend node for echo packet. So, when the final destination processes the UDP header, the verification of Checksum will be passed.

In the encapsulation of transport mode, the computation of UDP Checksum only involves the inner IPv6 header, which does not change en route. No additional action needs to be taken.

5. Control of Inserting Additional IPv6 Address in SRH

An implementation may have local configurations to control whether to insert a headend or tail-end IPv6 address as the first segment in the SRH. Or, an implementation may always insert additional IPv6 address.

6. Example

In the following network, the headend A installs an SRv6 Policy to tail-end D with the segment list <SID-A1, SID-B1, SID-C1, SID-D2>. SID-A1, SID-B1 and SID-C1 are all SRv6 End.X SIDs.

A-------------B-------------C-----------D

Figure 7: example network

Assume that A uses SBFD to monitor that SRv6 Policy. A may send SBFD control packet in transport mode, as shown in Figure 8.
Assume that A uses U-BFD to monitor that SRv6 Policy. A may send U-BFD echo packet in tunnel mode, as shown in Figure 9.
7. Security Considerations

TBD.

8. IANA Considerations

This document has no IANA actions.
9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Weiqiang Cheng
China Mobile
China
Email: chengweiqiang@chinamobile.com

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com

Mengxiao Chen
New H3C Technologies
China
Email: chen.mengxiao@h3c.com

Min Xiao
ZTE Corp.
China
Email: xiao.min2@zte.com.cn
Network Resource Partition Identifier (NRP-ID) in SRv6 segment
draft-liu-spring-nrp-id-in-srv6-segment-00

Abstract
This document proposes a method to carry the Network Resource Partition Identifier (NRP-ID) with the packet in the SRv6 segment.

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on January 11 2023.

Copyright Notice
Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Network slicing provides the ability to partition a physical network into multiple isolated logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers. [I-D.ietf-teas-ietf-network-slices] defines the term "IETF Network Slice" and establishes the general principles of network slicing in the IETF context. [I-D.cheng-teas-network-slice-ucase] describes several use cases of IETF Network Slice. [I-D.ietf-teas-ns-ip-mpls] proposes a solution to realize network slicing in IP/MPLS networks. Network nodes need to identify a packet belonging to a network slice before it can apply the proper forwarding treatment, so a slice ID must be carried in each packet.

Packets belong to a network slice need to be forwarded using the specific network resources. [I-D.draft-ietf-teas-ietf-network-slices]
slices] defines the network resource mapped to the network slice as NRP, that is, the Network Resource Partition, and defines the NRP-ID to identify the NRP used in the forwarding process.

In a network that provides slicing services, the NRP-ID can be carried in the packet. In the process of packet forwarding, the routers on the forwarding path can extract NRP-ID from the packet, determine the NRP to which the packet belongs, and then forward the packet using the resources associated with the NRP.

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

When SRv6 network provides network slicing service, it is also necessary to consider how to carry NRP-ID with packet. This document proposes a method to carry the NRP-ID in the SRv6 network. By setting the NRP-ID in the SRv6 segment, the SRv6 endpoint or transit node can be aware the NRP to which the packet belongs and carry out relevant forwarding processing.

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. Encoding NRP-ID in SRv6 segment

The structure of SRv6 segment is defined in RFC[8986]. An SRv6 segment consists of three parts, LOC:FUNCT:ARG.

+---------------------------------------------------------------+-----+
| Locator | Function | Argument |
+---------------------------------------------------------------+-----+
| LOC -----><- FUNCT ->< ARG -> |

Figure 1: structure of segment

After the packet enters the SRv6 domain, the ingress node (headend) adds SRv6 Encapsulation to packet. In SRv6 TE (traffic engineer) mode, the headend node encapsulates an IPv6 header and an SRH header at the same time. A group of SRv6 segments is encapsulated in the
SRH header to indicate the forwarding path. NRP-ID can be carried in segment to identify the NRP to which the packet belongs.

This document proposes to use the ARG part of the segment to carry the NRP-ID.

```
+-----------------+-----------------+-----------------+
|     Locator     | Function        | NRP-ID          |
+-----------------+-----------------+-----------------+
|<--------------- LOC ------------><- FUNCT -><--------------- ARG ------------->|
```

Figure 2: Encoding NRP-ID in segment

3. Deployment consideration of NRP-ID In segment

In the SRv6 TE mode, multiple segments are encoded in the SRH. The last segment in the SRH is usually the service or End SID of the tailend node and does not need to carry the NRP-ID.

Other segments in SRH are usually End or End.X segments, which are used to guide intermediate endpoint nodes to forward packets. These segments do not need argument and can carry NRP-ID.

Different segments can carry the same or different NRP-ID, which is arranged by the controller or operator by CLI according to the actual requirement.

Segment[0]:
```
+-----------------+-----------------+-----------------+
|     Locator5    | Function        | Argument        |
+-----------------+-----------------+-----------------+
```

Segment[1]:
```
+-----------------+-----------------+-----------------+
|     Locator4    | Function        | NRP-ID2         |
+-----------------+-----------------+-----------------+
```

Segment[2]:
```
+-----------------+-----------------+-----------------+
|     Locator3    | Function        | NRP-ID2         |
+-----------------+-----------------+-----------------+
```

Segment[3]:
```
+-----------------+-----------------+-----------------+
|     Locator2    | Function        | NRP-ID1         |
+-----------------+-----------------+-----------------+
```

Segment[4]:
```
+-----------------+-----------------+-----------------+
|     Locator1    | Function        | NRP-ID1         |
+-----------------+-----------------+-----------------+
```

Figure 3: Encoding NRP-ID in segment list
4. NRP-ID position information advertisement

If the network slicing service needs to be supported, when creating a locator, the SRv6 node needs to determine the encoding position of NRP-ID in the segment according to its own role.

The locator and encoding positions of NRP-ID need to be advertised to the controller and other network nodes. With this information, the controller arranges the SRv6 policy, and other network nodes need to extract the NRP-ID from address during forwarding.

4.1. Static configuration mode

In the static configuration mode, configure the locator encoding information on the controller and network nodes respectively. For the convenience of description, the locator carrying NRP-ID is named slice prefix in this document.

All nodes that support network slicing need to be configured, including SRv6 nodes and IPv6 only nodes.

The aggregation attribute of locator can be used. The following figure is an example. The P nodes only provide End/End.x type segments, and the positions used to encode NRP-ID are usually the same. Therefore, common prefix can be configured to indicate the position of NRP-ID.

If the encoding position of a P node is different from that of most nodes, the slice prefix corresponding to the locator of the P node can be configured separately to specify its encoding position.

These slice prefixes will create a local slice prefix table (LSPT) on the forwarding plane. When forwarding packets, the network node uses the destination address to lookup the LSPT according to the longest matching principle, and then extracts the NRP-ID from the destination address according to the information of the hit table entry.

Referring to the topology and locators of each node in the Figure 4, the following slice prefix can be configured.

The encoding positions of P1 and P4 nodes are the same, and a slice prefix corresponding to common prefix can be configured to identify the coding position as the low 16 bits. The encoding position of P3 is 96bit to 112bit of segment, so a slice prefix corresponding to its locator is configured separately to explain its coding position.
Slice-Prefix1: 2001:1:1::/64 (common prefix)

NRP-ID Position: [112..127]

Slice-Prefix2: 2001:1:1:0:130::/80 (locator for P3)

NRP-ID Position: [96..112] in segment

IPv6-only

+-----+   +----+   +----+   +----+   +-----+
--- PE1 ++++ P1 ++++ P2 ++++ P3 ++++ P4 ++++ PE2 ---
+-----+   +----+   +----+   +----+   +-----+
SRv6-node      SRv6-node  SRv6-node

common Prefix: 2001:1:1::/64
locator
P1: 2001:1:1:0:110::/80 for End/End.x
P3: 2001:1:1:0:130::/80 for End/End.x
P4: 2001:1:1:0:140::/80 for End/End.x

Figure 4: example topology for slice prefixes

4.2. Dynamic advertising mode

To simplify the configuration, slice prefix can be advertised to network nodes in the domain through IGP and to controllers through BGP-LS. This reduces the configuration of controllers and SRv6 nodes.

However, nodes that only support IPv6 need to be configured as described in the previous section.

Relevant protocol extensions will be provided in subsequent versions.

5. Behavior of headend

If the network slice function is enabled, the SRv6 headend node determines the network slice to which the customer traffic belongs according to the relevant policies.

The headend node steers the customer traffic to the SRv6 policy and encapsulates the IPv6 header and SRH header for the customer traffic. The headend node encapsulates the segment list of the SRv6 policy in the SRH header.
At the same time, set NRP-ID into segment. These NRP-IDs can be the same or different values according to actual requirement.

Generally, the nodes along the route of the message use the same NRP-ID to identify the NRP associated with the network slice. Therefore, when the headend node encapsulates the segment list in the SRH, it writes the same NRP-ID into segments except the last segment.

In some special cases, such as cross domain scenarios, different NRP-IDs may be used on the forwarding path. In this case, the controller may need to write the NRP-ID into each segment of the segment list in advance, and then issue the SRv6 policy to the headend node. The headend node only needs to use SRv6 policy to encapsulate customer traffic.

```
+-----------------------------------------------------------+
| IPv6 Header                                               |
| . Source IP Address = IPv6 Address of Ingress             |
| . Destination IP Address = SegmentList[SL]                |
| . Next-Header = SRH (43)                                  |
| +-----------------------------------------------------------+
| SRH as specified in RFC 8754                             |
| . <Segment0>                                             |
| . <Segment1|NRP-ID1>                                           |
| . <Segment2 NRP-ID2>                                    |
| . <SegmentN|NRP-IDN>                                         |
| +-----------------------------------------------------------+
| Payload                                                   |
| +-----------------------------------------------------------+
```

Figure 5: Format of SRv6 TE with slice ID

6. Behavior of endpoint

When a SRv6 node receives a packet, the destination address of the packet is the segment instantiated locally by the SRv6 node. At this time, the SRv6 node processes the packet as endpoint node. The endpoint node extracts the NRP-ID from the segment and forwards the packet with the NRP identified by the NRP-ID.

When N receives a packet whose IPv6 DA is S and S is a local End SID, the pseudo code processed is modified as follows based on RFC[8986]:

```
When \( N \) receives a packet whose IPv6 DA is \( S \) and \( S \) is a local End.X SID, the pseudo code processed is modified as follows based on RFC[8986]:

S01 - S11.
S12. Decrement IPv6 Hop Limit by 1
S13. Decrement Segments Left by 1
S14. Update IPv6 DA with Segment List[Segments Left]
Insert Extract NRP-ID from destination address.

Modify:
S15. Uses the NRP-ID to select a NRP and apply NRP policies to forward packet
S16. }

7. Behavior of transit node

For the transit node, the destination address of the packet is not a local segment, and only IPv6 forwarding is performed for the packet.

The transit node may be a node that supports SRv6 or a node that only supports IPv6.

When processing SRv6 packets, the transit node can use the destination address to lookup the local slice prefix table according to the longest matching principle. If a prefix is hit, the NRP-ID could be extracted according to the configuration information.

Therefore, the processing pseudo code can be modified as follows:
S1. If ((Slice forwarding is enabled) && (Destination address hits network slice prefix)) {
  S2. Extracts NRP-ID from destination address by using the position information of hit prefix
  S2. Uses the NRP-ID to apply NRP policies to forward packet
  S3. }
S4. Else {
  S5. Forwards the packet without applying any NRP policies
  S6. }

8. Example

As shown in the following figure, the IP backbone network deploys the network slicing service. The network operator has created two NRPs, NRP1 and NRP2. NRP1 guarantees 100Mbps bandwidth and NRP2 guarantees 200Mbps bandwidth. Set the IDs NRP-ID1 and NRP-ID2 for the two NRPs respectively.

The IP backbone network provides customers with two network slices. Network slice1 is mapped to NRP1 and network slice2 is mapped to NRP2. SRv6 is used to carry traffic and network slicing services.

Along with the forwarding path <PE1-P1-P2-PE2>, dedicated queues with guaranteed bandwidth for NRP1 and NRP2 are configured at corresponding interfaces of each router. Taking the interface P1-P2 of router P1 as an example, Queue 1 is configured with NRP-ID1 and guaranteed bandwidth of 100Mbps, and Queue 2 is configured with NRP-ID2 and 200Mbps. When P1 transmits a packet through interface P1-P2, the NRP-ID carried in the destination address is checked.

If NRP-ID1 is encapsulated in the destination address, P1 uses queue 1 to transmit the packet. If NRP-ID2 is encapsulated in the destination address, P1 uses queue 2 to transmit the packet.
9. IANA Considerations

This document has no IANA actions.

10. Security Considerations

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

This document does not introduce any new security consideration.
11. References

11.1. Normative References


Authors’ Addresses

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Changwang Lin
New H3C Technologies
China
Email: linchangwang.04414@h3c.com

Hao Li
New H3C Technologies
China
Email: lihao@h3c.com

Liyan Gong
China Mobile
China
Email: gongliyan@chinamobile.com
Generalized Arguments of SRv6 Segment  
draft-lm-spring-srv6-generalized-arguments-00

Abstract

This document analyzes the challenges of Arguments of SRv6 SID, and the chance of using Arguments of SRv6 SID to reduce the length of the IPv6 extension header. According to these analysis, this document specifies a kind of generalized and structured Arguments for SRv6 SID, which can carry multiple Arguments parts for a SRv6 SID.

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 12 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction .................................................. 2
2. Requirements Language ........................................ 2
3. Terminologies .................................................. 2
4. Problem Statement and Requirements .......................... 2
5. Generalized Arguments ......................................... 4
   5.1. Method A .................................................. 4
   5.2. Method B .................................................. 4
   5.3. Consideration of SRv6 C-SID Compression ................. 5
6. Flavor for Generalized Arguments ............................. 6
7. IANA Considerations ........................................... 6
8. Security Considerations ....................................... 6
9. References ..................................................... 6
   9.1. Normative References .................................... 6
   9.2. Informative References ................................... 7
Authors’ Addresses .................................................. 8

1. Introduction

This document analyzes the challenges of the Arguments of SRv6 SID, and the chance of using Arguments of SRv6 SID to reduce the length of the IPv6 extension header. According to these analysis, this document specifies a kind of generalized and structured arguments for SRv6 SID, which can carry multiple Arguments parts for a SRv6 SID.

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 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminologies

SRv6: Segment Routing over IPv6

4. Problem Statement and Requirements

With the development of SRv6, several kinds of SRv6 Arguments for the SRv6 End SID and End.X SID emerge [I-D.ietf-spring-srv6-srh-compression], including:

1. SRv6 C-SID compression (NEXT Flavor): using Arguments to carry multiple C-SIDs.
2. SRv6 C-SID compression (REPLACE Flavor): using Arguments to carry the CL field.

3. SRv6 C-SID compression (NEXT & REPLACE Flavor): using Arguments to carry multiple C-SIDs and the CL field.

In addition, some new features are created, including network slicing[I-D.ietf-6man-enhanced-vpn-vtn-id], IOAM[RFC9197], Alternate Marking[I-D.ietf-6man-ipv6-alt-mark][I-D.fz-spring-srv6-alt-mark], APN6[I-D.li-apn-ipv6-encap][I-D.li-apn-header], DetNet[I-D.pthubert-detnet-ipv6-hbh], etc.

The instructions of these new features can be processed at:

1. All nodes along a SR path: the instructions can be carried in the IPv6 Hop-by-Hop Options header (HBH).

2. Endpoints of an SR path: the ones can be carried in the IPv6 Destination Options Header (DOH) or the SRH TLV.

In the second scenario, especially the second one, the usages of the options or TLVs will cause the following two issues:

1. Lengthening the packet header, and reducing the transmission efficiency.


Besides these issues, in the SRv6 C-SID compression (NEXT Flavor) solution, if all the C-SIDs of the SID list which should have been encapsulated in the SRH can be put in the IPv6 destination address of the packet, because there is no SRH or DOH before SRH any more after the compression there will be no space for the instructions which should have been encapsulated in the IPv6 SRH or Destination Options Header before SRH.

In order to address these challenges, a feasible solution is to use the Arguments of the SRv6 SID to carry those instructions. Using SRv6 Arguments to do that will bring following benefits:

1. Reducing the needed space of IPv6 extension header or SRH TLV, so as to reduce the transmission overhead.
2. SRv6 SID can reside in the IPv6 destination address field, so the SRv6 Arguments can be read and processed as a part of IPv6 address, from which the forwarding performance will benefit, because it avoids to process the extension header or SRv6 TLV behind the IPv6 header.

3. Unify and simplify the processing: the instructions of both the SRv6 and the new features are all put in the Arguments part of SRv6 SID or IPv6 address.

5. Generalized Arguments

In order to carry the instructions of multiple features in the SRv6 Arguments, this section defines two methods to make the SRv6 Arguments generalized and structured to allocate spaces for the instructions.

5.1. Method A

Network devices are configured a template for the purpose of parsing the SRv6 Arguments, and the devices read and process the content of the Arguments according to the template.

The template defines what features are carried, and which bits they are used.

For example, if the length of the Arguments is z bits and the number x, y, and z have the relationship 0<x<y<z, then the template can define that:

* The [0, x) bits carry the instructions of feature A;

* The [x, y) bits carry the instructions of feature B;

* The [y, z) bits carry the instructions of feature C.

5.2. Method B

Define a bitmap in the Arguments, and each bit in the bitmap indicates whether the instructions of a specific feature exist. The correspondence of the bit and the feature, the length of the space of Arguments to carry the instructions for the feature, and the instructions needed to be carried for a specific feature can be defined further in a standardization way.

The bitmap can be encoded from the most significant bit (MSB) or the least significant bit (LSB).
When the bit is set (1), it indicates the instructions of the feature exist. If the bit is reset (0), there can be two options:

Option 1: it indicates the instructions of the feature don’t exist.

Option 2: it indicates the instructions of the feature exist but is invalid.

5.3. Consideration of SRv6 C-SID Compression

Since it is required to shift the C-SID in the SRv6 SID while applying the NEXT or NEXT & REPLACE behavior for SRv6 C-SID compression, when method A or B is adopted, when C-SIDs are encoded in the generalized Arguments of the SRv6 SID which is used as the IPv6 destination address, these C-SIDs MUST be placed from the most significant bit (MSB), that is, these C-SIDs MUST immediately following the LOC:FUNCT part of the SRv6 SID.

The remaining part of the generalized Arguments following the C-SIDs SHOULD NOT be shifted when C-SIDs part is shifted. This means the position of the remaining part after the C-SIDs in the generalized arguments SHOULD be fixed.
6. Flavor for Generalized Arguments

This section defines a new flavor to support processing the
Generalized Arguments, named as Structured Arguments flavor.

The pseudocode of the Structured Arguments flavor is as follows:

Method A:
S01. If (some NEXT-C-SIDs are encoded in the Generalized Arguments) {
S02.       Left shift the C-SIDs by the length of one C-SID
S03. }
S04. Load the relative template
S05. Parse the Generalized Arguments as per section 4.1
S06. For each parsed feature {
S07.       Perform actions according to the parsed instructions
             as per the specifications of that feature
S08. }

Method B:
S01. If (some NEXT-C-SIDs are encoded in the Generalized Arguments) {
S02.       Left shift the C-SIDs by the length of one C-SID
S03. }
S04. For each bit in the bitmap {
S05.       If (the bit == 1) {
S06.           Parse the instructions of the feature from the Generalized
                     Arguments as per section 4.2
S07.           Perform actions according to the parsed instructions
                     as per the specifications of that feature
S08.       }
S09. }

7. IANA Considerations

TBD

8. Security Considerations

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,

9.2. Informative References


[I-D.li-apn-header]

[I-D.li-apn-ipv6-encap]

[I-D.pthubert-detnet-ipv6-hbh]


Authors’ Addresses
Zhenbin Li
Huawei Technologies
Beijing
100095
China
Email: lizhenbin@huawei.com

Jianwei Mao
Huawei Technologies
Beijing
100095
China
Email: MaoJianwei@huawei.com

Cheng Li
Huawei Technologies
Beijing
100095
China
Email: c.l@huawei.com
Abstract

This document defines the Path MTU (PMTU) for SR Policy (called SR-PMTU) and it applies to both SRv6 and SR-MPLS. The framework of SR-PMTU for SR Policy is specified, including link MTU collection, SR-PMTU Computation, SR-PMTU Enforcement, and Handling behaviors on the headend.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components
1. Introduction

Segment Routing (SR) [RFC8402] allows a node to steer a packet flow along any path. The headend is a node where the instructions for source routing (i.e., segments) are written into the packet and hence becomes the starting node for a specific segment routing path. Intermediate per-path states are eliminated thanks to source routing.

A Segment Routing Policy (SR Policy) [I-D.ietf-spring-segment-routing-policy] is an ordered list of segments (i.e., instructions) that represent a source-routed policy. The headend node is said to steer a flow into a SR Policy. The
packets steered into an SR Policy have an ordered list of segments associated with that SR Policy written into them. [RFC8660] describes the representation and processing of this ordered list of segments as an MPLS label stack for SR-MPLS, while [RFC8754] and [RFC8986] describe the same for Segment Routing over IPv6 (SRv6) with the use of the Segment Routing Header (SRH).


This document extends the SR Policy to also include the Path MTU information to SR Policy and applies to both SRv6 and SR-MPLS. The SRv6 specific handling are specified in a separate section.

1.1. Motivation

The motivation for handling SR-PMTU for the SR paths includes (but not limited to):

* Being able to avoid fragmentation by being aware of the SR-PMTU associated with the SR paths and policies at the headend.

* Being able to generate ICMP messages at the headend.

* When fragmentation is unavoidable, ability to do it correctly at the headend.

* Ability to use SR-PMTU as path computation constraint and optimization criteria at the headend or the controller/PCE.

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 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology
Link MTU: As per [RFC4821], the Maximum Transmission Unit, i.e., maximum IP packet size in bytes, that can be conveyed in one piece over a link. This includes the IP header, but excludes link layer headers and other framing that is not part of IP or the IP payload. In case of MPLS, it also includes the label stack and in case of IPv6, it includes IPv6 extension headers (including SRH).

Path MTU, or PMTU: The minimum link MTU of all the links in a path between a source node and a destination node. In the scope of this document, this is also called SR-PMTU for the SR paths and policies. Note that the link MTU takes the SR overhead (label stack or SRH) into consideration.

4. SR-PMTU Definition for SR Policy

Segment Routing policy architecture is specified in [I-D.ietf-spring-segment-routing-policy]. An SR Policy is associated with one or more candidate paths. A candidate path is selected when it is valid and it is determined to be the best path of the SR Policy. The selected path is referred to as the "active path" of the SR policy. A candidate path is either dynamic, explicit, or composite. The related concepts with the SR-PMTU definition in this document are listed as follows.

An explicit/dynamic candidate path is expressed as a Segment-List or a set of Segment-Lists directly or by computation. If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with weight for weighted load balancing. The default weight is 1.

A composite candidate path acts as a container for grouping SR Policies. The composite candidate path construct enables the combination of SR Policies, each with explicit candidate paths and/or dynamic candidate paths with potentially different optimization objectives and constraints, for load-balanced steering of packet flows over its constituent SR Policies [I-D.ietf-spring-segment-routing-policy].

4.1. SR-PMTU of a Segment List

A Segment-List represents a specific source-routed path to send traffic from the headend to the endpoint of the corresponding SR policy [I-D.ietf-spring-segment-routing-policy]. The SR-PMTU of a segment list is defined as the minimum link MTU of all the links in a path between a source node and a destination node. Refer Section 5.2 for specific handling for Node, Adjacency and Binding SID (as well as their combinations).
4.2. SR-PMTU of a Candidate Path

In the case of an explicit/dynamic candidate path, if it is expressed as a single Segment-List, then the SR-PMTU of the candidate path is the same as that of the SR-PMTU of the segment list as described in the above section.

In the case of an explicit/dynamic candidate path, if it is expressed as a set of Segment-Lists (for load-balancing), then the SR-PMTU of the candidate path is defined as the minimum SR-PMTU of all the Segment-Lists in the set.

In the case of a composite candidate path, then the SR-PMTU of the composite candidate path is defined as the minimum SR-PMTU of all the constituent SR Policies of this composite candidate path. The SR-PMTU of each SR Policy is defined in the following subsection.

4.3. SR-PMTU of an SR Policy

According to [I-D.ietf-spring-segment-routing-policy], an SR Policy is associated with one or more candidate paths. A candidate path is selected when it is valid and it is determined to be the best path of the SR Policy. The selected path is referred to as the "active path" of the SR policy. Then the SR-PMTU for an SR Policy is defined as the SR-PMTU of the selected/active candidate path of this SR policy.

In the case of an explicit/dynamic candidate path, the SR-PMTU definition can be referred to in the above subsection.

In the case of a composite candidate path, the SR-PMTU is defined as the minimum SR-PMTU of all the constituent SR policies. Since the constituent SR Policies of a composite candidate path do not use composite candidate paths, only explicit/dynamic candidate paths will be used, then the SR-PMTU definition of explicit/dynamic candidate path can be referred to in the above subsection.

5. The Framework of SR-PMTU for SR Policy

The framework of SR-PMTU for SR Policy includes link MTU collection, SR-PMTU computation, SR-PMTU enforcement, and handling behaviors on the headend.
5.1. Link MTU Collection

SR-PMTU is defined as the minimum link MTU of all the links in a path between a source node and a destination node. The link MTU needs to be first collected. The link MTU can be collected through various protocols such as IGP [I-D.hu-lsr-igp-path-mtu] and BGP-LS [I-D.ietf-idr-bgp-ls-link-mtu], etc.

5.2. SR-PMTU Computation

The collected link MTU of all the related links are sent to the network controller where the SR-PMTU is computed. Depend upon the path type, the computation methods are different, which are described in the following subsections.

5.2.1. Loose TE Path

In a loose TE path [RFC7855], only Node SIDs are used along the path. Between two adjacent Node SIDs, generally there are equal-cost multipaths (ECMP). The SR-PMTU of the loose TE path is computed by finding out the minimum SR-PMTU of all the ECMPs between two adjacent Node SIDs along the loose TE path.

Note that an implementation could maintain the SR-PMTU value associated with a Node SIDs at the time of best path computation. The details of which are out of scope of this document.
5.2.2. Strict TE Path

In a strict TE path [RFC7855], only Adj SIDs are used along the path. Since the link MTU of all the links being indicated by the Adj SIDs of the strict TE path are sent to the network controller, the SR-PMTU of the strict SR-TE path is computed by finding out the minimum link MTU of all the links in the strict SR-TE path between its source node and destination node.

5.2.3. Mixed Path

In a mixed path, both Node SIDs and Adj SIDs are used along the path. The PMTU of the mixed TE path is computed by finding out the minimum value of all the ECMPs between two adjacent Node SIDs and the link MTU of the all links indicated by the Adj SIDs.

5.2.4. Binding Path

The Binding SID (BSID) [RFC8402] is bound to an SR Policy, instantiation of which may involve a list of SIDs. The SR-PMTU of the binding path is the same as that of an SR Policy as specified in the above section modulo that it also includes the encapsulation overhead associated with it (i.e. in case of SR-MPLS, the additional label stack pushed in case of SR-MPLS and the outer IPv6 header with its own SRH in case of SRv6). This is done to make sure the headend of the SR path that includes a BSID is able to compute the SR-PMTU correctly by taking the correct SR-PMTU of the binding path into consideration along with other SIDs in the SR path.

5.2.5. TI-LFA

Topology Independent Loop-free Alternate Fast Re-route (TI-LFA) [I-D.ietf-rtgwg-segment-routing-ti-lfa], aimed at providing protection of node and adjacency segments within the SR framework. The repair path is to pre-compute SPT_new(R,X) for each destination, that is, the Shortest Path Tree rooted at node R in the state of the network after the resource X has failed. An implementation is free to use any local optimization to provide smaller SID lists by combining Node SIDs and Adjacency SIDs. In addition, the usage of Node-SIDs allow to maximize ECMPs over the repair path. Note that while the PMTU of repair path might be different from the original path, which could lead to fragmentation while the repair path is in use. When the controller has computed the new path, its new PMTU would be updated to the headend.
Note that it is possible for the headend implementation to take an FRR overhead into consideration when determining if fragmentation would be needed for the SR Path with TI-LFA enabled. If this is used, an implementation SHOULD allow the value to be configured by an operator.

5.2.6. Others

All other types of path can be considered here in future updates.

5.3. SR-PMTU Enforcement

Currently there is still no SR-PMTU in the SR Policy encoding structure [I-D.ietf-spring-segment-routing-policy]. As specified in [I-D.ietf-idr-sr-policy-path-mtu], the SR-PMTU is encoded in the SR policy structure as shown in Figure 2. After the SR-PMTU computation, the SR-PMTU is enforced along with the SR Policy to the headend of the corresponding path.

SR Policy SAFI NLRI: <Distinguisher, Policy-Color, Endpoint>
Attributes:
  Tunnel Encaps Attribute (23)
  Tunnel Type: SR Policy
    Binding SID
    Preference
    Priority
    Policy Name
    Explicit NULL Label Policy (ENLP)
    Segment List
    Weight
----> Path MTU (SR-PMTU)
      Segment
      Segment
      ...
...  

Figure 2. The SR Policy encoding structure with SR-PMTU

When there are multiple paths that can be selected, the one with the highest SR-PMTU will be enforced in order to avoid the fragmentation on the headend.
5.4. Handling behaviors on the headend

After the SR-PMTU is computed and enforced on the headend, the headend is going to perform the handling behaviors such as encapsulation, fragmentation, etc. Note that this behavior is similar to the existing behavior of MPLS and IPv6 dataplane.

5.4.1. SR-PMTU Constraints and Optimization

Generally, considering its services being carried, the operators set a SR-PMTU limit aiming to a proper path selection that fulfill packet size requirements hence avoiding fragmentation. Furthermore, the encapsulation on the headend will introduce the overhead on top of the packet to be encapsulated. Generally, the encapsulation overhead has to be estimated according to the possible path hops and sometimes the repair paths. Therefore, the SR-PMTU constraint is set considering both the carried services and the encapsulation overhead.

When SR-PMTU based path optimization is done, PCE will select the path with the highest SR-PMTU among all the possible paths.

Even SR-PMTU is not considered by the PCE at the time of path computation

Once the SR-PMTU constraint is set on the headend, it is supposed to be the lowest bound of the SR-PMTUs of all the paths being computed locally or enforced by the controller in order to avoid fragmentation.

5.4.2. Fragmentation processing

If the SR-PMTU of all the paths being computed locally or enforced by the controller are smaller than the SR-PMTU constraint set on the headend, the fragmentation will have to be handled. If the fragmentation is not possible, the headend could generate the ICMP messages to notify the traffic source.

Over this selected path, on the headend the packets are fragmented in order to guarantee the size of the encapsulated packets smaller than the PMTU of the selected path.

6. SRv6 Specific Handling

In case of SRv6, the SRH is included in the calculation of the Link MTU and thus in the SR-PMTU. Note that the PMTU considerations for IPv6 [RFC8201] apply for the SRv6. [RFC8754] also specify the MTU considerations related to encapsulation with an outer IPv6 header with SRH.
7. Security Considerations

TBD

8. IANA Considerations

This document does not include an IANA request.

9. Acknowledgement

Thanks to xx for useful discussions and comments.

10. References

10.1. Normative References


10.2. Informative References


Peng, et al. Expires 6 January 2023
[I-D.ietf-idr-sr-policy-path-mtu]

Authors’ Addresses

Shuping Peng
Huawei Technologies
Huawei Campus, No. 156 Beijing Rd.
Beijing
100095
China
Email: pengshuping@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore 560066
India
Email: dhruv.ietf@gmail.com

Ketan Talaulikar
Arrcus Inc
India
Email: ketant.ietf@gmail.com

Gyan Mishra
Verizon Inc.
United States of America
Email: gyan.s.mishra@verizon.com
This document describes how Segment Routing (SR) policies can be used to satisfy the requirements for strict bandwidth guarantees, end-to-end recovery and persistent paths within a segment routing network. SR policies satisfying these requirements are called "circuit-style" SR policies (CS-SR policies).

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
1. Introduction

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

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

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

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

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

2. Terminology

* BSID : Binding Segment Identifier
* CS-SR : Circuit-Style Segment Routing
* ID : Identifier
* LSP : Label Switched Path
* LSPA : LSP attributes
The reference model for CS-SR policies is following the Segment Routing Architecture [RFC8402] and SR Policy Architecture [I-D.ietf-spring-segment-routing-policy] and is depicted in Figure 1.

```
+--------------+                     +--------------+
|               |         <<<<<<<< CS-SR Policy >>>>>>>>>         |               |
|               |                     |               |
|   A           | SR-policy from A to Z | Z             |
|               |<-------------------------------|------------------>
|               | SR-policy from Z to A +-------+
```

Figure 1: Circuit-style SR Policy Reference Model

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

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

* The bandwidth available for CS-SR policies specified

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

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


4. CS-SR Policy Characteristics

A CS-SR policy has the following characteristics:

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

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

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

* Not automatically recomputed or reoptimized: the SID list of a candidate path must not change automatically to a SID list representing a different path (for example upon topology change)
* Multiple candidate paths in case of protection/restoration:
  
  - Following the SR policy architecture, the highest preference valid path is carrying traffic
  
  - Depending on the protection/restoration scheme (Section 6), lower priority candidate paths
    o may be pre-computed
    o may be pre-programmed
    o may have to be disjoint
  
  * Connectivity verification and performance measurement is activated on each candidate path (Section 7)

5. CS-SR Policy Creation

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

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

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

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

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

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

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

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

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

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

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

5.1. Maximum Segment Depth

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

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

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

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

6. Recovery Schemes

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

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

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

6.1. Unprotected

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

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

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

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

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

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

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

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

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

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

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

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

6.3.1. 1+R Restoration

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

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

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

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

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

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

6.3.2. 1:1+R Restoration

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

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

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

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

7. Operations, Administration, and Maintenance (OAM)

7.1. Connectivity Verification

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

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

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

7.2. Performance Measurement

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

7.3. Candidate Path Validity Verification

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

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

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

8. External Commands

8.1. Candidate Path Switchover

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

8.2. Candidate Path Recomputation

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

9. Security Considerations

TO BE ADDED
10. IANA Considerations

This document has no IANA actions.

11. Acknowledgements

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

12. Contributors

Contributors’ Addresses

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

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

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

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

13. References

13.1. Normative References


13.2. Informative References

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

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

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

[I-D.ietf-pce-multipath]

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

[I-D.ietf-pce-segment-routing-policy-cp]
[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong,

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

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

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


Authors’ Addresses

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

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

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

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

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

Reza Rokui
Ciena
Email: rrokui@ciena.com

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

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
Circuit Style Segment Routing Policies
draft-schmutzer-spring-cs-sr-policy-00

Abstract

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

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the
document authors. All rights reserved.

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

Table of Contents

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

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

* Persistent end-to-end traffic engineered paths that provide predictable and identical latency in both directions

* Strict bandwidth commitment per path to ensure no impact on the Service Level Agreement (SLA) due to changing network load from other services

* End-to-end protection (<50msec protection switching) and restoration mechanisms

* Monitoring and maintenance of path integrity

* Data plane remaining up while control plane is down

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

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

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

2. Terminology

* BSID : Binding Segment Identifier

* CS-SR : Circuit-Style Segment Routing

* ID : Identifier

* LSP : Label Switched Path

* LSPA : LSP attributes
3. Reference Model

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

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

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

* The bandwidth available for CS-SR policies specified

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

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


4. CS-SR Policy Characteristics

A CS-SR policy has the following characteristics:

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

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

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

* Not automatically recomputed or reoptimized: the SID list of a candidate path must not change automatically to a SID list representing a different path (for example upon topology change)
* Multiple candidate paths in case of protection/restoration:
  - Following the SR policy architecture, the highest preference valid path is carrying traffic
  - Depending on the protection/restoration scheme (Section 6), lower priority candidate paths
    - may be pre-computed
    - may be pre-programmed
    - may have to be disjoint
* Connectivity verification and performance measurement is activated on each candidate path (Section 7)

5. CS-SR Policy Creation

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

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

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

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

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

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

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

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

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

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

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

5.1. Maximum Segment Depth

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

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

* The BSID is unprotected, hence only has one candidate path

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

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

6. Recovery Schemes

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

* Protection : another candidate path is computed and fully established in the data plane and ready to carry traffic

* Restoration : a candidate path may be computed and may be partially established but is not ready to carry traffic

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

6.1. Unprotected

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

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

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

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

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

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

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

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

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

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

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

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

6.3.1. 1+R Restoration

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

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

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

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

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

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

6.3.2. 1:1+R Restoration

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

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

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

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

7. Operations, Administration, and Maintenance (OAM)

7.1. Connectivity Verification

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

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

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

7.2. Performance Measurement

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

7.3. Candidate Path Validity Verification

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

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

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

8. External Commands

8.1. Candidate Path Switchover

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

8.2. Candidate Path Recomputation

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

9. Security Considerations

TO BE ADDED
10. IANA Considerations

This document has no IANA actions.

11. Acknowledgements

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

12. Contributors

Contributors’ Addresses

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

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

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

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

13. References

13.1. Normative References


13.2. Informative References

[I-D.ietf-idr-bgpls-srv6-ext]
[I-D.ietf-lsr-isis-srv6-extensions]

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

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

[I-D.ietf-pce-multipath]

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

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

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

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

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


Authors’ Addresses

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

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

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

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

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

Reza Rokui
Ciena
Email: rrokui@ciena.com

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

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
Shuping Peng
Huawei Technologies
Email: pengshuping@huawei.com

Tarek Saad
Juniper Networks
Email: tsaad@juniper.net

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