

DAY ONE: INTRODUCTION TO SRV6

This book takes you through the basics of SRv6 and a detailed description of configuring SRv6 on Junos®.



By Shraddha Hegde, Abhishek Chakraborty, Salih KA, Sunil Malali

DAY ONE: INTRODUCTION TO SRV6

SRv6 is a segment routing paradigm applied to an IPv6 underlay with a new IPv6 extension header called Segment Routing Header (SRH). While SR-MPLS for IPv4 and IPv6 underlay does not require any change in the forwarding plane, SRv6 requires forwarding plane support to process the newly defined extension header. SRv6 uses IPv6 addresses to represent transport endpoints as well as service instances such as Virtual Private Networks (VPN). This brings in an unique advantage because these service instance identifiers can be advertised as prefixes. It also has the advantage that these service instance identifiers can be easily summarized and improve scalability. In SRv6, the transit routers, which are not anchor points, use simple IPv6 forwarding and this ability makes SRv6 much simpler to migrate the networks.

SRv6 also has network programming ability. The large IPv6 address space provides flexibility to encode instructions in the destination address. This capability is attractive to cloud networking use cases.

This Day One book will take you through the basics of SRv6 and a detailed description of configuring SRv6 on Junos. The authors cover all the supported features in SRv6 until the 22.3 release. Previous knowledge of basic Segment Routing features such as TI-LFA, Flexible-Algorithms, and SR-TE will greatly help as you cruise through this book.

"This book takes you step-by-step all the way from the basic principles of SRv6 to advanced topics such as micro-loop avoidance and SRv6-MPLS interworking. It's accompanied by detailed configuration examples and router outputs. It's an invaluable reference to network architects, engineers and net-ops alike."

Julian Lucek, Sr Distinguished Systems Engineer, Juniper Networks

IT'S DAY ONE AND YOU HAVE A JOB TO DO:

- Understand the basics of SRv6
- Configure basic SRv6 transport and services
- Learn advanced features like TI-LFA, micro-loop avoidance, and SR-TE
- Deploy SRv6 for various use cases
- Troubleshoot SRv6

Day One: Introduction to SRv6

by Shraddha Hegde, Abhishek Chakraborty,
Salih K A, and Sunil Malali

<i>Chapter 1: Preparing the Network for SRv6 Migration</i>	7
<i>Chapter 2: Configuring SRv6 Transport</i>	19
<i>Chapter 3: Configuring SRv6 Services</i>	25
<i>Chapter 4: Enabling TI-LFA Protection</i>	42
<i>Chapter 5: SRv6 to MPLS Interworking</i>	52
<i>Chapter 6:BGP-LS Support with SRv6</i>	72
<i>Chapter 7: Configuring SRv6 Flex Algo</i>	81
<i>Chapter 8: IPv6 Only Networks</i>	98
<i>Chapter 9: Micro-loop Avoidance</i>	101
<i>Chapter 10: Configuring SRv6-TE</i>	118
<i>Chapter 11: EVPN-VPWS</i>	141
<i>Chapter 12: Multi-domain SRv6</i>	161
<i>Chapter 13: cRPD Support</i>	177
<i>Appendices</i>	193

© 2022 by Juniper Networks, Inc. All rights reserved.

Juniper Networks and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo and the Junos logo, are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Published by Juniper Networks Books

Authors: Shraddha Hegde, Abhishek Chakraborty, Salih K A, and Sunil Malali

Technical Reviewers: Julian Lucek, Ron Bonica, Krzysztof Szarkowicz

Editor in Chief: Patrick Ames

Copyeditor: Nancy Koerbel

Version History: v1, October 2022

2 3 4 5 6 7 8 9 10

About the Authors

Shraddha Hegde is a Distinguished Engineer in Juniper Networks' Routing Protocols Group. She (co)-authored five IETF RFC documents and several internet drafts. Shraddha has 20 years of experience in Networking Domain in the area of IGP, FRR, Segment Routing, MPLS, and IP Security. She has vast experience in building and deploying highly scalable data-communications platforms. Prior to joining Juniper, Shraddha served as a Systems Architect for a major telecom equipment vendor. .

Abhishek Chakraborty is a Software Engineer 4 in Juniper Networks' Routing Protocols Group. He co-authored 5 patents in Juniper. Abhishek has an overall of 10 years of experience in the Networking and Embedded Systems, and Network Management Systems. His contributions in networking domain lies in Multicast, Segment-Routing, Segment-Routing Traffic Engineering. Recently his focus has been on SRv6 TE development along with BGP Link State Distribution of SR-TE and SR P2MP. Prior to joining Juniper, Abhishek served a System Engineer for Z Enterprise Servers in IBM.

Salih KA is a Software Engineering Senior Manager in Juniper Networks' Routing Protocols Group. He co-authored 8 patents in Juniper. He also co-authored few drafts in ietf. Salih has an overall of 14 years of experience in the Networking and Embedded Systems, and Network Management Systems. His contributions in networking domain lies in ISIS, OSPF, RIP, BFD, Segment-Routing, Segment-Routing Traffic Engineering, SRv6. Prior to joining Juniper, Salih served as a Software Engineer for Cisco Systems, Yahoo and HP.

Sunil Malali is currently a Staff Engineer with the Routing Protocols group in Juniper Networks, Bangalore, India. Having worked for close to 15 years in MPLS, L2VPN and EVPN technologies and part of Juniper EVPN team for last 10 years. He co-authored 5 patents in Juniper. Recently involved in developing L2 service over classful transport and SRv6. He leads the effort to support EVPN VPWS over SRv6 in Junos. Prior to joining Juniper, Sunil Malali served as Software Engineer for ECI Telecom, Nevis Networks.

Authors' Acknowledgments

The authors would like to thank Patrick Ames for his great edits, and the reviewers Julian Lucek, Ron Bonica, and Krzysztof Szarkowicz for their careful review and suggestions. Rajesh M, Bharath R , Parag Kaneria, Mohana Sundari M, and Amit Kumar Srivastava for their technical inputs, and Ram Santhanakrishnan, Ramesh Kandula, and Syed Hussain for their constant support for the work on the book.

Welcome to Day One

This book is part of the *Day One* library, produced and published by Juniper Networks Books. *Day One* books cover Juniper Networks network administration with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow.

- Download a free PDF edition at <https://www.juniper.net/dayone>.
- Purchase the paper edition at Vervante Corporation (www.vervante.com).

Key Segment Routing Resources

The authors of this book highly recommend the following SR resources. Also examine the [Juniper TechLibrary](#) and its up to date information and specifications.

The following books are available on Segment Routing at <https://www.juniper.net/dayone>: Look for an SR book bundle with all SR book titles.

- *Day One: Configuring Segment Routing With Junos*
- *Day One: Migrating To Segment Routing*
- *Day One: Inside Segment Routing*

What You Need to Know Before Reading This Book

Before reading this book, you need to be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the Day One Fundamentals Series on learning the Junos OS, at <http://www.juniper.net/dayone>. This book makes a few assumptions about you, the reader:

- You are familiar with and versed in using the Junos CLI for router configuration.
- You are familiar with basic ISIS protocol functioning and CLI configurations.
- You are familiar with IPv6 protocol and functioning.
- You have a basic understanding of Segment Routing.
- You can build out the lab topologies used in this book without detailed set up instructions.

What You Will Be Able To Do After Reading This Book

- Understand the basics of SRv6
- Configure basic SRv6 transport and services
- Learn advanced features like TI-LFA, micro-loop avoidance, and SR-TE
- Deploy SRv6 for various use cases
- Troubleshoot SRv6

Preface

Segment Routing (SR) is a source routing paradigm and promises to remove transit state from the network. The network path is encoded in segments and sent in the packet header. Segment Routing can be applied to an MPLS dataplane or an SRv6 dataplane. SR-MPLS is fully compliant to MPLS architecture and does not require any changes to the forwarding plane. SR-MPLS can be applied to an IPv4 or IPv6 underlay or dual stack networks.

SRv6 is a segment routing paradigm applied to an IPv6 underlay with a new IPv6 extension header called Segment Routing Header (SRH). While SR-MPLS for IPv4 and IPv6 underlay does not require any change in the forwarding plane, SRv6 requires forwarding plane support to process the newly defined extension header. SRv6 uses IPv6 addresses to represent transport endpoints as well as service instances such as Virtual Private Networks (VPN). This brings in a unique advantage because these service instance identifiers can be advertised as prefixes. It also has the advantage that these service instance identifiers can be easily summarized and improve scalability. In SRv6, the transit routers which are not anchor points use simple IPv6 forwarding and this ability makes SRv6 much simpler to migrate the networks.

SRv6 also has network programming ability. The large IPv6 address space provides flexibility to encode instructions in the destination address. This capability is attractive to cloud networking use cases.

This Day One book will take you through the basics of SRv6 and a detailed description of configuring SRv6 on Junos. We have tried to cover all the supported features in SRv6 until the 22.3 release. A previous knowledge of basic Segment Routing features such as TI-LFA, Flexible-Algorithms, and SR-TE greatly helps in cruising through this book. The authors would also like to point readers to previous *Day One* books listed a few pages back, which are a great resource on Segment Routing.

Chapter 1

Preparing the Network for SRv6 Migration

1.2 SRv6

1.2.1 Introduction to SRv6

SRv6, Segment Routing IPv6, is a next-gen IP protocol that combines Segment Routing (SR) and IPv6, leveraging existing IPv6 forwarding technology. SRv6 implements network programming through flexible IPv6 extension headers.

Network programming is the capability of a network to encode a network program into individual instructions. In SR-MPLS these instructions are carried in MPLS labels whereas in SRv6, these network instructions are carried natively in the IPv6 extension headers.

These are called the SRv6 segment identifiers (SIDs) and are represented by a 128-bit IPv6 address. The IPv6 packet carrying the network instructions explicitly tells the network about the precise SRv6 nodes available for packet processing.

The ISIS protocol distributes the SIDs for each node and links them through the network. The 128 bit SID can be used to convey a specific network programming instruction.

1.2.2 SRv6 SIDs

A SID represents a specific segment in a segment routing domain. In an IPv6 network, the SID-type used is a 128-bit IPv6 address also referenced as an *SRv6 Segment* or *SRv6 SID*.

An SRv6 SID consists of the following parts:

- *Locator*— Locator is the first part of a SID that consists of the most significant bits representing the address of a particular SRv6 node. The locator is very similar to a network address that provides a route to its parent node. The IS-IS protocol installs the locator route in the inet6.0 routing table. IS-IS routes the segment to its parent node, which subsequently performs a function defined in the second part of the SRv6 SID. You can also specify the algorithm associated with this locator or define a flexible algorithm as per your network requirements.
- *Function*—The second part of the SID defines a function that is performed locally on the node identified by the locator. There are several functions that have already been defined in [RFC 8986](#), *SRv6 Network Programming*. However, Junos® supports the following transport functions that are signaled in IS-IS. IS-IS installs these function SIDs in the inet6.0 routing table.
 - *End*— This is an endpoint function for SRv6 instantiation of a Prefix SID. It does not allow for decapsulation of an outer header for the removal of an SRH. Therefore, an End SID cannot be the last SID of a SID list and cannot be the Destination Address (DA) of a packet without an SRH (unless combined with the PSP, USP or USD flavors).
 - *End.X*— This is an endpoint X function for SRv6 instantiation of an adjacency SID. It is a variant of the endpoint function with Layer 3 cross-connect to an array of Layer 3 adjacencies.
 - *End.DT4*— This is an endpoint DT4 function for SRv6 instantiation of a Global or Layer 3 VPN IPv4 services. It is a variant of the endpoint decapsulation function associated with a Global or Layer 3 VPN instance to transport IPv4 services over SRv6 underlay.
 - *End.DT6*— This is an endpoint DT6 function for SRv6 instantiation of a Global or Layer 3 VPN IPv6 services. It is a variant of the endpoint decapsulation function associated with a Global or Layer 3 VPN instance to transport IPv6 services over SRv6 underlay.
 - *End.DT46*— This is an endpoint DT46 function for SRv6 instantiation of a Global or Layer 3 VPN IPv4 and IPv6 services. It is a variant of the endpoint decapsulation function associated with a Global or Layer 3 VPN instance to transport both IPv4 and IPv6 services over SRv6 underlay. Basically, END.DT46 is shared across IPv4 and IPv6 prefixes.
- You can specify End SID behavior such as Penultimate Segment Pop (PSP), Ultimate Segment Pop (USP) or Ultimate Segment Decapsulation (USD).
- *PSP*— When the last SID is written in the destination address, the End and End.X functions with the PSP flavor pop the top-most SRH. Subsequently stacked SRHs may be present but are not processed as part of the function.

- **USP**—When the next header is SRH and there are no more segments left, the IS-IS protocol pops the top SRH, looks up the updated destination address and forwards the packet based on a match table entry.
- **USD**—When the next header in the packet is 41, or is SRH, and there are no more segments left, the node decapsulates the outer IPv6 header and its extension headers, looks up the exposed inner IP destination address and forwards the packet to the matched table entry.

Figure 1.1 shows the SRv6 SID format.

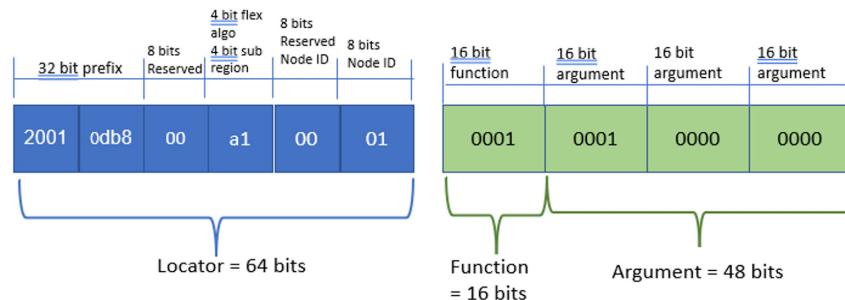


Figure 1.1 Pictorial representation of a SRv6 SID

1.2.3 Segment Routing Extension Header or SR

The header which contains these SIDs is called the *Segment Routing Extension Header* or SRH. SRv6 stacks up these IPv6 addresses instead of MPLS labels in a segment routing extension header as shown in Figure 1.2's simple pictorial representation of SRH.

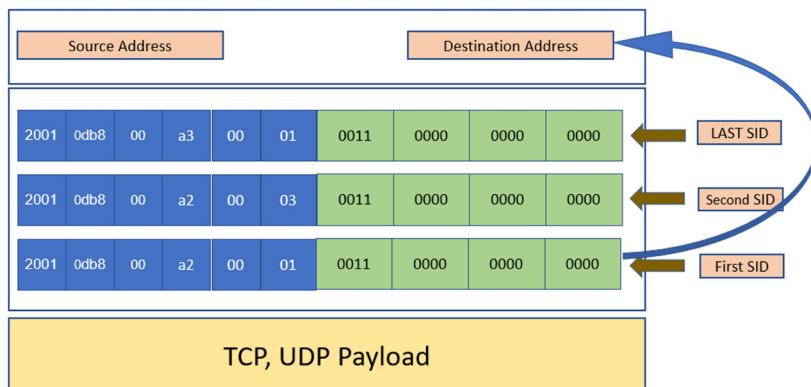


Figure 1.2 A Simple Pictorial Representation of SRH

You can see in the figure that each time a SRv6 Node is visited the SRH is processed based on the SID's Endpoint behavior in that SRv6 Node and the IPv6 destination address of the packet is updated with the next SID in the stack. This processing continues until the remaining segment becomes 0 and the SRH is decapsulated and the payload is exposed. After this point forwarding happens on the Payload destination address.

A more protocol specific representation can be found in Figure 1.3.

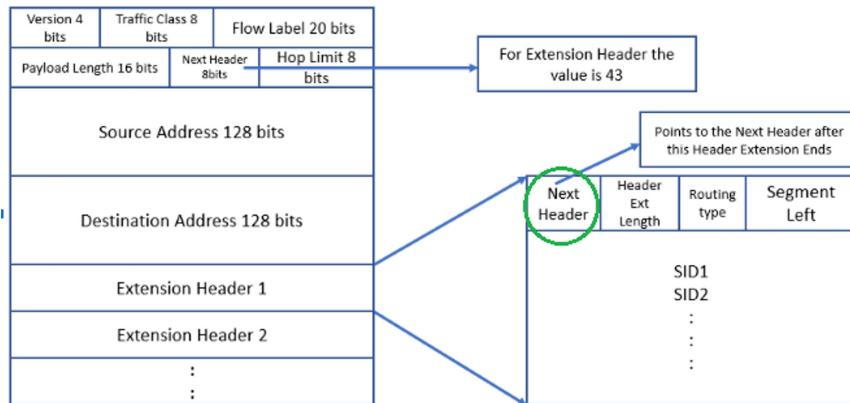


Figure 1.3 A Protocol Specific Representation of the IPv6 Extension Header and SRH

Let's review the figure more closely. The left part of the figure represents an IPv6 header with its header fields. As marked, the Next Header bits signify whether the extension header is present or not. SRH is an extension header in an IPv6 header. To identify the SRH, the Next Header field in the IPv6 header has a value 43.

The SRH also has a Next Header field marked with a green circle. This contains the Next Header information. In the case where you are transporting IPv4 Service payload via the SRv6 Transport, the Next Header field will be marked as 4, and in case of IPv6 payload, this will be marked as 41.

A SRv6 Node, therefore where a SRv6 SID is programmed in the forwarding plane, is visited by a packet whose destination is set to its programmed SRv6 SID, and then that SRv6 SID route is matched in the FIB due to longest prefix match. Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

The Destination Address of the packet changes at each segment termination/completion and therefore the final DA of the packet MUST be encoded as the last segment of the path.

It needs to carry out a SRH operation on the basis of this Next Header. If the Next Header still has bits value 43 then there are more SIDs in the SRH and it needs to program the Next SID in the SRH Segment List into the IPv6 Destination and reduce the segment-left value by 1.

A best effort ISIS SRv6 Path, where there is no SRH, the only VPN SID is copied to the Destination Address of the Packet. The Next Header of the outer IPv6 header will not have 43 as there is no extension header.

A SRv6 TE Path or SRv6 TI-IFA path can have more than 1 SID in the segment list. This will impose the value of 43 in the Next Header of the outer IPv6 header and the Next Header of the SRH will carry a value depending on the actual payload as mentioned above. (More about the operations of SRH continues in the next section.)

There can be cases where:

IPv6 Header [Next Header 43]>SRH [Next Header 41] > IPv6 Header [Next Header 43]> SRH [Next Header 41] > IPv6 Payload

IPv6 Header [Next Header 43]>SRH1 [Next Header 43] >SRH [Next Header 41] > IPv6 Payload

1.2.4 SRH Operations in a SRv6 Network

Let's explain some of the abbreviations used Figure 1.4. NH means Next Header and this is the Next Header mentioned in the previous section. DA is destination address, SRH is Segment routing header.

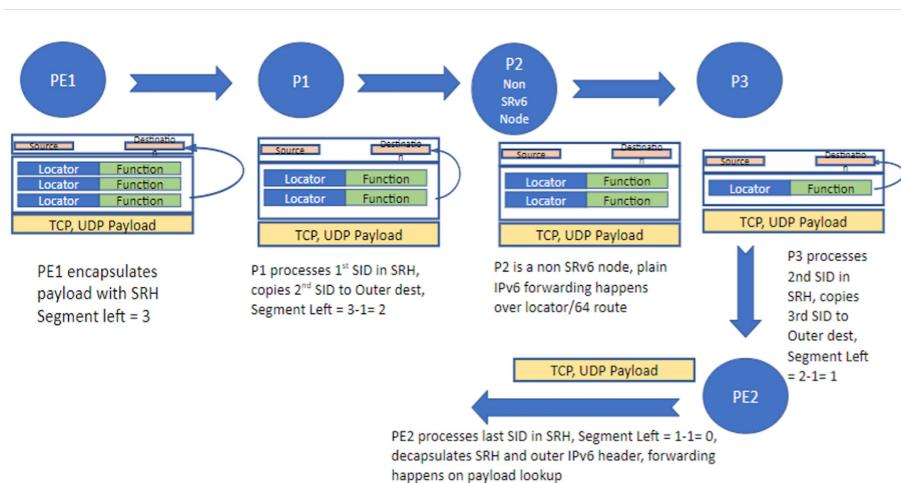


Figure 1.4 SRH and Outer IPv6 Header Processing in a SRv6 Topology

The assumption of this topology depicted in Figure 1.4 is that PE1 has SRv6 TE as transport with a Segment List visiting P1, P3, and PE2 L3 VPN. Referring to the figure, you can see what is happening.

- PE1 encapsulates the payload (here let's take an IPv4 packet which has a protocol value 4), places the first SRv6 SID (End SID of P1) in the DA of the outer IPv6 header. Since there are more SRv6 SIDs in the stack, a SRH is created and the SIDs are placed in SRH. The last SID is End.DT4 as the original service payload is IPv4.
- P1 processes the DA, as the DA is the End SID of P1 and P1 has a forwarding state with dedicated functions attached to it. P1 copies Next SRv6 SID to DA, reduces the Segment remaining by 1. Since SL (Segment Left > 0), it forwards the packet to the next node by looking up the DA.
- P2 is a non-SRv6 Node and it is just plain IPv6 forwarding in P2 based on the locator route.
- At P3, DA = End SID of P3. Follow step 2. Now since SL = 0, the next node cannot process the SRH. So, copy the NH of SRH which is equal to IPv4 packet (4) to IPv6 NH.
- At PE3, DA = End.DT4 SID. IPv6 NH = 4, pop the IPv6 encapsulation along with all its extension and forward the packet to the inet.0 of the corresponding VPN instance.
- At PE3 VPN instance forward packet based on actual payload lookup.

1.3 Network Topology

In the previous sections we have talked about SRv6. Here we will meet with our legacy network topology where there are 4 CE routers each having external BGP sessions with one PE router on an L3-VPN instance. So, in total we have four PE routers as well. Each of those four PE routers has internal BGP sessions with the other three PE routers exporting all the VPN prefixes to its internal BGP peers.

- The IGP reachability of each node in the core network is delivered by ISIS.
- The CEs will play the Customer Edge roles, PEs will play the provider edge roles and the core is formed by the P routers which are providers.
- In the legacy network IPv4 traffic from CEs are carried by LDP transport.

This topology in consideration is shown in Figure 1.5 and 1.6 represents the topology with its links, nodes, and IPv4 address details.

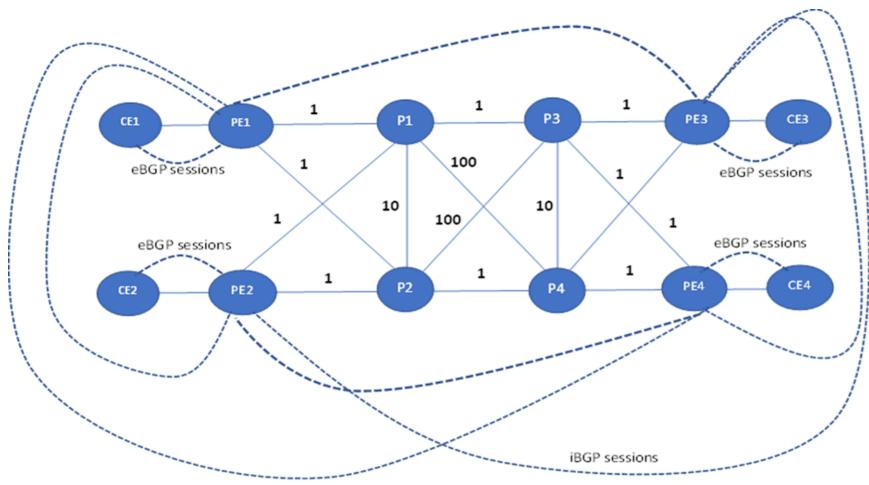


Figure 1.5 A simple representation of the topology in consideration for this Day One Book

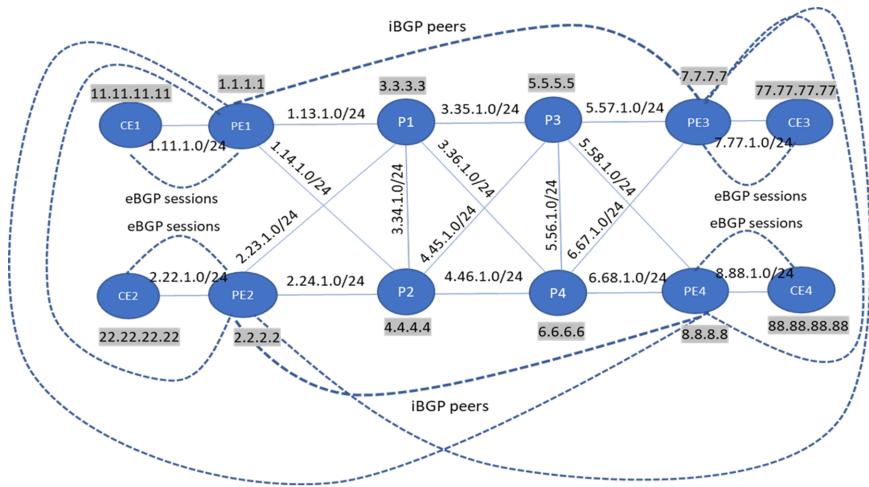


Figure 1.6 Depicting All IPv4 Link Addresses of the Topology in Consideration

In this consideration topology each CE is configured as a logical system on its corresponding PE, therefore the CE1 is a logical system on PE1, CE2 is a logical system of PE2, CE3 is a logical system of PE3, and CE4 is a logical system of PE4.

We start with a flat ISIS L2 domain in the legacy network. In later chapters we will also see multi-domain networks.

Here's an example of L3-VPN services on PE1.

Towards CE2:

```
root@PE1# run show route 22.22.22.22
L3VPN_PE1_CE1.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

22.22.22.22/32 *[BGP/170] 00:37:21, localpref 100, from 2.2.2.2
    AS path: 100 I, validation-state: unverified
        > to 1.13.1.2 via ge-0/0/2.0, Push 16, Push 18(top)
            to 1.14.1.2 via ge-0/0/3.0, Push 16, Push 16(top)
```

The above address 22.22.22.22 is configured as the loopback address of CE2 which is sitting behind PE2. Here we verified the route towards CE2's loopback address and validated that the underlay transport used is MPLS.

Towards CE3 (same as above but this is towards CE3):

```
[edit]
root@PE1# run show route 77.77.77.77
L3VPN_PE1_CE1.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
77.77.77.77/32 *[BGP/170] 00:34:47, localpref 100, from 7.7.7.7
    AS path: 100 I, validation-state: unverified
        > to 1.13.1.2 via ge-0/0/2.0, Push 16, Push 22(top)
            to 1.14.1.2 via ge-0/0/3.0, Push 16, Push 22(top)
```

Towards CE4:

```
[edit]
root@PE1# run show route 88.88.88.88
L3VPN_PE1_CE1.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
88.88.88.88/32 *[BGP/170] 00:37:34, localpref 100, from 8.8.8.8
    AS path: 100 I, validation-state: unverified
        > to 1.13.1.2 via ge-0/0/2.0, Push 16, Push 21(top)
            to 1.14.1.2 via ge-0/0/3.0, Push 16, Push 21(top)
```

The transport in this example is contributed by LDP:

```
root@PE1# run show route table mpls.0
    mpls.0: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

21      *[LDP/9] 00:38:36, metric 1
        > to 1.14.1.2 via ge-0/0/3.0, Swap 20
22      *[LDP/9] 00:38:19, metric 1
        > to 1.13.1.2 via ge-0/0/2.0, Swap 21
        to 1.14.1.2 via ge-0/0/3.0, Swap 21
23      *[LDP/9] 00:38:36, metric 1
        > to 1.13.1.2 via ge-0/0/2.0, Swap 22
        to 1.14.1.2 via ge-0/0/3.0, Swap 22
```

As you can see, our legacy network's underlay transport is LDP-based . The data plane is MPLS.
Verifying ping from CE1 to CE2:
root@PE1# run ping 22.22.22.22 logical-system CE1
PING 22.22.22.22 (22.22.22.22): 56 data bytes
64 bytes from 22.22.22.22: icmp_seq=0 ttl=61 time=4.757 ms
64 bytes from 22.22.22.22: icmp_seq=1 ttl=61 time=4.829 ms

```
64 bytes from 22.22.22.22: icmp_seq=2 ttl=61 time=3.985 ms
64 bytes from 22.22.22.22: icmp_seq=3 ttl=61 time=4.198 ms
^C
--- 22.22.22.22 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.985/4.442/4.829/0.360 ms
```

The same can be validated among each PE.

1.3.1 Enabling v6 Addresses on CEs

You enable IPv6 services by configuring IPv6 addresses on CEs.

Global iBGP sessions among all four PEs are configured with family inet6-vpn unicast. Transport layer has MPLS ipv6-tunneling enabled, and the typical 6PE style services are configured. With these configurations, IPv6 addresses of CEs can be reachable over the LDP core. Here CE to CE IPv6 service payload will be transported over legacy IPV4 LDP core (eBGP sessions between CE and PE continue to be v4 sessions.) Refer to Figure 1.7 for CE to PE v6 addresses.

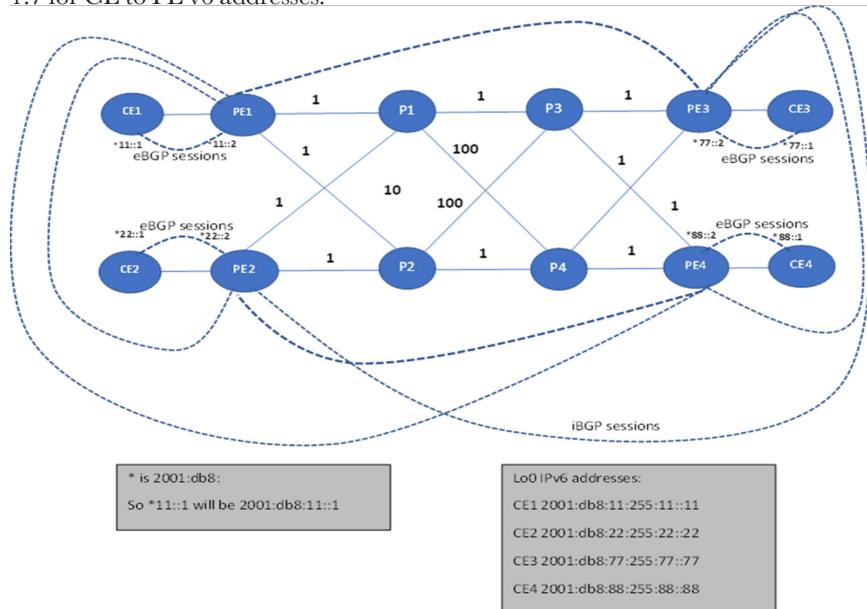


Figure 1.7 IPv6 Addresses on CE to PE

Config change

- Under each PE global bgp config:

```
set protocols bgp group ibgp family inet6-vpn unicast
```

This will enable internal BGP sessions among PEs to exchange their L3-VPN IPv6 service prefixes and resolve over legacy LDP Core.

2. Configure family inet6 unicast under each PE to CE eBGP sessions.

On PE1 and CE1:

```
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv4 family inet6 unicast
set logical-systems CE1 protocols bgp group eBGPv4 family inet6 unicast
```

Similar configurations are done in all the PEs and CEs. This will enable external BGP sessions between PE and CE to exchange CE's IPv6 prefixes to PEs L3-VPN instance.

3. Enable IPv6-tunneling under mpls for all routers:

```
set protocols mpls ipv6-tunneling
```

This enables BGP to tunnel IPv6 transport over IPv4 core.

4. Enable IPv6 address on PE to CE interface:

```
set logical-systems CE1 interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:11::1/48
set logical-systems CE1 interfaces lo0 unit 1 family inet6 address 2001:db8:11:255:11::11/128
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:11::2/48
```

Similarly in all CEs the IPv6 configuration is applied.

5. Enable eBGP export policies in CE to export all inet6 routes to its eBGP peer and configure the next-hop of those inet6 prefixes to its eBGP interface local address:

CE1:

```
set logical-systems CE1 policy-options policy-statement BGP_export term 1 from family inet6
set logical-systems CE1 policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:11::1
set logical-systems CE1 policy-options policy-statement BGP_export term 1 then accept
```

Similarly, in all the CEs this configuration is applied.

Verification of IPv6 L3-VPN Routes on PE1 to Other CEs:

Once these configs are applied, the CEs advertise their IPv6 addresses to corresponding eBGP peer PEs and from there PEs advertise it to other PEs through BGP NLRIs over iBGP sessions. That's how all the PE's L3-VPN instances are aware of all CEs' IPv6 address reachability. Here is an example from PE1 to other CEs IPv6 network reachability in L3-VPN instance, towards CE2's lo0 IPv6 Address:

```
[edit]
root@PE1# run show route 2001:db8:22:255:22::22
L3VPN_PE1_CE1.inet6.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:22:255:22::22/128
 *[BGP/170] 01:28:52, localpref 100, from 2.2.2.2
     AS path: 100 I, validation-state: unverified
     > to 1.13.1.2 via ge-0/0/2.0, Push 16, Push 18(top)
     to 1.14.1.2 via ge-0/0/3.0, Push 16, Push 16(top)
```

Similarly, one can verify the route towards each CE from a PE.

Ping from CE1 to Other CE2s' IPv6 Addresses

Here's the ping from CE1, which is the Logical Systems on PE1 in our topology, towards CE2's lo0 Address:

```
root@PE1# run ping 2001:db8:22:255:22::22 logical-system CE1
PING6(56=40+8+8 bytes) 2001:db8:11::1 --> 2001:db8:22:255:22::22
16 bytes from 2001:db8:22:255:22::22, icmp_seq=0 hlim=61 time=5.446 ms
16 bytes from 2001:db8:22:255:22::22, icmp_seq=1 hlim=61 time=4.801 ms
16 bytes from 2001:db8:22:255:22::22, icmp_seq=2 hlim=61 time=4.756 ms
^C
--- 2001:db8:22:255:22::22 ping6 statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 4.756/5.001/5.446/0.315 ms
```

```
root@PE1# run traceroute 2001:db8:22:255:22::22 logical-system CE1
traceroute6 to 2001:db8:22:255:22::22 (2001:db8:22:255:22::22) from 2001:db8:11::1, 64 hops max, 12
byte packets
 1  2001:db8:11::2 (2001:db8:11::2)  5.764 ms  2.140 ms  1.668 ms
 2  * * *
 3  2001:db8:22::2 (2001:db8:22::2)  38.173 ms  26.513 ms  3.680 ms
 4  2001:db8:22:255:22::22 (2001:db8:22:255:22::22)  5.482 ms  5.475 ms  4.798 ms
```

Similarly other CE-to-CE connectivity can be verified this same way.

1.4 Enabling IPv6 in the Legacy Network

Let's talk about enabling IPv6 addresses in the transport network domain and IPv6 eBGP and iBGP sessions. It's the first stepping stone in moving our network towards SRv6.

1.4.1 Enabling IPv6 on all Interfaces in the Network:

Enable IPv6 addresses on all interfaces on PEs and P routers to make the core Network IPv6 aware.

On PE1:

```
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:13::1/64
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:14::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:1:255:1::1/128
```

NOTE Refer to the Appendices for the configurations on all nodes.

ISIS can operate IPv4 and IPv6 in two different modes. One is *dual stack* mode and another is *multi-topology* mode. In dual stack mode a single SPF is run and the IPv4 routes, as well as IPv6 routes, derive next hops based on this SPF. (Dual stack mode is used throughout the book.) Since single SPF derives IPv4 next hops as well as IPv6 next hops, it is imperative that the IPv4 and IPv6 topologies are congruent. If certain interfaces are assigned IPv4 addresses but not assigned IPv6 addresses, it could result in traffic blackholes.

Let's verify our ISIS interfaces and adjacencies. In the interest of saving time and avoiding lengthy outputs in the book, let's verify this output only on PE1 and P1. All the other nodes will have the same output and will be left to the reader to bringing up the setup on their own:

```
[edit]
root@PE1# run show isis adjacency detail
P1
  Interface: ge-0/0/2.0, Level: 2, State: Up, Expires in 24 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2d 14:06:53 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  IP addresses: 1.13.1.2
  IPv6 addresses: fe80::5603:7ff:fe00:3ff0
  IPv6 Global Interface Address: 2001:db8:13::2

P2
  Interface: ge-0/0/3.0, Level: 2, State: Up, Expires in 19 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 2d 14:05:48 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  IP addresses: 1.14.1.2
  IPv6 addresses: fe80::5603:7ff:fe00:3fe4
  IPv6 Global Interface Address: 2001:db8:14::2
```

Let's verify some ISIS IPv6 Shortest Path from PE1 to other PEs as shown in Figure 1.8 depicting the ISIS Shortest path towards each PE from PE1.

```
[edit]
root@PE1# run show route table inet6.0 protocol isis

inet6.0: 29 destinations, 29 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:d00::/24    *[IS-IS/18] 15:45:48, metric 3
                  to fe80::5603:7ff:fe00:3ff0 via ge-0/0/2.0
                  > to fe80::5603:7ff:fe00:3fe4 via ge-0/0/3.0
2001:db8:2:255.2::/128   blue arrow pointing right
                         *[IS-IS/18] 15:45:57, metric 2
                         > to fe80::5603:7ff:fe00:3ff0 via ge-0/0/2.0
                         to fe80::5603:7ff:fe00:3fe4 via ge-0/0/3.0
2001:db8:3:255.3::/128   blue arrow pointing right
                         *[IS-IS/18] 15:45:57, metric 1
                         > to fe80::5603:7ff:fe00:3ff0 via ge-0/0/2.0
2001:db8:4:255.4::/128   blue arrow pointing right
                         *[IS-IS/18] 15:45:57, metric 1
                         > to fe80::5603:7ff:fe00:3fe4 via ge-0/0/3.0
2001:db8:5:255.5::/128   blue arrow pointing right
                         *[IS-IS/18] 15:45:57, metric 2
                         > to fe80::5603:7ff:fe00:3ff0 via ge-0/0/2.0
2001:db8:6:255.6::/128   blue arrow pointing right
                         *[IS-IS/18] 15:45:57, metric 2
                         > to fe80::5603:7ff:fe00:3fe4 via ge-0/0/3.0
```

Figure 1.8 ISIS Shortest Path from PE1 to other PEs

Chapter 2

Configuring SRv6 Transport

2.1 Locators

Locators are the most important constructs in SRv6. Every node in the network gets associated with one or more locators. The SIDs associated with the locators are derived from the locators. Locators get advertised in the IGP as normal prefixes and are routable within the IGP domain. Locators can be allocated from the Global Unicast Address (GUA RFC 4291) space or from Unique Local Address (ULA RFC 4193) space. ULA addresses are well known and are easy to filter at the boundaries and are prevented from leaking outside the domain. ULA addresses have high probability of uniqueness due to the presence of globally unique index component in the address. Due to this uniqueness, if two different operator domains that are under independent administration get merged, the locator addresses from ULA space can continue to be used without the need for re-numbering. Locators and the SIDs derived from the locators are used to address the operator infrastructure nodes and functions. These addresses and SIDs need not be advertised into the internet. The locators and SIDs are confined to the operator's own domain or at most to a closely co-operating set of domains.

2.1.1 Deciding the Locator Size

Locator size is decided based on the various factors that are specific to operator's network topology, network size, growth prospects and slicing requirements.

Table 2.1 Example Locator Planning

Locator block (32 bit)				Node ID (32 bit)		
Prefix	Reserved	Flex-algo	Region	sub-region	Reserved	NE-ID
20 bits	4 bits	4 bits	4 bits	4 bits	20 bits	8 bits

Table 2.1 shows the locator address planning for an example network. Initial 20 bits are from the customer's allocated GUA address block or the well known ULA address starting with FC00::.

The next 4 bits are reserved for future expansion. Flex-algo has been allocated 4 bits which can provide up to 16 flex-algos in the network. Flex-algo is mostly used for network slicing and 16 topological slices are sufficient in most cases. The region-id is allocated 4 bits and allows for 16 instances. In larger networks which divide the network into regions, 4 bits may not be enough and will need more bits to be allocated for the region. The node ID part of the locator address consists of 32 bits in this example with 4 bits of ub-region id and 8 bits of node ID. 20 bits of reserved space is allocated in Node ID part and it can be used for other features such as anycast addresses or for future innovation.

The size of the locator and the number of bits allocated for each category is specific to the network and is fully in control of the operator.

2.1.2 Configuring locators

Let us configure locators on our sample network. We use the 2001:db8:: address range which is allocated IPv6 range for documentation. We use 48 bits of locator block consisting of 4 bit flex-algo and 4 bit sub-region followed by 16 bits of Node ID space where 8 bits used for NE-ID.

Eight bits of reserved space in Node-ID space can be used for allocating anycast addresses.

Table 2.2 Locator Design for Lab Topology

Locator Block(48 bits)				Node id (16 bits)	
Prefix	Reserved	Flex-algo	Sub-region	Reserved	NE ID
32 bits	8 bits	4 bits	4 bits	8 bits	8 bits
2001:db8	0	a	1	0	1

Locators are configured under routing options and get associated with ISIS SID configuration.

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:1::/64
```

The nodes and the individual locator address configuration are shown in Figure 2.1.2.

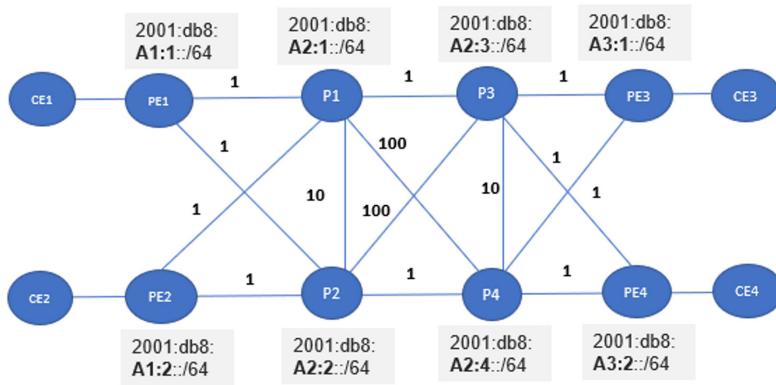


Figure 2.1 Locator Assignment

NOTE Commands for configuring locators on each node are in the Appendices.

In order to get the SRv6 transport ready, the SIDs corresponding to the locators need to be configured on each node. For example, let's look at the locator routes in inet6.0:

```
regress@PE1> show route table inet6.0
...
2001:db8:a1:2::/64 *[IS-IS/18] 00:03:17, metric 2
    > to fe80::5668:adff:fec0:3f12 via ge-0/0/2.0
        to fe80::5668:adff:fec0:3e73 via ge-0/0/3.0
2001:db8:a2:1::/64 *[IS-IS/18] 00:03:11, metric 1
    > to fe80::5668:adff:fec0:3f12 via ge-0/0/2.0
2001:db8:a2:2::/64 *[IS-IS/18] 00:03:06, metric 1
    > to fe80::5668:adff:fec0:3e73 via ge-0/0/3.0
2001:db8:a2:3::/64 *[IS-IS/18] 00:03:01, metric 2
    > to fe80::5668:adff:fec0:3f12 via ge-0/0/2.0
2001:db8:a2:4::/64 *[IS-IS/18] 00:02:56, metric 2
    > to fe80::5668:adff:fec0:3e73 via ge-0/0/3.0
2001:db8:a3:1::/64 *[IS-IS/18] 00:02:52, metric 3
    > to fe80::5668:adff:fec0:3f12 via ge-0/0/2.0
        to fe80::5668:adff:fec0:3e73 via ge-0/0/3.0
2001:db8:a3:2::/64 *[IS-IS/18] 00:00:15, metric 3
    > to fe80::5668:adff:fec0:3f12 via ge-0/0/2.0
        to fe80::5668:adff:fec0:3e73 via ge-0/0/3.0
```

Locator routes are usual IPv6 routes and get programmed in the inet6.0 table like any other IPv6 prefix. The next hops for the locators in this table are normal router next hops that forward the traffic along the next hop interface. Locators also get programmed in the inet6.3 table. These routes have a tunnel next hop (of type SRv6). Tunnel next hops are used to encapsulate the incoming packet in an additional tunnel header based on the type of the next hop. In case of SRv6 tunnel type, packet gets encapsulated in additional IPv6 header and the SRH. In case of SRv6 the tunnel next hop also contains the SRH information.

```
regress@PE2# run show route table inet6.3
...
2001:db8:a2:1::/64 *[SRV6-ISIS/14] 2d 09:06:58, metric 1
    > to fe80::5603:1ff:fe00:56a7 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:: .
```

The command shows the locator route 2001:db8:a2:1::/64 in inet6.3 table. Notice the tunnel next hop in the inet6.3 table with destination address 2001:db8:a2:1:: .

2.2 SIDs and Flavors

The basic transport SIDs are End SID and End.X SIDs. End SIDs are similar to Node-SIDs in SR-MPLS and provide shortest path to the node from other nodes in the network. End.X SIDs are similar to adjacency SIDs in SR-MPLS. End.X SIDs help to steer the traffic on a particular interface.

The SIDs are associated with flavors. A SID can be configured with no flavor. SIDs with no flavors can only be used as middle SIDs in a SID stack. There are three flavors defined in RFC 8986, namely, *Penultimate Segment Pop* (PSP), *Ultimate Segment Pop* (USP), and *Ultimate Segment Decapsulate* (USD). When a node advertises a PSP SID, if this PSP SID is the last SID in the SID stack, then the penultimate node of this node must remove the Segment Routing Header (SRH) and forward the packet. If the USP flavor is set on the SID then this node removes the SRH header if the SID is the last SID in the stack whose Segment Left is zero. (SL=0).

If the USD flavor is set on the SID the node will remove SRH as well as decapsulate the additional IPv6 header on the packet. A node may advertise separate SIDs for each flavor or advertise single SIDs with one or more flavors. The capability of the forwarding ASIC decides whether a node can support single SID with all flavors or separate SIDs for each flavor. Juniper MX and ACX Series are both capable of advertising all flavors on a single SID or a combination of one or more flavors on the SID. If the node is capable of supporting all flavors on the SID, it is a good practice to set all flavors on the SID to reduce the number of SIDs a node needs to advertise and hence contributing to better scalability.

NOTE Chapter 4 on TI-LFA gives more details on how the SIDs are chosen based on flavor while constructing TI-LFA SID stack.

2.2.1 End SID Configuration

Look at this Junos End SID configuration:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:12:0::0 flavor psp
```

The configuration shows two different End SIDs one with flavor USD and another with flavor PSP. The function length used in 48 bits and arguments of 16 bits. In this book we are not covering any use case that requires argument. A larger argument space can be reserved for future evolution of use cases that would have to use arguments.

Here are the End SID routes on PE1:

```
regress@PE1> show route table inet6.0

...
2001:db8:a1:1::/64 *[IS-IS/18] 00:03:54, metric 0
    Reject
2001:db8:a1:1:11::/128
    *[IS-IS/18] 00:03:54, metric 0
        Receive
2001:db8:a1:1:12::/128
    *[IS-IS/18] 00:03:54, metric 0
        Receive
```

The End SID routes are programmed on the node with a receive next hop. When any packet having the End SID address in the IPv6 destination field is received on the node, the lookup hits this End SID route and results into SRH processing. If SL=0, the SRH is removed and if the SID has USD flavor, the additional IPv6 header decapsulation is also done. If SL!=0 then SL is decremented by 1 and the next segment in SRH is copied into the destination address of the IPv6 header. The packet is then forwarded based on the destination address lookup.

A receiving router will compute a shortest path to the End SID and place the End SID with USD flavor in the inet6.3 table with a tunnel next hop. These End SIDs may be used to resolve service prefixes if End SID IPv6 address are used as the the BGP protocol next hop in the service advertisement.

```
regress@PE2> show route table inet6.3

...
2001:db8:a2:1:11::/128
    *[SRV6-ISIS/14] 00:05:14, metric 1
        > to fe80::5603:1ff:fe00:5513 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:11::
```

Note that the End SID 2001:db8:a2:1:11:: advertised by P1 is programmed on PE2 with a tunnel next hop.

2.2.2 End.X SID Configuration

Let's now configure End.X SIDs. These SIDs are configured under the ISIS interface:

```
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:213:0:0:0 flavor psp usp usd
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:214:0:0:0 flavor psp usp usd
```

The **unprotected** keyword implies that the **end-x-sid** will not have any backup path programmed for the SID. If the **protected** keyword is configured the End.X SID will also have a link programmed protecting the backup path (computed using TI-LFA). Protected End.X SIDs are not supported in Junos as of writing this book.

Here's the End.X route on PE1:

```
regress@PE1> show route table inet6.0

...
2001:db8:a1:1:213::/128
```

```
*[IS-IS/18] 00:07:20, metric 0
    > to fe80::5603:1ff:fe00:5513 via ge-0/0/2.0
2001:db8:a1:1:214::/128
*[IS-IS/18] 00:07:20, metric 0
    > to fe80::5603:1ff:fe00:3794 via ge-0/0/3.0
```

The End.X route has a valid next hop that points to the interface on which the SID is configured.

When a packet is received on the node with End.X SID in the destination address, the SL is decremented and the next SID is copied into the IPv6 destination address and the packet is forwarded on the interface which is programmed in the next hop. If SL=0, then the SRH is removed if it is a PSP-flavored SID.

2.3 SRv6 Ping and Traceroute

Once the locators and SIDs are configured, you can use SRv6 ping and traceroute to validate the connectivity to the SIDs.

```
regress@PE1# run ping srv6 isis 2001:db8:a3:1:11:0::
!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.807/5.207/6.296/0.552 ms
[edit]
regress@PE1> traceroute srv6 isis 2001:db8:a3:1:11:0:
traceroute to destination 2001:db8:a3:1:11::, from source 2001:db8:1:255:1::1, 64 hops max, retries 3
 1  2001:db8:13::2  2.378 ms  1.459 ms  1.846 ms
      DA: 2001:db8:a3:1:11::
 2  2001:db8:35::2  3.231 ms  3.016 ms  2.516 ms
      DA: 2001:db8:a3:1:11::
 3  2001:db8:57::2  6.158 ms  4.819 ms  4.167 ms
      DA: 2001:db8:a3:1:11::
Traceroute status: Success
```

Chapter 3

Configuring SRv6 Services

In Chapter 2 we saw how to configure SRv6 locators and End and End.X SIDs in the network. For us to get an IPv6 transport only network we need to make our services aware of SRv6. This chapter will introduce the configuration of End.DT4, End.DT6 SRv6 SIDs through which now service payloads can be delivered through core to its corresponding L3-VPN instance. This chapter will demonstrate L3 VPN services over Best Effort (i.e., shortest path) SRv6 Transport where no SRH will be present in the data plane.

3.1 What is End.DT4 and End.DT6 SIDs

The “Endpoint with decapsulation and specific IPv6 table lookup” behavior (End.DT6 for short) is a variant of the End.T behavior. One of the applications of the End.DT6 behavior is the L3VPNv6 use case where a FIB lookup in a specific tenant table at the egress PE is required. This is equivalent to the per-VRF VPN label in MPLS. Note that an End.DT6 may be defined for the main IPv6 table, in which case an End.DT6 supports the equivalent of an IPv6-in-IPv6 decapsulation (without VPN/tenant implication).

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

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

```
S01. When an SRH is processed {  
S02. If (Segments Left != 0) {  
S03. 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.  
S04. }  
S05. Proceed to process the next header in the packet  
S06. }
```

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

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

End.DT4: Decapsulation and Specific IPv4 Table Lookup

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

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

Note that an End.DT4 may be defined for the main IPv4 table, in which case an End.DT4 supports the equivalent of an IPv4-in-IPv6 decapsulation (without VPN/tenant implication).

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

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

```
S01. When an SRH is processed {  
S02. If (Segments Left != 0) {  
S03. 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.  
S04. }  
S05. Proceed to process the next header in the packet  
S06. }
```

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

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

3.2 Migrating Services to SRv6

3.2.1 SRv6 service SIDs

While configuring the End.DT4 and End.DT6 SIDs in L3-VPN instances of all the PEs, you must make sure the SIDs belong to the locator to which that PE belongs.

For example, let's configure SRv6 End.DT4 and End.DT6 SIDs under L3-VPN routing-instances of all PEs:

```
PE1
set routing-instances L3VPN_PE1_CE1 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a1:1:3111:0:0:0
set routing-instances L3VPN_PE1_CE1 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a1:1:4111:0:0:0
#end-dt4-sid : <loc1> = 2001:db8:a1:1 function: 3111 argument: 0:0:0
#end-dt6-sid : <loc1> = 2001:db8:a1:1 function: 4111 argument: 0:0:0
```

PE2

```
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a1:2:3222:0:0:0
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a1:2:4222:0:0:0
#end-dt4-sid : <loc1> = 2001:db8:a1:2 function: 3222 argument: 0:0:0
#end-dt6-sid : <loc1> = 2001:db8:a1:2 function: 4222 argument: 0:0:0
```

PE3

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3777:0:0:0
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:1:4777:0:0:0
#end-dt4-sid : <loc1> = 2001:db8:a3:1 function: 3777 argument: 0:0:0
#end-dt6-sid : <loc1> = 2001:db8:a3:1 function: 4777 argument: 0:0:0
```

PE4

```
set routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:2:3888:0:0:0
set routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:2:4888:0:0:0
#end-dt4-sid : <loc1> = 2001:db8:a3:2 function: 3888 argument: 0:0:0
#end-dt6-sid : <loc1> = 2001:db8:a3:2 function: 4888 argument: 0:0:0
```

The locator part of the above Service SIDs are the same as the locators defined in Chapter 2 in each PE. The function or arguments changes here.

Let's take one pair of service SIDs and peek into the details of its function and arguments.

PE3's End.DT4 SID is:

```
2001:db8:a1:1:3111:0:0:0:
```

The first 64 bits represent the Locator to which this SID belongs. In this case 2001:db8:a1:1 is the Locator.

The next 16 bits, therefore 3111, is our Function describing the SID behavior in the data plane where this SID is configured. In this case PE3. By default, in the Junos 22.3 release, the function length is 16. But one can configure it to have a 32 bit Function length by configuring the following from 22.3 onwards:

```
set routing-options source-packet-routing srv6 locator loc1 function-length 32
```

In this case, since the last 48 bits are the argument of the SID and our argument in this case is 0. The same holds True for End.DT6 SID on PE3:

```
2001:db8:a1:1:4111:0:0:0:
```

The first 64 bits represent the Locator to which this SID belongs. In this case 2001:db8:a1:1 is the Locator. The next 16 bits, therefore 4111, is our Function describing the SID behavior in the data plane where this SID is configured. In our case PE3.

In the above case, since the last 48 bits are the Argument of the SID and our Argument in this case is 0.

The router where End.DT4/DT6 SID is configured, the route for that SID will be downloaded to inet6.0 having the VPN context. The route's IPv6 prefix in inet6.0 will be the same as the Service SID and the next hop will point to the corresponding VPN routing table where the payload prefix will be looked up. Unlike ISIS End SID or End.X SID configurations, a service SID cannot be configured with a flavor. The SID will have a decapsulation flavor by default.

The SRv6 Service SID will be programmed in the inet6.0 routing table and in the FIB with the VPN context, meaning that the next hop of this Prefix will clearly say to which VPN must be the decapsulated payload packet forwarded.

The IPv6 prefix of this route is the same as the SRv6 SID.

Verifying End.DT4 SID on PE1:

```
root@PE1# run show route 2001:db8:a1:1:3111:0:0:0 extensive expanded-nh
inet6.0: 43 destinations, 43 routes (43 active, 0 holddown, 0 hidden)
2001:db8:a1:1:3111::/128 (1 entry, 1 announced)
Installed-nexthop:
Table (0x747d044) Index:642
TSI:
KRT in-kernel 2001:db8:a1:1:3111::/128 -> {Table}
Opaque data client: SRV6
Address: 0x8fde8e4
Opaque-data reference count: 2
*BGP Preference: 170
```

```
Next table: L3VPN_PE1_CE1.inet.0 >> This will ensure the Outer header is decap and Next
Lookup in L3-VPN inet.0 table
  Next-hop index: 642
  Address: 0x747d044
  Next-hop reference count: 3, key opaque handle: 0x0, non-key opaque handle: 0x0
  State: <Active NoReadvrt Int Ext OpaqueData>
  Age: 8:30
  Validation State: unverified
  Task: BGP_RT_Background
  Announcement bits (3): 0-KRT 6-Resolve tree 5 9-Resolve tree 10
  AS path: I
  Route-nexthop:
  Table (0x747d044) Index:642
  Thread: junos-main
```

As you can see in the case of End.DT4 SID:

- The routing table is inet6.0
- Next Table to lookup after decapsulation is L3VPN's inet.0. This clearly indicates that the payload is an IPv4 packet.

Verifying End.DT6 SID:

```
[edit]
root@PE1# run show route 2001:db8:a1:1:4111:0:0:0 extensive expanded-nh
inet6.0: 43 destinations, 43 routes (43 active, 0 holddown, 0 hidden)
2001:db8:a1:1:4111::/128 (1 entry, 1 announced)
Installed-nexthop:
Table (0x747f504) Index:674
TSI:
KRT in-kernel 2001:db8:a1:1:4111::/128 -> {Table}
Opaque data client: SRV6
Address: 0x8fe0d44
Opaque-data reference count: 2
  *BGP    Preference: 170
    Next table: L3VPN_PE1_CE1.inet6.0>> This will ensure the Outer header is decap and Next
Lookup in L3-VPN inet.6 table
      Next-hop index: 674
      Address: 0x747f504
      Next-hop reference count: 2, key opaque handle: 0x0, non-key opaque handle: 0x0
      State: <Active NoReadvrt Int Ext OpaqueData>
      Age: 10:13
      Validation State: unverified
      Task: BGP_RT_Background
      Announcement bits (3): 0-KRT 6-Resolve tree 5 9-Resolve tree 10
      AS path: I
      Route-nexthop:
      Table (0x747f504) Index:674
      Thread: junos-main
```

As you can see in the case of End.DT6 SID:

- The routing table is inet6.0
- Next Table to Lookup after decapsulation is L3VPN's inet6.0. This clearly indicates that the payload is an IPv6 packet.

The same can be verified on all PE routers, an exercise left best to the reader.

3.2.2 Enable iBGP Peer Advertisements and Accept SRv6 Services

With SRv6 service SIDs in place now we need to enable iBGP v4 peers to advertise its services with its own SRv6 service SIDs and at the same time accept other services from its iBGP v4 peers with peer's SRv6 service SIDs.

When we advertise SRv6 service SIDs in iBGP, v4 services gets mapped to End.DT4 SIDs and v6 services get mapped to End.DT6 SIDs.

Enable this configuration on all PE routers:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet6 unicast extended-nexthop-color
```

Once the config is in place, you'll see the advertisement from iBGP with SRv6 service SID associated with the L3-VPN prefixes. For example, on PE1 let's verify what PE2 is advertising over iBGP. Note this advertised route should be the same on Egress and Ingress PE. In this example, let's take PE1 as ingress and PE3 as egress. You can verify the each taking of ingress and egress.

Let's check what Egress PE3 advertises to PE1 (showing one IPv4 and one IPv6 service prefix). PE1's IPv4 Loopback address is 1.1.1.1. This next command shows what routes are advertised by bgp to peer 1.1.1.1:

```
[edit]
root@PE3# run show route advertising-protocol bgp 1.1.1.1
L3VPN_PE3_CE3.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (2 entries, 1 announced)
  BGP group ibgpv4 type Internal
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1:: >> The NextHop that is configured in policy-options
    Flags: Nexthop Change
    Localpref: 100
    AS path: [64512] I
    Communities: target:64512:1
      SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0 >> SRv6
End.DT4 SID that is configured under eBGP in L3 VPN routing-instance
...
...
L3VPN_PE3_CE3.inet6.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (2 entries, 1 announced)
  BGP group ibgpv4 type Internal
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1:: >> The NextHop that is configured in policy-options
    Flags: Nexthop Change
    Localpref: 100
    AS path: [64512] I
    Communities: target:64512:1
      SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0 >> SRv6
End.DT6 SID that is configured under eBGP in L3 VPN routing-instance
```

The same routes should be present on PE1 as receiving routes from peer PE3. PE3's IPv6 Loopback address is 7.7.7.7. The next command shows what route is received by protocol bgp from peer 7.7.7.7:

```
[edit]
root@PE1# run show route receive-protocol bgp 7.7.7.7 extensive
inet.0: 40 destinations, 40 routes (40 active, 0 holddown, 0 hidden)
inet.3: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
L3VPN_PE1_CE1.inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1::
    Localpref: 100
    AS path: I
    Communities: target:64512:1
        SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
    ...
    ...
L3VPN_PE1_CE1.inet6.0: 11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1::
    Localpref: 100
    AS path: I
    Communities: target:64512:1
        SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

As can be seen, the advertisement from PE3 is identical to the receiving route at PE1. This is how you can verify or troubleshoot any issue in these advertisements.

3.2.3 VPN inet-vpn/inet6-vpn Service Configuration

These configurations are required on ingress and egress PEs for BGP services to resolve over SRv6:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
```

These configurations were discussed in section 3.2.2, so now let's explain a bit more about these configs. Since the L3 VPN routes are advertised to all the peers through iBGP and since now all L3 VPN routes have End.DT4 and End.DT6 service SID mapping, the iBGP sessions in the global BGP configuration should allow advertisement of SRv6 Service SID from its own L3 VPN instance and accept advertisement of SRv6 Service SID from other peers.

This is achieved mainly by the configurations:

- **accept-srv6-service:** Accepts SRv6 Compliant Service advertisements from other PEs.
- **advertise-srv6-service:** Allows advertisement of SRv6 Compliant Service advertisements from its own L3 VPN instance.

Notice that the family for these configurations are `inet-vpn` and `inet6-vpn` which clearly indicates these are for VPN routes.

If you want SRv6 Service SID for global BGP advertised routes as well, then you need to configure these two SRv6 knobs under family `inet` and `inet6`.

By default, the protocol next-hop of all BGP advertised routes from L3-VPN instances will be a self loopback-address. However, for SRv6, you need to point all these advertisements to the locator routes from the PE. This can be done in two ways.

3.2.4 Change Next Hop of Services from Self-Loopback Address to SRv6 Service SIDs:

3.2.4.1 Service Mapping to SRv6 Locator prior to Junos 22.3 release:

On your egress you can add a policy to point all your BGP advertisements to the locator that the egress VPN instance belongs to, or by adding a simple export policy such as:

```
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 2001:db8:a3:1::  
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then community add L3VPN_1_COM  
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then accept
```

PE1:

```
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then community add L3VPN_1_COM  
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 2001:db8:a1:1::  
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then accept
```

PE2:

```
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then community add L3VPN_1_COM  
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then next-hop 2001:db8:a1:2::  
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then accept
```

PE3:

```
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COM  
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 2001:db8:a3:1::  
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then accept
```

PE4:

```
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then community add L3VPN_1_COM  
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then next-hop 2001:db8:a3:2::  
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then accept
```

This simplifies our config on all the PEs that are pointing to the locators of the SRv6 Service SIDs. For IPv4 prefixes BGP will now automatically pick up SRv6 End.DT4 and for v6 prefixes it will pick up End.DT6 SID and advertise it to the other PEs.

On the ingress the protocol next hop of these BGP service prefixes will be present as an SRv6 Tunnel in inet6.3.

Verifying a v4 and v6 service prefix route from PE3 on PE1

CE3's V4 prefix from PE1 over SRv6:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1:: >> NextHop is changed via policy config
    Localpref: 100
    AS path: I
    Communities: target:64512:1
        SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
...
...
```

CE3's V6 prefix from PE1 over SRv6:

```
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
...
...
L3VPN_PE1_CE1.inet6.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:a3:1:: >> NextHop is changed via policy config
    Localpref: 100
    AS path: I
    Communities: target:64512:1
        SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

Let's check our transport on the protocol next hop 2001:db8:a1:1:: in the ine6.3:

```
inet6.3: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a1:1::/64 *[SRV6-ISIS/14] 00:45:38, metric 3
    > to fe80::5603:7ff:fe00:63f8 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
        to fe80::5603:7ff:fe00:402b via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
```

Our service prefixes will resolve over this transport and will be downloaded to FIB:

```
[edit]
root@PE4# run show route 1.11.1.0 extensive
L3VPN_PE4_CE4.inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
1.11.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr (0x747d3c4) 2001:db8:a1:1::
```

```

Nexthop Reference count: 2
Composite Session-ID: 421
  Krt_cnh (0x6fd3174) Index:689
    Krt_inh (0x9302044) Index:1048590 PNH: 2001:db8:a1:1::
      List (0x79193d4) Index:1048588
      Nexthop Reference count: 13
      Nexthop flags: explicit-add on-nhid-tree unilist
        Chain (0x747ea14) Index:683
        Nexthop Reference count: 1
        Nexthop flags: explicit-add on-nhid-tree
        Fully resolved tunnel
        ELNH address: 0x747e544
        SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
        Src: 2001:db8:8:255:8::8 Dest: 2001:db8:a1:1::
        Segment-list[0] 2001:db8:a1:1::
          Installed-nexthop:
Router (0x747e544) Index:745 fe80::5603:7ff:fe00:63f8
  Nexthop Reference count: 34
  Nexthop flags: explicit-add on-nhid-tree
  Session-ID: 418 via ge-0/0/2.0
  Chain (0x747ea84) Index:685
  Nexthop Reference count: 1
  Nexthop flags: explicit-add on-nhid-tree
  Fully resolved tunnel
  ELNH address: 0x747e5b4
  SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
  Src: 2001:db8:8:255:8::8 Dest: 2001:db8:a1:1::
  Segment-list[0] 2001:db8:a1:1::
    Installed-nexthop:
    Router (0x747e5b4) Index:746 fe80::5603:7ff:fe00:402b
    Nexthop Reference count: 34
    Nexthop flags: explicit-add on-nhid-tree
    Session-ID: 419 via ge-0/0/3.0

```

Some important things to notice in this output:

- The protocol next hop is the locator of PE1 that is advertised in BGP to PE4 from PE1.
- The mode of the tunnel is an ENCAP mode. We will cover Encap and Insert mode later in this Chapter in section 3.4 and then in Chapter 7.
- The SRH mode is Reduced. We will explain that in detail later in this chapter.

Let's now verify ping and traceroute to these service prefixes over SRv6 transport, with ping CE1 prefixes (IPv4 and IPv6) from CE4:

```

[edit]
root@PE4# run ping 1.11.1.1 logical-system CE4
PING 1.11.1.1 (1.11.1.1): 56 data bytes
64 bytes from 1.11.1.1: icmp_seq=0 ttl=62 time=5.405 ms
64 bytes from 1.11.1.1: icmp_seq=1 ttl=62 time=5.069 ms
64 bytes from 1.11.1.1: icmp_seq=2 ttl=62 time=9.320 ms
^C
--- 1.11.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.069/6.598/9.320/1.930 ms

```

```
[edit]
root@PE4# run ping 2001:db8:11::1 logical-system CE4
PING6(56=40+8+8 bytes) 2001:db8:88::1 --> 2001:db8:11::1
16 bytes from 2001:db8:11::1, icmp_seq=0 hlim=62 time=5.992 ms
16 bytes from 2001:db8:11::1, icmp_seq=1 hlim=62 time=5.747 ms
16 bytes from 2001:db8:11::1, icmp_seq=2 hlim=62 time=5.117 ms
^C
--- 2001:db8:11::1 ping6 statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 5.117/5.619/5.992/0.369 ms
```

Traceroutes:

```
[edit]
root@PE4# run traceroute 1.11.1.1 logical-system CE4
traceroute to 1.11.1.1 (1.11.1.1), 30 hops max, 52 byte packets
1 8.88.1.2 (8.88.1.2) 1.906 ms 1.922 ms 1.577 ms
2 1.11.1.2 (1.11.1.2) 4.316 ms 4.583 ms 4.652 ms
3 1.11.1.1 (1.11.1.1) 5.815 ms 6.222 ms 5.544 ms
[edit]
root@PE4# run traceroute 2001:db8:11::1 logical-system CE4
traceroute6 to 2001:db8:11::1 (2001:db8:11::1) from 2001:db8:88::1, 64 hops max, 12 byte packets
1 2001:db8:88::2 (2001:db8:88::2) 2.320 ms 2.139 ms 1.866 ms
2 2001:db8:11::2 (2001:db8:11::2) 5.386 ms 4.674 ms 4.508 ms
3 2001:db8:11::1 (2001:db8:11::1) 6.118 ms 6.510 ms 5.501 ms
```

3.2.4.2 Service Mapping to SRv6 Locator from JUNOS 22.3 release:

Things get easier and simpler with Junos 22.3R1. From Junos 22.3R1 customers are recommended to use Transport Class to achieve simpler resolution schemes. To get an overview of what Transport Class entails, check out this material:

- <https://datatracker.ietf.org/meeting/108/materials/slides-108-idr-bgp-classful-transport-planes-00>
- Draft: <https://datatracker.ietf.org/doc/html/draft-kaliraj-idr-bgp-classful-transport-planes-02>

To achieve the full advantage of service resolutions in 22.3R1 the following configs are strongly recommended:

```
set routing-options transport-class auto-create
set protocols source-packet-routing use-transport-class
```

The first CLI config is required for the underlying Classful Transport infrastructure to get active and to ensure auto-creation of the transport RIBs. The second command is to enable SRv6 TE routes to be downloaded in auto created transport RIB. The SRv6 TE route also gets downloaded to the inet6color table as well but that inet6color is discouraged and may get discontinued in future releases. Transport Class is the new future.

The BGP service route also carries another property which is called SRv6 SID Transposition. It tries to transpose the SID in such a way that it can resolve over SRv6 Locators so that different VPN SIDs can be summarized under one umbrella:

- More info on this can be found in this draft: <https://datatracker.ietf.org/doc/html/draft-ietf-bess-srv6-services-03#section-3.2.1>.
- We'll discuss more on SRv6 TE in Chapter 10 where we will again give this context and show how resolution and fallback works.
- Service Mapping on SRv6 Flex Algo with Transport Class in Junos 22.3 release will be covered in Chapter 7.

Okay, let's continue with having only ISIS SRv6 routes.

Since we are not using any colored SRv6 TE policies or SRv6 Flex Algo in this chapter, Transport Class is not yet required. With Junos 22.3 the resolution schemes change and the next hop change through policy configurations at egress is no longer required. So, we don't need the following configurations any more:

```
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 2001:db8:a3:1:::  
This configuration is shown from PE3 and not that the same configurations are not required in other PEs  
as well. So without this configuration in Junos 22.3, the BGP advertisement now looks like this from  
PE3's advertisement to PE1:  
[edit]  
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive  
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)  
* 7.77.1.0/24 (1 entry, 1 announced) >>> IPv4 Service Prefix  
    Import Accepted MultiNexthop  
    Route Distinguisher: 7.7.7.7:1  
    VPN Label: 14199 >>> TRANSPOSED bits of 3777 are placed in Label  
    Nexthop: 2001:db8:7:255:7::7 >> BGP NH pointing to loopback address of PE3  
    Localpref: 100  
    AS path: I  
    Communities: target:64512:1 color:0:3  
        SRv6 SID: 2001:db8:a3:1:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 T0: 64 >> SRv6 SID  
transposed.  
    ... ... ...  
    ... ... ...  
    L3VPN_PE1_CE1.inet6.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)  
* 2001:db8:77::/48 (1 entry, 1 announced) >>> IPv6 Service Prefix  
    Import Accepted MultiNexthop RecvNextHopIgnored  
    Route Distinguisher: 7.7.7.7:1  
    VPN Label: 18295>>> TRANSPOSED bits of 4777 are placed in Label  
    Nexthop: 2001:db8:7:255:7::7 >> BGP NH pointing to loopback address of PE3  
    Localpref: 100  
    AS path: I  
    Communities: target:64512:1  
        SRv6 SID: 2001:db8:a3:1:: Behavior: 18 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 T0: 64
```

The SRv6 SID in this advertise is SRv6 SID: 2001:db8:a3:1::. If we closely look at the bits Behavior: 18 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 T0: 64, The TL: 16 is the number of bits that has to be transposed from the service/VPN SID. The T0: 64 is the offset from where the TL number of bits has to be transposed. So, in this case from the 64th bit, 16 bits need to be transposed.

NOTE For L3 VPN use cases, the MPLS Label value can at most contain 20 bits. So, if our function size is more than 20 bits the lower order 20 bits are transposed from SRv6 SID from an offset after 64bits. For example, if the function length is 24 bits, then our Transposed Offset will be 68 (locator length + number of bits not transposed from function) and our Transposed Length will be lower order 20 bits.

A simple example: On PE3 let our function length be 24 and our static SID max range is configured with 2^{24} value:
`set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:1::/64;`

```
set routing-options source-packet-routing srv6 locator loc1 function-length 24
set routing-options source-packet-routing srv6 locator loc1 static-function-max-entries 16777215
```

And our SRv6 SID is:

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3777:7700::0
```

As it can be seen here, our function length is 24 and our function in the SRv6 end-dt4-sid is 3777:77. But since our MPLS label can only hold 20 bits, 20 bits starting from the 68th offset are transposed. In this case 777:77 is transposed.

If we view the above SID as 2001:db8:a3:1:3777:7700::0, then we can clearly see 777:77 are the 20 bits that need to go in the MPLS label and transposed from the SRv6 SID. So, in PE1 the service advertisement from PE3 will look like:

```
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
  Import Accepted MultiNexthop
  Route Distinguisher: 7.7.7.7:1
  VPN Label: 489335 >>> Decimal value of hex 77777
  Nexthop: 2001:db8:7:255:7::7
  Localpref: 100
  AS path: I
  Communities: target:64512:1 color:0:3
    SRv6 SID: 2001:db8:a3:1:3000:: Behavior: 19 BL: 64 NL: 0 FL: 24 AL: 0 TL: 20 T0: 68 >>>
  777:77 is transposed to]
```

Our VPN SID from Egress PE3:

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3777:0:0:0
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:1:4777:0:0:0
```

And you can clearly see that “3777” and “4777” are transposed to “0” until 2001:db8:a3:1 is 64 bits and it belongs to the locator loc1 configured in the Egress PE3. After that “3777” and “4777” comprises 16 bits. These transposed bits are placed as VPN Labels in the above output:

```
VPN Label: 14199
VPN Label: 18295
```

BGP service will use this transposed SID for resolution if BGP NH is not resolved.

Now since the BGP next hop pointing to the loopback address with or without color is not available in any transport RIBs, or there is no Flex Algo routes for this color, service will get resolved over the SRv6 locator in inet6.3 which is populated by ISIS SRv6. This is because the longest prefix match on the SRv6 SID will give us the SRv6 locator route of Egress PE3 in PE1:

```
[edit]
root@PE1# run show route 2001:db8:a3:1:: table inet6.3
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 01:33:03, metric 3
    > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:::
        to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

The BGP Service route is now resolved over the locator and downloaded in L3VPN inet.0:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x7da2c1c) 2001:db8:7:255:7::7
    Krt_inh (0x0)
    Krt_cnh (0x73d210c) Index:621
        Krt_inh (0x7edc544) Index:1048588 PNH: 2001:db8:a3:1:0:7:77:0
            List (0x7da1d44) Index:1048581
                Chain Fully resolved tunnel (0x787d434) Index:558 SRv6
                    Router (0x787ddd4) Index:612 fe80::5603:7ff:fe00:2d9f Session-ID: 344 via ge-0/0/2.0
                Chain Fully resolved tunnel (0x787e314) Index:559 SRv6
                    Router (0x787de44) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
    ...
    ...
    ...
    Indirect next hops: 2
        Protocol next hop: 2001:db8:7:255:7::7
        Indirect next hop: 0x0 - INH Session ID: 0 Weight 0x1
        Protocol next hop: 2001:db8:a3:1:0:7:77:0 Metric: 3
        Composite next hop: 0x73d210c 621 INH Session ID: 374
        Indirect next hop: 0x7edc544 1048588 INH Session ID: 374 Weight 0x2
        Indirect path forwarding next hops: 2
            Next hop type: List
            Next hop: fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0
            Next hop: fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0
            2001:db8:a3:1::/64 Originating RIB: inet6.3
            Metric: 3 Node path count: 1
            Forwarding nexthops: 2
                Next hop type: List
                Next hop: fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0
                Next hop: fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0
```

The first protocol next hop as seen is Unresolved and has an Indirect next hop as 0x0. The second protocol next hop as seen is resolved over locator 2001:db8:a3:1::/64 and originating RIB is inet6.3.

3.4 Reduced and Non-Reduced SRH

There are two modes of Encapsulation in SRH. Insert mode and Encap mode are explained in detail in Chapter 4. In this section we will focus on the Reduced and Non-Reduced SRH. Before we dive into these modes let's briefly discuss Encap and Insert mode.

In an Encap Mode a new IPv6 Header with or without SRH is imposed on the current incoming packet. There can be two scenarios:

- When the actual payload arrives at the ingress and the payload prefix has a SRv6 transport to carry the payload to the Egress PE, the ingress PE will impose an IPv6 Header on the payload with or without a SRH.
- When the packet hits the node where the DA (Destination Address) of the incoming packet is the SRv6 SID configured on the node and has a FIB state, the node will process the current SRH, therefore copy the Next SID in the DA, reduce the SL (Segment-Left) by 1, and then impose the new IPv6 header on top of the current one and mark the next header of SRH as 41 which is IPv6 Encapsulation.

In insert mode, the SRH header is inserted into an existing IPv6 header of the packet and no new IPv6 header is imposed. In the Insert mode the incoming packet may already have an SRH imposed. The DA of the incoming packet is equal to the SRv6 SID programmed in that node. The current SRH is processed and a new SRH is inserted before the current SRH in the existing IPv6 Encap header. The new SRH has the next header marked as 43.

Today, SRv6 TI-LFA supports both Insert and Encap mode. Best effort ISIS SRv6 transport supports only Encap without SRH. SRv6 TE transport supports only Encap mode today.

With that in mind, Reduced and Non-Reduced modes are the two methods of constructing SRH in Encap and Insert mode. Let's explore.

3.4.1 Reduced Mode

In Reduced SRH, the first SID or segment in the SRv6 Segment List is copied to the IPv6 Destination of the outer header and the rest of the SID resides in the SRH.

In SRv6 the primary path will have only one SID and that gets copied to the IPv6 Destination address of the outer IPv6 header. Since there are no SIDs left in the stack, the SRH is not populated. This means for a best effort SRv6 Path there will be *no* SRH imposed in the outgoing packet from ingress. This has the advantage that transit nodes in the network need not be SRv6 nodes and does only IPv6 forwarding over the locator route. When the packet reaches the node where the destination Address has a SRv6 route state in FIB, the outer header is decapsulated and payload is exposed. Details of End, or End.X SID routes, are explained in Chapter 2.

A BGP global/L3 VPN service, with an End.DT4/End.DT6 SID, resolves over ISIS SRv6 having only primary PATHs. As mentioned, in Reduced Mode the only SID in primary PATH is copied to the destination of the outer IPv6 packet. But in this case there's Service SID as well. In Junos, the first SID is always replaced by the service SID to avoid two times decapsulation in the Egress PE avoiding performance and throughput degradation. Thus, in this case, the only SID in ISIS SRv6 Primary Path will be replaced by the Service SRv6 SID and hence that Service SID will be copied to the destination of the outer IPv6 Header. And in this case, as well, we won't have a SRH.

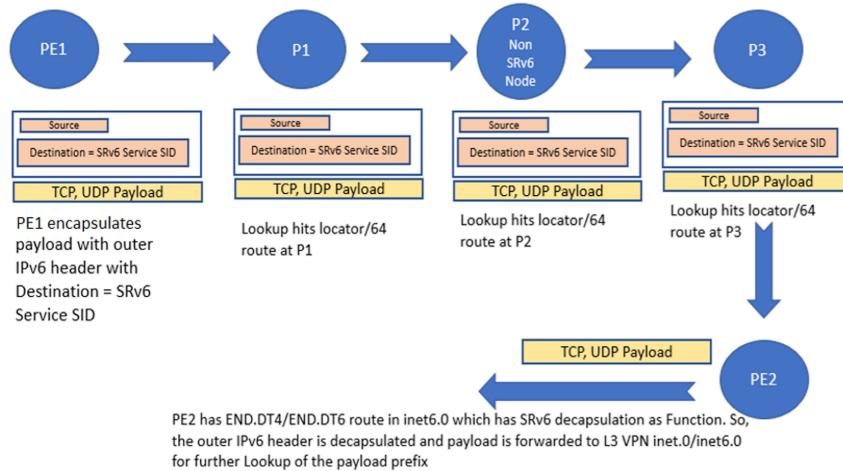


Figure 3.1 Depicting how a Reduced Encap Mode works with ISIS SRv6 Best Effort or Primary Path. SRH Less Packet.

Note, that in Junos, even if an explicit Non-Reduced Mode is configured on the router, ISIS SRv6 Best Effort, or Primary Path will always have Reduced Encap Mode at ingress. The reason behind this is to omit SRH from the data path.

When the Service Resolution involves SRv6 TI-LFA path or SRv6 TE policies and those paths have more than one SID, then there will be an SRH where the first SID goes in IPv6 Destination and the rest of the SIDs are packed in the SRH.

3.4.2 Non-Reduced Mode

By default, Junos follows Reduced Encap Mode for SRv6 unless explicitly configured on the router. In a Reduced Encap Mode you might be unable to preserve the entire SID List. This is a way where the first SID is copied in the outer IPv6 Destination address and also placed in the SRH to hold the SID context in the transport packet.

Since ISIS SRv6 Best Effort or Primary Path always follow Reduced Mode SRH, so non-reduced mode is applicable for TI-LFA path, SRv6 TE policies, and SRv6 Flex Algo TI-LFA Paths.

This is also true for an SRv6 TE policy having a Segment List with one SID. When BGP services resolve over such SRv6 TE policy, the first SID in the Segment List is replaced by SRv6 Service SID and that Service SID will be present in both IPv6 Destination of outer IPv6 header as well in the SRH with Segment Left as 1.

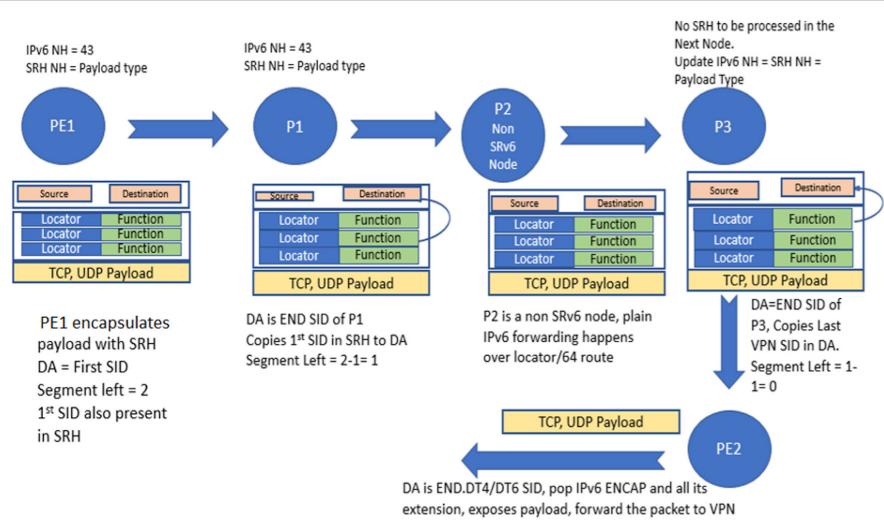


Figure 3.2 NonReduced Encap Mode Where All the SIDs Are Placed in SRH

Chapter 4

Enabling TI-LFA Protection

4.1 Introduction

In its simplest form, Topology-Independent Loop-Free Alternate, or TI-LFA, provides protection against the failure of a single link or node. TI-LFA establishes a fast reroute path over a post-convergence path. More details on TI-LFA can be found in the *Day One* books cited in Key Segment Routing Resources at the beginning of this book. When link protection is configured, a backup FRR path is computed that avoids the failed link. When node protection is configured, a backup FRR path is computed that avoids the failed node.

SRv6 locators may be configured as anycast and as such the same locator can be advertised by multiple routers. It is useful for other routers to know that the advertisement is for an anycast locator. For TILFA computation, Anycast SIDs cannot be picked as anchor points.

4.2 Insert Mode and Encap Mode

The TI-LFA backup paths are represented as a group of SRv6 SIDs inside an SRH. For ingress traffic we encapsulated this SRH in a new IPv6 header (encap mode). But this SRH can be inserted into the data traffic in two different ways at transit, namely *insert mode* or *encap mode*.

When an SRH is inserted as the next header in the original IPv6 packet, the original IPv6 header is changed. In particular, the next header is modified according to the value of the SRH and the IPv6 destination address is replaced with the IPv6 address of the first SID in the segment list while the original IPv6 destination address is carried in the SRH header as the last segment of the list. This is illustrated in Figure 4.1.

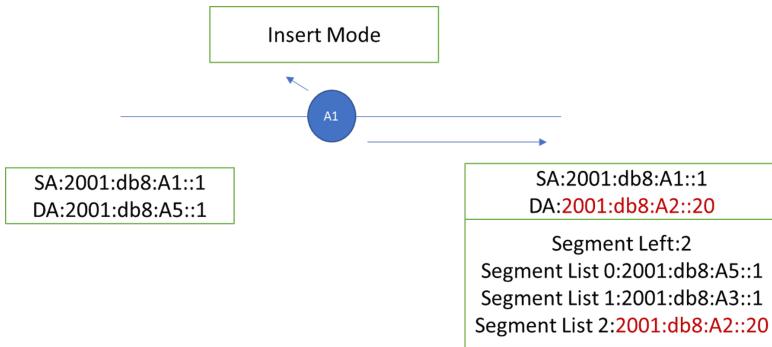


Fig 4.1 Insert Mode

You can see in Fig 4.1 that the incoming packet is coming with a destination address 2001:db8:A5::1. A new SRH is getting inserted whereas the first SID in the SRH 2001:db8:A2::20 becomes the new destination address of the outgoing packet. The original destination address 2001:db8:A5::1 is carried as the last segment of the SRH.

There is also another option for SRv6. With encap mode the original IPv6 packet is transported as the inner packet of an IPv6-in-IPv6 encapsulated packet and travels unmodified in the network. The outer IPv6 packet carries the SRH header with the segments list as shown in Figure 4.2.

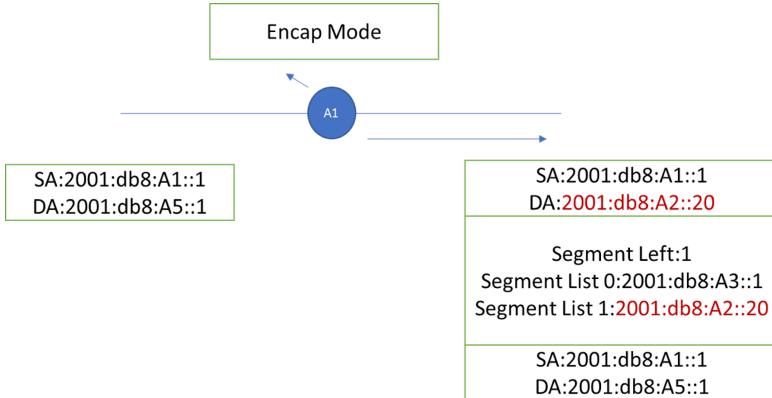


Fig 4.2 Encap Mode

As shown, the incoming packet is coming with a destination address 2001:db8:A5::1. A new IPv6 header with SRH is getting added on top of the incoming packet in this mode. The first SID in the new SRH is set as the destination address of the new outer IPv6 header. The incoming packet as it is going as the inner IPv6 header.

By default, encap mode is used. A knob should be enabled for the insert mode. By default, on Juniper routers, reduced mode is enabled for encap or inserted. With the help of the knob `set routing-options source-packet-routing srv6 no-reduced-srh` the normal mode can be achieved.

The MSD (Maximum Sid Depth) value for SRv6 in the case of the MX is 6 and in case of the ACX it is 12.

4.2.1 The Flavors of SIDs to be Used

The TI-LFA path consists of one or more additional SRv6 SIDs which represent the anchor points in the path.

In encapsulation mode on a transit router, an additional IPv6 header is imposed by the PLR router which has to be removed before the packet reaches the last node. If it is not removed then the last node has to do two decapsulations, which is expensive. Junos implementation mandates the use of USD flavored SID for the last SID in the TI-LFA path so that the TI-LFA encapsulation can be removed before the packet reaches the destination.

The mandatory flavor check applies only to the last SID in the path, other SIDs in the path can be of any flavor or without any flavor.

In insert mode on a transit router, an additional SRH is inserted by the PLR router which has to be removed before the packet reaches the last node. Juniper implementation mandates use of a PSP-flavored SID for the last SID in the TI-LFA path so that the inserted SRH gets removed before the packet reaches the destination. The mandatory flavor check applies only to the last SID in the path, other SIDs in the path can be of any flavor or without any flavor.

4.3 TI-LFA at Headend

Let's look at the service prefix 77.77.77.77 advertised by PE1.

With the existing topology ECMP is at the ingress PE1 towards PE3. To explain the TI-LFA at ingress, the cost of links between P4-PE3 and P4-PE4 is increased to 100 (see Figure 4.3). Similarly, the cost of the link between P2 and PE2 is set to 100:

```
root@P4# set protocols isis interface ge-0/0/3.0 level 2 metric 100
root@P4# set protocols isis interface ge-0/0/4.0 level 2 metric 100
root@P2# set protocols isis interface ge-0/0/4.0 level 2 metric 100
root@PE3# set protocols isis interface ge-0/0/3.0 level 2 metric 100
root@PE4# set protocols isis interface ge-0/0/3.0 level 2 metric 100
root@PE2# set protocols isis interface ge-0/0/3.0 level 2 metric 100
```

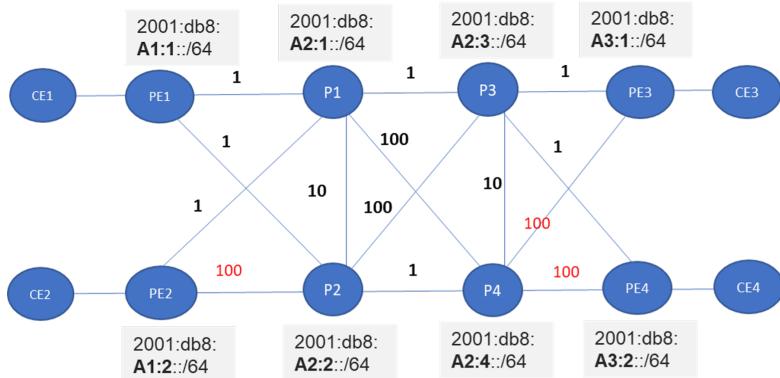


Fig 4.3 TILFATI-LFA at Headend Topology

You can see that the payload prefix has SRv6 SID 2001:db8:a3:1:3777:0:0:0:

```
root@PE1# run show route 77.77.77.77 extensive | match SID
SRv6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

The inet6.3 has a helper locator route 2001:db8:a3:1::/64 installed by ISIS. It has a backup route programmed using TILFA. The protecting interface is ge-0/0/2.0. Let's look at the TI-LFA for protection on the failure of the primary interface. First enable TI-LFA:

```
root@PE1# set protocols isis backup-spf-options use-post-convergence-lfa
root@PE1# set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
root@PE1# set protocols isis backup-spf-options use-source-packet-routing
```

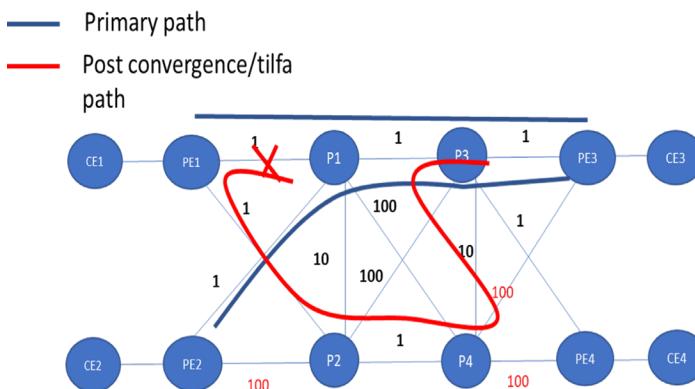


Fig 4.4 TILFA at Headend Post Convergence Path

The primary path as shown in Figure 4.3.2 is PE1-P1-P3-PE3. Once the protection is configured the post convergence path for PE1 to P1 link flap is PE1-P2-P4-P3-PE3. The backup path is via ge-0/0/3.0 and the SRH contains P4 end SID (2001:db8:a2:4:11::) followed by P4-P3 End.X SID (2001:db8:a2:4:265:0:0:0).

Since this is encap mode at headend, the extra anchor SRv6 SIDs are packed in the same SRH and hence any flavor of the SIDs from the anchor nodes are sufficient. The service SID will be the last SID in the SRH.

The service route resolving over the helper route also inherits the same:

```
root@PE1# run show route 2001:db8:a3:1::/64 table inet6.3 extensive
inet6.3: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
  *SRV6-ISIS Preference: 14
    Level: 2
    Next hop type: List, Next hop index: 1048609
    Address: 0x7da3a5c
    Next-hop reference count: 13, key opaque handle: 0x0, non-key opaque handle: 0x0
    Next hop: ELNH Address 0x787f114 weight 0x1, selected
    .....
      SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
      Src: 2001:db8:1:255:1::1 Dest: 2001:db8:a3:1::
      Segment-list[0] 2001:db8:a3:1::
        Next hop type: Router, Next hop index: 743
        Address: 0x787e0e4
        Next-hop reference count: 43, key opaque handle: 0x0, non-key opaque handle: 0x0
        Next hop: fe80::5604:17ff:fe00:b59a via ge-0/0/2.0 weight 0x1
      Next hop: ELNH Address 0x787eaf4 weight 0xf000
    .....
      SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
      Src: 2001:db8:1:255:1::1 Dest: 2001:db8:a3:1::
      Segment-list[0] 2001:db8:a2:4:11::
      Segment-list[1] 2001:db8:a2:4:265::
      Segment-list[2] 2001:db8:a3:1::
        Next hop type: Router, Next hop index: 744
        Address: 0x787d2e4
        Next-hop reference count: 13, key opaque handle: 0x0, non-key opaque handle: 0x0
        Next hop: fe80::5604:17ff:fe00:b49f via ge-0/0/3.0 weight 0xf000
    .....
```

The service route will inherit the same:

```
root@PE1# run show route 2001:db8:a3:1:3777:0:0:0 table inet6.3 extensive
inet6.3: 21 destinations, 21 routes (21 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
  *SRV6-ISIS Preference: 14
    Level: 2
    Next hop type: List, Next hop index: 1048609
    .....
      SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
      Src: 2001:db8:1:255:1::1 Dest: 2001:db8:a3:1::
      Segment-list[0] 2001:db8:a3:1::
        Next hop type: Router, Next hop index: 743
        Address: 0x787e0e4
        Next-hop reference count: 43, key opaque handle: 0x0, non-key opaque handle: 0x0
        Next hop: fe80::5604:17ff:fe00:b59a via ge-0/0/2.0 weight 0x1
    .....
      SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
      Src: 2001:db8:1:255:1::1 Dest: 2001:db8:a3:1::
      Segment-list[0] 2001:db8:a2:4:11::
      Segment-list[1] 2001:db8:a2:4:265::
      Segment-list[2] 2001:db8:a3:1::
        Next hop type: Router, Next hop index: 744
        Address: 0x787d2e4
        Next-hop reference count: 13, key opaque handle: 0x0, non-key opaque handle: 0x0
        Next hop: fe80::5604:17ff:fe00:b49f via ge-0/0/3.0 weight 0xf000
```

In the headend, there will be a single encap even when encap mode is used with TI-LFA. The SRH contains the extra anchor node SRv6 SIDs required to give the configured protection for the primary interface. You can see that the last segment is the same as the primary SRv6 SID; but has two extra SIDs in the SRH for the backup path.

Once this test is over, the metrics are restored to previous values on the respective routers in readiness for the next discussion:

```
root@P4# set protocols isis interface ge-0/0/3.0 level 2 metric 1
root@P4# set protocols isis interface ge-0/0/4.0 level 2 metric 1
root@P2# set protocols isis interface ge-0/0/4.0 level 2 metric 1
root@PE3# set protocols isis interface ge-0/0/3.0 level 2 metric 1
root@PE2# set protocols isis interface ge-0/0/3.0 level 2 metric 1
root@PE2# set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

4.4 TI-LFA at Transit

Junos supports both insert and encap mode for TI-LFA at transit. By default, it's encap mode. The reference topology is shown in Figure 4.5.

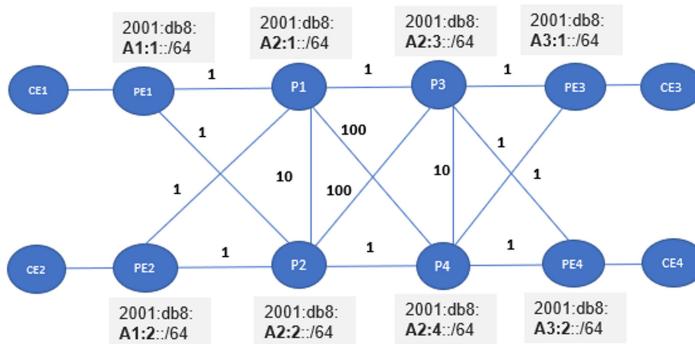


Fig 4.5

TILFATI-LFA at Transit Topology Reference Topology

4.4.1 Insert and Encap for Transit TI-LFA

4.4.1.1 Encap mode

Now let's see transit traffic passing through P1 towards PE3.

We know that SRv6 packets to PE3 hitting P1 have a destination address belonging to an SRv6 SID advertised by PE3. So, at P1, the routing happens based on the locator (longest prefix match).

Let's check the locator at PE3 :

```
[edit]
root@PE3#
source-packet-routing {
    srv6 {
        locator loc1 2001:db8:a3:1::/64;
    }
}
```

Now at P1 let's see how the route is programmed for that PE3 locator:

```
root@P1> show route 2001:db8:a3:1::/64 table inet6.0 extensive
inet6.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
    State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {fe80::5604:dff:fe00:59bd}
IS-IS level 2, SRv6 Loc Sid 0x98645b4
    *IS-IS Preference: 18
        Level: 2
        Next hop type: Router, Next hop index: 610
        Address: 0x7adacc4
        Next-hop reference count: 27, key opaque handle: 0x0
        Next hop: fe80::5604:dff:fe00:59bd via ge-0/0/1.0, selected
        Session Id: 326
        State: <Active Int>
        Age: 2w0d 17:23:47      Metric: 2
        Validation State: unverified
        ORR Generation-ID: 0
        Task: IS-IS
        Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
        AS path: I
        Thread: junos-main
```

You can see that there is no backup path present. Now let's enable TI-LFA and observe Figure 4.6:

```
root@P1# set protocols isis backup-spf-options use-post-convergence-lfa  
root@P1# set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa  
root@P1# set protocols isis backup-spf-options use-source-packet-routing  
          Primary path
```

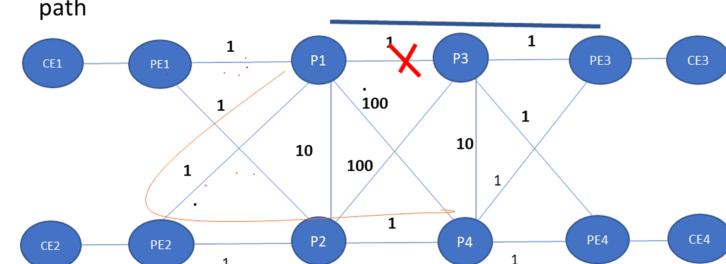


Fig 4.6

TILFA at Transit Post Convergence Path TI-LFA Backup Path

As shown in the figure, the primary interface is P1-P3 and once protection is enabled the TI-LFA backup path is installed which encapsulates the traffic up to P4 and after that it follows the normal SPF path towards the destination address:

```
[edit]
root@P1# run show route 2001:db8:a3:1::/64 table inet6.0 extensive
inet6.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
    State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {fe80::5604:dff:fe00:59bd}
IS-IS level 2, SRv6 Loc Sid 0x98645b4
    *IS-IS Preference: 18
        Level: 2
        Next hop type: List, Next hop index: 0
        Address: 0x79ebd60
        Next-hop reference count: 1, key opaque handle: 0x0
        Next hop: ELNH Address 0x7ade5b8 weight 0x1, selected
        .....
        Next hop: fe80::5604:dff:fe00:59bd via ge-0/0/1.0 weight 0x1
        Next hop: ELNH Address 0x7ada598 weight 0xf000
        .....
        SRV6-Tunnel: Reduced-SRH Encap-mode
        Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1::
        Segment-list[0] 2001:db8:a2:4:11:: >> (USD flavor sid)
            Next hop type: Router, Next hop index: 0
            Address: 0x7adf0b0
            Next-hop reference count: 8, key opaque handle: 0x0
            Next hop: fe80::5604:dff:fe00:552d via ge-0/0/4.0 weight 0xf000
        .....
```

The anchor node is P4 and a USD-flavored SID is chosen since the mode is encap:

```
root@P4# show | display set | match 2001:db8:a2:4:1:1:::
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:4:11:0::0 flavor usd
```

4.4.1.2 Insert Mode

Let's look at insert mode. We'll keep the configurations same on all devices as in the previous section but enable insert mode at P1 with this configuration:

```
root@P1# set routing-options source-packet-routing srv6 transit-srh-insert
root@P1# run show route 2001:db8:a3:1::/64 table inet6.0 extensive
inet6.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
    State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {fe80::5604:dff:fe00:59bd}
IS-IS level 2, SRv6 Loc Sid 0x98645b4
    *IS-IS Preference: 18
        Level: 2
        Next hop type: List, Next hop index: 0
        Address: 0x79ebccc
        Next-hop reference count: 1, key opaque handle: 0x0
        Next hop: ELNH Address 0x7ade5b8 weight 0x1, selected
            Next hop type: Router, Next hop index: 610
            Address: 0x7ade5b8
```

```

Next-hop reference count: 24, key opaque handle: 0x0
Next hop: fe80::5604:dff:fe00:59bd via ge-0/0/1.0 weight 0x1
Next hop: ELNH Address 0x7ada160 weight 0xf000
    Next hop type: Chain, Next hop index: 0
    Address: 0x7ada160
    Next-hop reference count: 1, key opaque handle: 0x0
    Next hop: via Chain Tunnel Composite, SRv6
    Next hop: ELNH Address 0x7adf0b0
    SRV6-Tunnel: Reduced-SRH Insert-mode
        Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1::
        Segment-list[0] 2001:db8:a2:4:12:: >> PSP flavor SID
            Next hop type: Router, Next hop index: 0
            Address: 0x7adf0b0
            Next-hop reference count: 8, key opaque handle: 0x0
            Next hop: fe80::5604:dff:fe00:552d via ge-0/0/4.0 weight 0xf000
State: <Active Int>
Age: 46      Metric: 2
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
AS path: I
Thread: junos-main

```

The anchor node is P4 and a PSP-flavored SID is chosen since the mode is insert:

```
root@P4# show | display set | match 2001:db8:a2:4:12::
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:4:1:2::0 flavor psp
```

4.5 Anycast SID

An SRv6 anycast locator is a type of locator that identifies a set of nodes (End SIDs). SRv6 anycast locators and their associated End SIDs may be provisioned on multiple nodes in a topology. When an anycast locator is configured on multiple nodes, the configuration of SIDs derived out of the anycast locator must match on all the nodes. This is because when a packet is routed based on the anycast locator, it may hit any of these nodes. If some nodes have a SID configured and others don't have the exact same SID, then traffic will be dropped on the other node when traffic arrives with the SID that is not configured.

The following behaviors apply to anycast locators:

- SIDs allocated from anycast locators will not be used in constructing TI-LFA backup paths or microloop avoidance primary paths (described in Chapter 9). TI-LFA backup and microloop avoidance paths for an anycast locator prefix may terminate on any node advertising that locator, which may be different from the node terminating the original primary path.
- SRv6 anycast locators may have non-zero algorithm (flexible algorithm) values.

Now let's demonstrate an example of TI-LFA not picking anycast SIDs. At P4, which is our anchor node for our previous examples, we disabled the End-SID which was configured, and then configured another anycast locator loc2. We configured a SID for the anycast locator loc2:

```
set routing-options source-packet-routing srv6] locator loc2 2001:db8:aa:0::/64;
set routing-options source-packet-routing srv6 locator loc2 anycast
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:aa:0:1::0 flavor psp
deactivate protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:4:12::0
```

You can see that the anycast SRv6 SIDs are not picked up for TI-LFA backup and that no backup path is programmed due to the non-availability of a non-anycast SID at P4:

```
root@P1# run show route 2001:db8:a3:1::/64 table inet6.0 extensive
inet6.0: 49 destinations, 49 routes (49 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[IS-IS/18] 00:23:50, metric 2
    > to fe80::5604:dff:fe00:59bd via ge-0/0/1.0
```

The post convergence path has P4 as one of the anchor nodes. Since there is no usable SID for programming backup, the backup is not programmed by Junos.

Chapter 5

SRv6 to MPLS Interworking

5.1 Introduction

When a network introduces new technology such as SRv6, it should always be able to inter-operate with existing technology. Since most WAN networks deploy MPLS today, this chapter covers SRv6 to MPLS service interworking where one side's service endpoint speaks SRv6 and other side speaks MPLS. The chapter begins by creating IPv6 BGP sessions with the book's aim is to introduce IPv6 and SRv6 into the network and slowly remove IPv4 addresses from the network when all the nodes are upgraded. In previous chapters BGP sessions have been running on IPv4 addresses. This chapter begins with creating eBGP and iBGP sessions based on IPv6 addresses and later discusses how to interoperate SRv6 service advertisement with MPLS's service advertisement over these IPv6 BGP sessions.

5.2 Create iBGP and eBGP v6 Sessions

5.2.1 Enabling IPv6 eBGP v6 Sessions Between CE and PE

Chapter 1 enabled IPv6 addresses and IPv6 routing in the transport network. In this chapter, let's configure IPv6 eBGP sessions between each CE and PE. We'll exchange IPv6 as well as IPv4 service prefixes over these IPv6 BGP sessions. You enable `inet unicast` and `inet6 unicast` family on the IPv6 BGP sessions in order to allow both v4 and v6 service prefixes to be exchanged over the session. Then create a new group `eBGPv6` on PE1 under `routing-instance` and add CE1 with an IPv6 neighbor address.

Here are the configs for setting up eBGP v6 sessions between PE and CE.

PE1:

```
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 neighbor 2001:db8:11::1
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet unicast local-ipv4-address
1.11.1.2
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet6 unicast
CE1:

set logical-systems CE1 protocols bgp group eBGPv6 type external
set logical-systems CE1 protocols bgp group eBGPv6 export BGP_export
set logical-systems CE1 protocols bgp group eBGPv6 peer-as 64512
set logical-systems CE1 protocols bgp group eBGPv6 neighbor 2001:db8:11::2
set logical-systems CE1 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 1.11.1.1
set logical-systems CE1 protocols bgp group eBGPv6 family inet6 unicast
```

Similar configs are done on all PEs. Now let's verify the BGP summary on each PE and CE.

CE1:

```
root@PE1# run show bgp summary group eBGPv6 logical-system CE1
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table      Tot Paths  Act Paths Suppressed History Damp State Pending
inet.0
          3        3        0        0        0        0
inet6.0
          8        3        0        0        0        0
Peer          AS  InPkt    OutPkt  OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/
Damped...
2001:db8:11::2  64512     28      26      0      0  11:10 Establ
          inet.0: 0/3/3/0
inet6.0: 3/4/4/0
```

PE1:

```
root@PE1# run show bgp summary group eBGPv6
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 3 Peers: 5 Down peers: 0
Table      Tot Paths  Act Paths Suppressed History Damp State Pending
inet.0
          0        0        0        0        0        0
bgp.l3vpn.0
          6        6        0        0        0        0
bgp.l3vpn-inet6.0
          8        8        0        0        0        0
Peer          AS  InPkt    OutPkt  OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/
Damped...
2001:db8:11::1  100     27      25      0      0  10:43 Establ
          L3VPN_PE1_CE1.inet.0: 0/2/2/0
L3VPN_PE1_CE1.inet6.0: 0/3/3/0
```

This same verification step can be performed at each PE and each CE.

5.2.2 Enabling IPv6 iBGP Sessions Among PEs:

Now it's time to configure the iBGP v6 sessions among PEs. First create a new group, ibgpv6, PE1 and add PE2, PE3, and PE4 with IPv6 neighbor addresses. Here are the config changes in the PEs.

PE1:

```
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6-vpn unicast
set protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
set protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
```

New iBGP sessions get created and moved to Established state. You can verify the sessions using show bgp summary:

```
PE1>show bgp summary
...
2001:db8:2:255:2::2      64512      9      8      0      0      17 Establ  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/2/2/0
  bgp.l3vpn-inet6.0: 0/2/2/0
  L3VPN_PE1_CE1.inet.0: 0/2/2/0
  L3VPN_PE1_CE1.inet6.0: 0/2/2/0
2001:db8:7:255:7::7      64512      9      8      0      0      12 Establ  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/2/2/0
  bgp.l3vpn-inet6.0: 0/3/3/0
  L3VPN_PE1_CE1.inet.0: 0/2/2/0
  L3VPN_PE1_CE1.inet6.0: 0/3/3/0
2001:db8:8:255:8::8      64512      9      8      0      0      8 Establ >>> PE4
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/2/2/0
  bgp.l3vpn-inet6.0: 0/3/3/0
  L3VPN_PE1_CE1.inet.0: 0/2/2/0
  L3VPN_PE1_CE1.inet6.0: 0/3/3/0
```

5.2 Setup SRv6 Over BGP v6 Sessions

Once complete you can now enable SRv6 advertisement and acceptance on iBGP v6 sessions among the PEs.

PE1:

```
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
```

You can see that PE1's exported route over these v6 iBGP sessions is advertised with PE1's L3-VPN End.DT4 and DT6 SIDs:

```
L3VPN_PE4_CE4.inet.0: 9 destinations, 18 routes (6 active, 0 holddown, 6 hidden)
 1.11.1.0/24 (2 entries, 1 announced)
   Import Accepted
   Route Distinguisher: 1.1.1.1:1
   VPN Label: 3
   Nexthop: 2001:db8:a1:1:3111:0:0:0
   Localpref: 100
   AS path: I
   Communities: target:64512:1
   SRv6 SID: 2001:db8:a1:1:3111:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0

L3VPN_PE4_CE4.inet6.0: 12 destinations, 28 routes (9 active, 0 holddown, 12 hidden)

 2001:db8:11::/48 (2 entries, 1 announced)
   Import Accepted
   Route Distinguisher: 1.1.1.1:1
   VPN Label: 3
   Nexthop: 2001:db8:a1:1:4111:0:0:0
   Localpref: 100
   AS path: I
   Communities: target:64512:1
   SRv6 SID: 2001:db8:a1:1:4111:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

5.2.1 BGP VPN Service Protocol Next Hop Change

Chapter 3 discussed how one can change their next hop in egress by configuring `next-hop` in `policy-options` and point it to the SRv6 locator of Egress PE. But there can be non-Juniper Nodes in the network which may not support changing BGP next hop through policy configurations in the Egress router.

There are two alternate mechanisms to address this interoperability.

5.2.1.1 Service Mapping in Ingress prior 22.3 JUNOS Release

Prior to its 22.3 release, Junos supports a mechanism where the Ingress PEs can be configured with a static IPv6 route prefix, the next hop same as BGP next hop in `inet6.3` table with a `next-hop` pointing to SRv6 Locator. In this way you can achieve inter-op with other vendor's routers.

Here are the configurations on the egress PE to set next hop to self:

```
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop self
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then accept
```

Now in the ingress PE, PE1 in our example, if you check the BGP routes advertised from PE3 (the egress PE in our example), you will see the routes as `hidden` as the BGP next hop is not resolvable via the SRv6 tunnel in `inet6.3`:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive hidden
L3VPN_PE1_CE1.inet.0: 9 destinations, 18 routes (5 active, 0 holddown, 8 hidden)
  7.77.1.0/24 (2 entries, 0 announced)
    Import Accepted
    Route Distinguisher: 7.7.7.7:1
    VPN Label: 3
    Nexthop: 2001:db8:7:255:7::7 >> BGP NextHop as Loopback IP of PE3
    Localpref: 100
    AS path: I
    Communities: target:64512:1
      SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
...
...
Now configure the static route in the Ingress PE in inet6.3 table with the same prefix as
BGP next hop:
```

```
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7 next-hop 2001:db8:a3:1::0 resolve
```

If you closely look at the next hop in the route, it is the locator of the egress PE3. We now have a route in inet6.3 pointing to the SRv6 Tunnel derived from the locator route present in inet6.3:

```
[edit]
root@PE1# run show route 2001:db8:7:255:7::7 table inet6.3
inet6.3: 23 destinations, 24 routes (23 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:7:255:7::7/128
  *[Static/5] 00:07:43, metric2 3
    > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

As you can see, the protocol of this route is `Static` and it has a next hop as the SRv6 tunnel derived from PE3's locator route. Now our service route becomes resolvable and downloaded in L3 VPN's inet.0 IPv4 services respectively:

```
[edit]
root@PE1# run show route 7.77.1.0
L3VPN_PE1_CE1.inet.0: 9 destinations, 18 routes (7 active, 0 holddown, 6 hidden)
+ = Active Route, - = Last Active, * = Both
  7.77.1.0/24      *[BGP/170] 00:10:34, localpref 100, from 2001:db8:7:255:7::7
    AS path: I, validation-state: unverified
    > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

```
[edit]
root@PE1# run show route 2001:db8:77::0
L3VPN_PE1_CE1.inet6.0: 12 destinations, 15 routes (10 active, 0 holddown, 3 hidden)
+ = Active Route, - = Last Active, * = Both
  2001:db8:77::/48  *[BGP/170] 00:13:04, localpref 100, from 2001:db8:7:255:7::7
    AS path: I, validation-state: unverified
    > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

In the same way these IPv6 services can get resolved and downloaded in L3-VPN `inet6.0`. This alternate mechanism works for BGP services without color community. With the presence of the color community in BGP service you'll need SRv6 Flex Algo route, which is explored in Chapter 7.

5.2.1.2 Service Mapping in Ingress from 22.3 JUNOS Release

With new service resolution schemes and usage of transport class RIBs (detailed description provided in Chapter 10) in Junos releases 22.3, you don't need any static route configurations or next hop changing policies for services advertised with or without color community.

If the BGP next hop is not resolvable with or without color, the fallback or default scheme always tells it to lookup the locator route from inet6.3 and resolve on that. This mechanism saves the operators from having to configure a number of static routes or next hop change policies.

BGP services advertised without COLOR community

Let's take a look at PE1's advertisements of PE3 for IPv4 services:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
  Import Accepted MultiNexthop RecvNextHopIgnored
  Route Distinguisher: 7.7.7.7:1
  VPN Label: 14199
  Nexthop: 2001:db8:7:255:7::7 >> BGP NextHop as Loopback address of PE3
  Localpref: 100
  AS path: I
  Communities: target:64512:1
  SRv6 SID: 2001:db8:a3:1:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 T0: 64
```

Now without any Static route configuration it looks up in the inet6.3 table to resolve the SRv6 locator route to which the SRv6 SID `2001:db8:a3:1::` belongs to. For routes without color community the default scheme is to look up on the SRv6 locator route that the SRv6 SID belongs to.

The locator for the SRv6 SID `2001:db8:a3:1::` in inet6.3:

```
[edit]
root@PE1# run show route 2001:db8:a3:1:: table inet6.3
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 22:02:44, metric 3
  > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
    to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

Note that from the 22.3 release onwards, by default Junos does transposition for service SRv6 SIDs by moving the function bits (maximum of 24 bits can be transposed) onto the MPLS label. More details on transposition can be found in Chapter 10. Here is the service route `7.77.1.0` L3 VPN `inet.0`:

```
[edit]
root@PE1# run show route 7.77.1.0
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
7.77.1.0/24      *[BGP/170] 12:24:23, localpref 100, from 2001:db8:7:255:7::7
                  AS path: I, validation-state: unverified
                  to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
> to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

BGP services advertised with color community:

If BGP services are advertised with color community without any next hop manipulation in egress then it looks this:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
  Import Accepted MultiNexthop
  Route Distinguisher: 7.7.7.7:1
  VPN Label: 14199
  Nexthop: 2001:db8:7:255:7::7
  Localpref: 100
  AS path: I
  Communities: target:64512:1 color:0:3 >>> Color Community
    SRv6 SID: 2001:db8:a3:1:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64
```

We advertised a color community 0:3 with the service prefix 7.77.1.0. As the network currently does not have any type of traffic-engineered tunnels, the route falls back onto best effort SRv6 tunnels.

In this case the fallback scheme is to look up the SRv6 locator route to which the SRv6 SID belongs. In this case the SRv6 SID 2001:db8:a3:1:: belongs to Locator 2001:db8:a3:1::/64 advertised from PE3 and installed in inet6.3 by ISIS SRv6. Here's the locator for the SRv6 SID 2001:db8:a3:1:: in inet6.3:

```
[edit]
root@PE1# run show route 2001:db8:a3:1:: table inet6.3
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 22:02:44, metric 3
  > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
    to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

The service route 7.77.1.0 L3 VPN inet.0:

```
[edit]
root@PE1# run show route 7.77.1.0
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
7.77.1.0/24      *[BGP/170] 12:24:23, localpref 100, from 2001:db8:7:255:7::7
                  AS path: I, validation-state: unverified
                  to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
> to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

Chapter 10 discusses these resolution schemes along with SRv6 TE colored and uncolored transport.

5.2.2 Global inet/inet6 Family Configuration in BGP for Global Internet Traffic

Before diving into SRv6 to MPLS inter-networking let's take a quick look at how global internet traffic can be transported through SRv6.

Until now you have seen many examples of L3 VPN services over SRv6. It is also possible to configure global services to carry internet traffic over SRv6 Services. In this section we will review in detail.

For Internet traffic or global inet/inet6 traffic over SRv6 you have to enhance your iBGP peer to have SRv6 capability and configure SRv6 service SIDs under iBGP.

For the sake of simplicity, let's take PE1 as our ingress and PE3 as our egress and enhance iBGP sessions between these two PEs and validate traffic over the SRv6 data plane. It's an exercise left to the reader to enhance the other iBGP peers and verify internet traffic over SRv6.

To achieve this you need one device behind PE1 and PE3 and they *must* have eBGP connections to PE1 and PE2 in global instances.

You don't have to do many configuration changes in the existing topology. To make it simple let's add CE12 behind PE1 and CE34 behind PE3. In this way only additional configurations are imposed and the rest of our configuration guide remains untouched.

For a quick review at the topology see Figure 5.1.

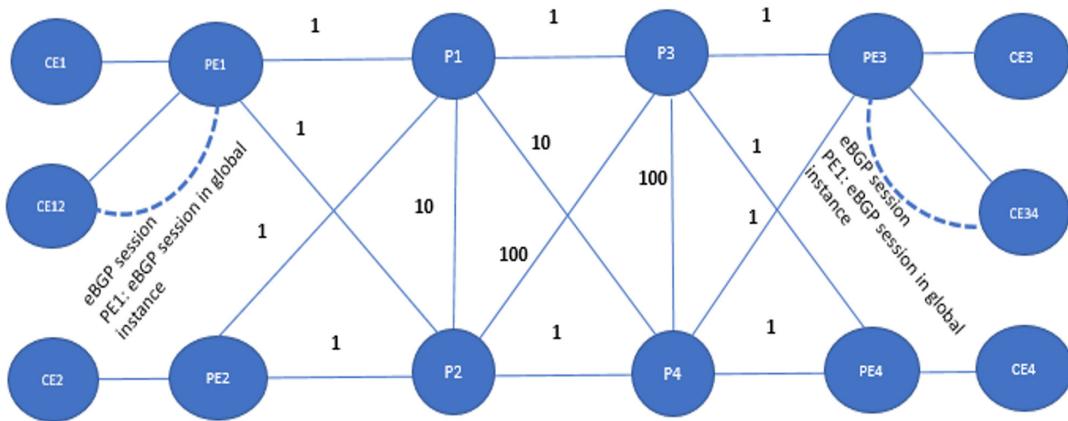


Figure 5.1 Network topology for global inet and inet6 services

Configurations in CE12, PE1, CE34, PE3:

On CE12 let's configure the interfaces:

```
set interfaces ge-0/0/0 unit 0 description ConnectedToPE1
set interfaces ge-0/0/0 unit 0 family inet address 12.1.1.1/24
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:12:1::1/64
set interfaces lo0 unit 0 family inet address 12.12.12.12/32
set interfaces lo0 unit 0 family inet6 address 2001:db8:12:255:12::12/128
```

These policy options export the routes to eBGP peers:

```
set policy-options policy-statement BGP_export term 0 from protocol direct
set policy-options policy-statement BGP_export term 0 from interface lo0.0
set policy-options policy-statement BGP_export term 0 then accept
```

Here are the configurations of eBGP with PE1. You can also see that the above policy is exported in eBGP:

```
set protocols bgp group eBGP_Global_v6 type external
set protocols bgp group eBGP_Global_v6 family inet unicast local-ipv4-address 12.1.1.1
set protocols bgp group eBGP_Global_v6 family inet6 unicast
set protocols bgp group eBGP_Global_v6 export BGP_export
set protocols bgp group eBGP_Global_v6 peer-as 64512
set protocols bgp group eBGP_Global_v6 neighbor 2001:db8:12:1::2
```

These routing-options configurations point to default routes towards PE1:

```
set routing-options rib inet.0 static route 0.0.0.0/0 next-hop 12.1.1.2
set routing-options rib inet6.0 static route ::/0 next-hop 2001:db8:12:1::2
```

Configuring AS for this peer and its IPv6 router ID:

```
set routing-options autonomous-system 101
set routing-options ipv6-router-id 2001:db8:12:255:12::12
```

At PE1, configure the interface towards CE12:

```
set interfaces ge-0/0/2 unit 0 description ConnectedToCE12
set interfaces ge-0/0/2 unit 0 family inet address 12.1.1.2/24
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:12:1::2/64
```

Configuring eBGP with CE12:

```
set protocols bgp group eBGP_Global_v6 type external
set protocols bgp group eBGP_Global_v6 family inet unicast local-ipv4-address 12.1.1.2
set protocols bgp group eBGP_Global_v6 family inet6 unicast
set protocols bgp group eBGP_Global_v6 peer-as 101
set protocols bgp group eBGP_Global_v6 neighbor 2001:db8:12:1::1
```

If you look closely, the config is the same as our L3VPN routing-instance in that it is a global instance and doesn't have any VPN context. CE34 and PE3 will be similarly configured. The full configuration can be found in the Appendix 5.2.2.

Configurations on both PE1 and PE3:

```
set protocols bgp group ibgpv6 family inet unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6 unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6 unicast advertise-srv6-service
```

These configurations are the same as the VPN except for under family inet and inet6. It enables iBGP to accept and allow SRv6 service advertisements.

Now configuring End.DT4 and End.DT6 SIDs in PE1 and PE3.

At PE1:

```
set protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid 2001:db8:a1:1:511:0:0:0
set protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid 2001:db8:a1:1:611:0:0:0
```

At PE3:

```
set protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid 2001:db8:a3:1:571:0:0:0
set protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid 2001:db8:a3:1:671:0:0:0
```

Let's do the SRv6 Service SID now, using PE3's End DT4 and DT6 SID as an example:
`end-dt4-sid 2001:db8:a3:1:571:0:0:0` .

The first 64 bits represent the locator of the SID. It belongs to locator loc1 on PE3.
`2001:db8:a3:1` is the locator part.

The next 16 bits represent the function of this SID. 571 is the function of this SID. At egress PE3 it represents the behavior of the SID in the data plane.

The last 48 bits represent the argument which is `0` for us.

Okay and now: DT6 SID: `2001:db8:a3:1:671:0:0:0` .

The first 64 bits represent the locator of the SID. It belongs to locator loc1 on PE3.
`2001:db8:a3:1` is the Locator part.

The next 16 bits represent the function of this SID. 671 is the function of this SID. At Egress PE3 it represents the behavior of the SID in the data plane.

The last 48 bits represent the argument which is `0` for us.

The state of the global End.DT4 and End.DT6 SRv6 SID on PE3 can be verified using the following commands. PE3 will download these two routes in inet6.0:

```
[edit]
root@PE3# run show route 2001:db8:a3:1:571:0:0:0
inet6.0: 45 destinations, 45 routes (45 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1:571::/128
    *[BGP/170] 00:09:22
        AS path: I, validation-state: unverified
        to table inet.0
```

As you can see the route programmed for DT4 SID at the egress (originator of DT4 SID) indicates that the packet outer header should be decapsulated when SL=0 and further route lookup of payload destination address should be done in the inet.0 table:

```
[edit]
root@PE3# run show route 2001:db8:a3:1:671::0:0:0
inet6.0: 45 destinations, 45 routes (45 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1:671::/128
    *[BGP/170] 00:09:19
        AS path: I, validation-state: unverified
        to table inet6.0
```

Here you can see the DT6 SID indicates that the packet outer header should be decapsulated when SL=0 and further route lookup of payload prefix destination address should be done in the inet.0 table.

Once these SIDs are configured in global BGP, at PE3, you need an export policy to export eBGP global routes learned from CE34 through BGP to its peer PE1 and set that export policy under the iBGP configuration. The same can be done on PE1 as well, so that the reverse path is also SRv6:

```
set policy-options policy-statement global_inet_ibgp_export_policy term 0 from protocol bgp
set policy-options policy-statement global_inet_ibgp_export_policy term 0 then accept
    set protocols bgp group ibgpv6 export global_inet_ibgp_export_policy
```

This is all you need from a configuration point of view, but there are two things more: add a filter with protocol `bgp` that will only export BGP routes, and, add this policy as export under iBGP. (You'll notice that the next hop is still not set. Those next hop changes are in the next subsection.)

In the ingress PE the resolution schemes can again be divided into two parts and their sub parts. These resolution schemes discussed earlier also apply for global BGP services.

Now let's go through the services advertised without the color community in BGP and see how resolution works prior to 22.3 Junos releases and then from 22.3 JUNOS releases.

5.2.2.1 Global services over SRv6 with resolution schemes prior to JUNOS 22.3 Release:

Prior to Junos 22.3 release, you needed to set the BGP next hop to the locator route in the iBGP export policy in the egress PE3.

In the egress PE3:

```
set policy-options policy-statement global_inet_ibgp_export_policy term 0 then next-hop
2001:db8:a3:1::0
```

This is setting the global iBGP export policy to have a `next-hop` as the SRv6 locator configured on that device PE3. Let's take the IPv4 loopback address of CE34 and verify its different states.


```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!! --- 34.34.34.34 ping statistics ---
766 packets transmitted, 765 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.156/5.152/36.983/1.565 ms
```

One thing to notice is that if the egress PE is advertising self as next hop for these global service routes then you have to configure a Static route in Ingress, the same way you saw in section 5.2.1. This is not described here as to simplify configurations that avoid static route configurations or next hop configuration: use Junos release 22.3 or later versions.

5.2.2.2 Global services over SRv6 with resolution schemes from JUNOS 22.3 Release:

With the locator as next hop, there's little difference between the previous section and this. So let's look at a use case where the egress PE is configured with `self` as BGP `next-hop`.

For description purposes, let's change the next hop changing policy on egress PE3 and make it point to a `next-hop` as `self`:

```
set policy-options policy-statement global_inet_ibgp_export_policy term 0 then next-hop self
```

All other configurations are the same as described in previous sections. The advantage with this approach is that inter-op with other vendors is easily achieved. Here's the CE34 loopback route advertisement from PE3 to PE1:

```
root@PE3# run show route advertising-protocol bgp 2001:db8:1:255:1::1 extensive
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
* 34.34.34.34/32 (1 entry, 1 announced)
  BGP group ibgpv6 type Internal
    Nexthop: Self >>> Changed to SELF/Loopback address of PE3
    Flags: Nexthop Change
    Localpref: 100
    AS path: [64512] 101 I
      SRv6 SID: 2001:db8:a3:1:571:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 0 TO: 0
```

The SRv6 SID is the one configured in bgp in the global instance in PE3. Let's check the same prefix on PE1:

```
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
* 34.34.34.34/32 (1 entry, 1 announced)
  Accepted MultiNexthop RecvNextHopIgnored
  Nexthop: 2001:db8:7:255:7::7 >>> PE3's Loopback address
  Localpref: 100
  AS path: 101 I
    SRv6 SID: 2001:db8:a3:1:571:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 0 TO: 0
```

So SRv6 End.DT4 SID properly received on Ingress. Resolution will be on the Locator `2001:db8:a3:1::` which is downloaded in inet6.3 by ISIS SRv6.

Not to the locator route in inet6.3 in PE1. This is achieved without any extra Static route configuration in ingress (the Static route would have been required if it were a pre- 22.3 Junos release).

Let's check the service route now in inet.0:

```
[edit]
root@PE1# run show route 34.34.34.34
inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
34.34.34.34/32    *[BGP/170] 00:10:20, localpref 100, from 2001:db8:7:255:7::7
                  AS path: 101 I, validation-state: unverified
                    > to fe80::5603:7ff:fe00:609e via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
                      to fe80::5603:7ff:fe00:2418 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

As you can see our BGP route has been resolved over the SRv6 Locator from inet.0.

Let's validate this traffic over this service prefix from CE12 to CE34:

```
root@CE12# run ping 34.34.34 rapid count 1000000 interval 0.1 source 12.12.12.12
PING 34.34.34 (34.34.34.34): 56 data bytes
!!!!!!^C
--- 34.34.34 ping statistics ---
766 packets transmitted, 765 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.156/5.152/36.983/1.565 ms
```

5.3 SRv6 to MPLS Interoperability

During the network migration to SRv6, it's possible that all devices in the network cannot be upgraded to enable SRv6. In such cases, SRv6 services should be able to interoperate with these MPLS devices. This section discusses how to configure SRv6 services to interwork with MPLS services (see Figure 5.2). The lab configurations and outputs in this section are taken from a pre- 22.3 release.

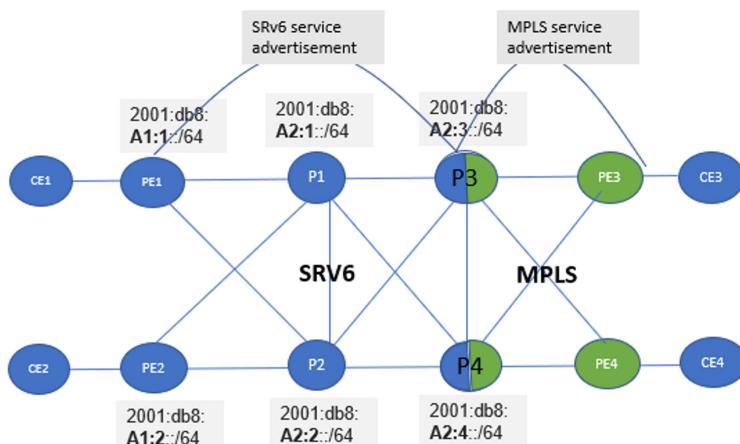


Figure 5.2

SRv6 to MPLS Service Interworking

In Figure 5.2, PE3 and PE4 are SRv6-*incapable* and run MPLS services while PE1 and PE2 are SRv6-*capable* and run SRv6 services. P1, P2, P3, P4 are SRv6 capable and P3 and P4 act as SRv6-to-MPLS gateway devices.

Let's take the L3VPN service advertisements from our lab exercise from Chapter 1 and 3 and see how to make it interoperate. In summary, the L3VPN prefix will be advertised from PE3 to P3 with an MPLS label. On P3, the L3VPN routing-instance will be configured and the L3VPN service prefix will be downloaded into the VPN table. From P3, the L3VPN service will be advertised to PE1 with a SRV6 VPN SID. The packet forwarding happens by PE1 imposing an SRv6 header, On P3 the SRv6 header gets removed and the packet is looked up in the VPN table. The VPN table lookup results in the MPLS header being imposed and the packet forwarded to PE3. PE3 removes the MPLS header and sends the packet to the appropriate CE.

5.3.1 Preparing Network for SRv6 to MPLS Interoperability

Let's prepare the network for SRv6 to MPLS interoperability. We have created two sets of BGP sessions, one based on IPv4 address and another based on IPv6 address. In order to simplify the contents of this section, let's use IPv6 BGP sessions and deactivate the IPv4 BGP sessions. On PE1 and PE2 let's deactivate the BGP sessions to PE3 and PE4 and create BGP sessions to P3 and P4. Similarly, on PE3 and PE4, let's deactivate BGP sessions to PE1 and PE2 and create BGP sessions to P3 and P4:

```
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
deactivate protocols bgp group ibgp
```

On PE3 and PE4, let's deactivate all SRv6 configurations:

```
deactivate routing-options source-packet-routing srv6
deactivate protocols bgp source-packet-routing srv6
deactivate routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6
deactivate protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
deactivate protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
deactivate protocols isis source-packet-routing srv6
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
deactivate protocols bgp group ibgp
```

For detailed configuration on each device please refer to the Appendices.

5.3.2 Setting Up BGP Sessions

Let's configure BGP sessions on PE3 and PE4 towards P3 and P4. These BGP sessions will advertise MPLS services and will use MPLS transport as the underlay:

```
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
```

Similarly, configure BGP sessions on P3 and P4 towards PE3 and PE4:

```
set protocols bgp group internal-peers-mpls type internal
set protocols bgp group internal-peers-mpls family inet-vpn unicast
set protocols bgp group internal-peers-mpls family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers-mpls family inet6-vpn unicast
set protocols bgp group internal-peers-mpls cluster 2.2.2.2
set protocols bgp group internal-peers-mpls peer-as 64512
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 local-address
2001:db8:5:255:5::5
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 local-address
2001:db8:5:255:5::5
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls export exportstatic2
```

P3 and P4 are set up as route reflectors as they need to reflect the routes from the IBGP session towards PE3 to IBGP session towards PE1.

Let's set up the IBGP sessions from PE1 and PE2 towards P3 and P4 and also in the reverse direction. These sessions will be SRv6-enabled and advertise and accept SRv6 SIDs:

```
set routing-options autonomous-system 64512
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers family inet6 unicast
set protocols bgp group internal-peers family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers cluster 2.2.2.2
set protocols bgp group internal-peers peer-as 64512
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 local-address 2001:db8:5:255:5::5
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 cluster 2.2.2.2
set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 local-address 2001:db8:5:255:5::5
set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 cluster 2.2.2.2
set protocols bgp group internal-peers export exportstatic1

set protocols bgp group internal-peers family inet-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet-vpn unicast accept-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast accept-srv6-service
```

Similarly let's configure BGP sessions on PE1:

```
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
```

Verify the BGP sessions get established:

```
regress@P3# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
...
2001:db8:1:255:1::1      64512      1272      1283      0      0  9:35:22 Establ
  inet6.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  bgp.l3vpn-inet6.0: 2/2/2/0
```

```

L3VPN.inet.0: 2/2/2/0
L3VPN.inet6.0: 2/2/2/0
2001:db8:2:255:2::2      64512     1272      1283      0      0 9:35:28 Establ
inet6.0: 0/0/0/0
bgp.l3vpn.0: 2/2/2/0
bgp.l3vpn-inet6.0: 2/2/2/0
L3VPN.inet.0: 2/2/2/0
L3VPN.inet6.0: 2/2/2/0
2001:db8:7:255:7::7      64512     1270      1294      0      0 9:35:28 Establ
bgp.l3vpn.0: 2/2/2/0
bgp.l3vpn-inet6.0: 2/2/2/0
L3VPN.inet.0: 2/2/2/0
L3VPN.inet6.0: 2/2/2/0
2001:db8:8:255:8::8      64512     1270      1294      0      0 9:35:26 Establ
bgp.l3vpn.0: 2/2/2/0
bgp.l3vpn-inet6.0: 2/2/2/0
L3VPN.inet.0: 2/2/2/0
L3VPN.inet6.0: 2/2/2/0

```

And you can observe that the BGP sessions are established from P3 to PE3/PE4 and to PE1/PE2.

5.3.3 Configuring VPN on a Gateway Device

You need VPN service configuration on gateway devices to interoperate SRv6 service to MPLS service., The SRv6 header will be removed on the gateway device and the packet is looked-up in the VPN table and the MPLS header is imposed. For traffic in the reverse direction, , the MPLS header is removed on the gateway device, and the packet destination address is looked up in the VPN table and the SRv6 header is imposed. On the gateway device, the VPN instance needs to be configured so that the VPN service routes are downloaded into appropriate VPN tables:

```

set routing-instances L3VPN instance-type vrf
set routing-instances L3VPN route-distinguisher 5.5.5.5:5
set routing-instances L3VPN vrf-import L3VPN_import
set routing-instances L3VPN vrf-export L3VPN_export
set routing-instances L3VPN vrf-table-label;

```

The VPN import policies should have matching route targets for the VPN service advertised from PE3 or PE1:

```

set policy-options policy-statement L3VPN_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_import term 0 then accept

```

You can verify that the service route 7.77.1.0 received from PE3 is resolved on the MPLS tunnel and placed in the VPN table:

```

regress@P3# run show route 7.77.1.0
L3VPN.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
7.77.1.0/24      *[BGP/170] 00:00:37, localpref 100, from 2001:db8:7:255:7::7
                  AS path: I, validation-state: unverified
                  > to 5.57.1.2 via ge-0/0/3.0, Push 16

```

5.3.4 Next Hop Manipulation

Okay, let's now advertise the service prefixes from the gateway device P3 towards PE1 over SRv6. You need to set the next hop to self on the gateway device so that SRv6 to MPLS translation can happen. You configure export policies on P3 to export routes received from BGP protocol and change the next hop to self:

```
set policy-options policy-statement exportstatic1 term 1 from protocol bgp
set policy-options policy-statement exportstatic1 term 1 from family inet-vpn
set policy-options policy-statement exportstatic1 term 1 then next-hop self
set policy-options policy-statement exportstatic1 term 1 then accept
```

Set the export policy on the SRv6 peer group:

```
set protocols bgp group internal-peers export exportstatic
```

Similarly, export policies would also have to be setup towards the MPLS peers:

```
set policy-options policy-statement exportstatic2 term exportstatic1 from protocol bgp
set policy-options policy-statement exportstatic2 term exportstatic1 then next-hop 5.5.5.5
set policy-options policy-statement exportstatic2 term exportstatic1 then accept
```

Note that the next-hop is set to 5.5.5.5 instead of to self. This is because the IBGP sessions we are advertising over are IPv6 sessions. If we used next-hop as self it would by default set to the IPv6 loopback address. The underlay tunnels towards PE3 are LDP tunnels and we have LDP labels only for IPv4 prefixes. In order to get the correct resolution you need to set the next-hop to 5.5.5.5 on P3 and 6.6.6.6 on P4.

Let's set the export policy of the MPLS peers:

```
set protocols bgp group internal-peers-mpls export exportstatic2
```

Verify the service routes are advertised with the correct next hops. Here are the service routes advertised towards PE1:

```
regress@P3# run show route advertising-protocol bgp 2001:db8:1:255:1::1
bgp.l3vpn.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix      Nexthop          MED Lclpref AS path
  2.2.2.2:1:2.22.1.0/24
*           Self            100   I
  2.2.2.2:1:22.22.22.22/32
*           Self            100   100 I
  7.7.7.7:1:7.77.1.0/24
*           Self            100   I
  7.7.7.7:1:77.77.77.77/32
*           Self            100   100 I
  8.8.8.8:1:8.88.1.0/24
*           Self            100   I
  8.8.8.8:1:88.88.88.88/32
*           Self            100   100 I
bgp.l3vpn-inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix      Nexthop          MED Lclpref AS path
  2.2.2.2:1:2001:db8:22::/48
*           2001:db8:a2:3::     100   I
  2.2.2.2:1:2001:db8:22:255:22::22/128
```

```

*          2001:db8:a2:3::      100    100 I
 7.7.7.7.1:2001:db8:77::/48
*          2001:db8:a2:3::      100    I
 7.7.7.7.1:2001:db8:77:255:77::77/128
*          2001:db8:a2:3::      100    100 I
 8.8.8.8.1:2001:db8:88::/48
*          2001:db8:a2:3::      100    I
 8.8.8.8.1:2001:db8:88:255:88::88/128
*          2001:db8:a2:3::      100    100 I

```

Service routes towards PE3 are advertised as shown here:

```

regress@P3# run show route advertising-protocol bgp 2001:db8:7:255:7::7
bgp.l3vpn.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED Lclpref AS path
  1.1.1.1:1:1.11.1.0/24
*          5.5.5.5          100    I
  1.1.1.1:1:11.11.11.11/32
*          5.5.5.5          100    100 I
  2.2.2.2:1:2.22.1.0/24
*          5.5.5.5          100    I
  2.2.2.2:1:22.22.22.22/32
*          5.5.5.5          100    100 I
  8.8.8.8.1:8.88.1.0/24
*          5.5.5.5          100    I
  8.8.8.8.1:88.88.88.88/32
*          5.5.5.5          100    100 I
bgp.l3vpn-inet6.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix      Nexthop      MED Lclpref AS path
  1.1.1.1:1:2001:db8:11::/48
*          5.5.5.5          100    I
  1.1.1.1:1:2001:db8:11:255:11::11/128
*          5.5.5.5          100    100 I
  2.2.2.2:1:2001:db8:22::/48
*          5.5.5.5          100    I
  2.2.2.2:1:2001:db8:22:255:22::22/128
*          5.5.5.5          100    100 I
  8.8.8.8.1:2001:db8:88::/48
*          5.5.5.5          100    I
  8.8.8.8.1:2001:db8:88:255:88::88/128
*          5.5.5.5          100    100 I

```

5.3.5 Static Route Configuration on Headend

When P3 advertises a service prefix with next hop set to self, the next hop address is an IPv6 loopback address of P3. Best effort SRv6 tunnels on the headend are created for a locator address. In order to correctly resolve the BGP service prefix protocol next hop, you need to configure a static route on the headend router for the IPv6 loopback that points to the correct locator address:

```

set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 next-hop 2001:db8:a2:3::
set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 resolve
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 next-hop 2001:db8:a2:4::
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 resolve

```

You can see that the service prefixes resolve correctly on PE1:

```
regress@PE1# run show route 7.77.1.0
L3VPN_PE1_CE1.inet.0: 9 destinations, 20 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
7.77.1.0/24    *[BGP/170] 06:46:47, localpref 100, from 2001:db8:5:255::5
                AS path: I, validation-state: unverified
                  to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:3::1
                  > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a2:4::1
[BGP/170] 06:46:47, localpref 100, from 2001:db8:6:255::6
                AS path: I, validation-state: unverified
                  > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a2:4::1
```

The service prefix on PE4:

```
regress@PE4# run show route 1.11.1.0
L3VPN_PE4_CE4.inet.0: 9 destinations, 20 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.11.1.0/24    *[BGP/170] 06:55:15, localpref 100, from 2001:db8:5:255::5
                AS path: I, validation-state: unverified
                  to 5.58.1.1 via ge-0/0/2.0, Push 16
                  > to 6.68.1.1 via ge-0/0/3.0, Push 17
[BGP/170] 06:55:15, localpref 100, from 2001:db8:6:255::6
                AS path: I, validation-state: unverified
                  > to 6.68.1.1 via ge-0/0/3.0, Push 17
```

5.3.6 Service Prefix Ping/traceroute Verification

Let's verify service route connectivity using ping and traceroute:

```
regress@PE2# run ping 7.77.1.1 routing-instance L3VPN_PE2_CE2
PING 7.77.1.1 (7.77.1.1): 56 data bytes
64 bytes from 7.77.1.1: icmp_seq=0 ttl=62 time=5.475 ms
64 bytes from 7.77.1.1: icmp_seq=1 ttl=62 time=4.748 ms
64 bytes from 7.77.1.1: icmp_seq=2 ttl=62 time=5.559 ms
^C
--- 7.77.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.748/5.261/5.559/0.364 ms
regress@PE2# run traceroute 7.77.1.2 routing-instance L3VPN_PE2_CE2
traceroute to 7.77.1.2 (7.77.1.2), 30 hops max, 52 byte packets
 1 * 128.207.209.55 (128.207.209.55)  3.629 ms  3.193 ms
 2 * 7.77.1.2 (7.77.1.2)  4.260 ms  4.744 ms
```

5.3.7 Converting the Network Back to SRv6 End-to-End

Let's now convert the lab topology back to its previous form by activating SRv6 on all nodes and removing the IBGP sessions on P3 and P4. Please refer to the Appendices for detailed configurations.

Chapter 6

BGP-LS Support with SRv6

6.1 Introduction

BGP-Link State (BGP-LS) provides mechanisms to advertise link state and traffic engineering information from the network to external components using BGP ([RFC 7752](#)). This information can be used by such components such as network controllers to create traffic-engineered paths, visualization, and network automation. BGP-LS is extended to carry SRv6 locators, SIDs, and other information as specified in draft [draft-ietf-idr-bgpls-srv6-ext](#). This chapter shows how to enable SRv6 in BGP-LS.

6.1.1 Setting Up Topology

In order to show the BGP-LS advertisements, we will set up the lab topology such that PE3 is a BGP-LS producer and PE1 is the consumer. In Junos, implementation the BGP-LS advertisements are derived from a Traffic Engineering Database (TED). We will enable `traffic-engineering` in ISIS which will construct the TED database and enable TED export into BGP-LS, as shown in Figure 6.1.

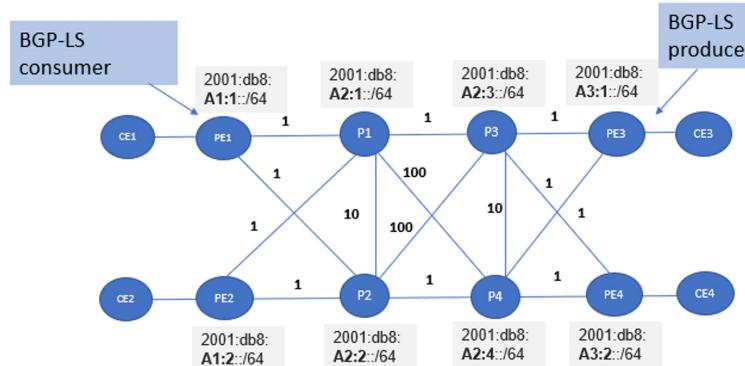


Figure 6.1

Setting up BGP-LS

6.1.2 BGP-LS Module Design

It is useful to understand the basic BGP-LS module design with various topologies and RIBs as shown in Figure 6.2.

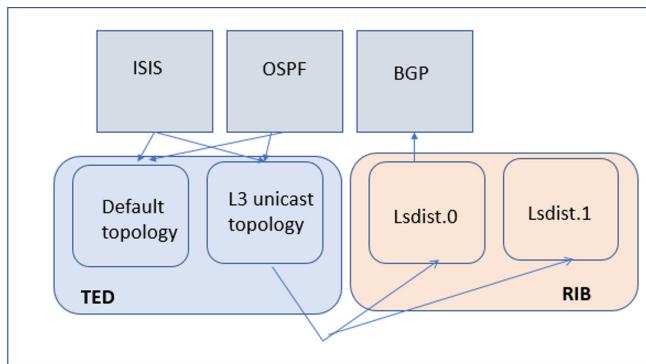


Figure 6.2 BGP-LS Software Design

Figure 6.2 depicts RPD module with ISIS, OSPF, BGP, TED and RIB submodules. ISIS and OSPF download the link state database into either default topology or l3-unicast topology or both. If RSVP is enabled the links are downloaded into default TED. If RSVP is not enabled, and the l3-unicast topology is enabled via a knob then the links are downloaded into l3-unicast topology.

The SRv6 use cases will not have RSVP enabled and hence will require l3-unicast topology to be enabled and nodes and links along with SRv6 SIDs downloaded into the l3-unicast topology TED. From the TED, BGP-LS NLRI are created and a RIB is populated with the routes. Lsdist.0 /Lsdist.1 are these RIB tables containing BGP-LS NLRI. By default, default TED is fed into Lsdist.0 and l3-unicast TED is fed into Lsdist.1. BGP will always listen to Lsdist.0 and advertise BGP-LS NLRI to peers. If l3-unicast topology needs to be advertised into BGP-LS, then a special knob has to be configured to download the contents of l3-unicast topology into Lsdist.0.

6.1.3 Enabling Traffic-engineering Advertisement

The first step towards BGP-LS is to enable ISIS traffic-engineering so that the TE attributes like `te-metric`, `admin-groups`, etc., can be advertised for the link. The `always` keyword ensures the TE attributes are advertised even if RSVP is not enabled. Enabling `l3-unicast-topology` facilitates downloading the link state database into a l3-unicast-topology TED.

Configure these commands on all the nodes in the network and verify if the TED is created with relevant information:

```
set protocols isis traffic-engineering advertisement always
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering ipv6
```

Verifying TED database L3-unicast topology. Verify all the links and nodes are present in the TED:

```
regress@PE1# ...show ted database topology-type l3-unicast
TED database: 8 ISIS nodes 8 INET nodes 8 INET6 nodes
ID          Type Age(s) LnkIn LnkOut Protocol
PE1.00(1.1.1.1)(2001:db8:1:255:1::1) Rtr 263 2    2 IS-IS(2)
  To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 1.14.1.1, Remote: 1.14.1.2
  IPv6 Local: 2001:db8:14::1, IPv6 Remote: 2001:db8:14::2
  Local interface index: 363, Remote interface index: 360
  To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 1.13.1.1, Remote: 1.13.1.2
  IPv6 Local: 2001:db8:13::1, IPv6 Remote: 2001:db8:13::2
  Local interface index: 362, Remote interface index: 360
ID          Type Age(s) LnkIn LnkOut Protocol
PE2.00(2.2.2.2)(2001:db8:2:255:2::2) Rtr 263 2    2 IS-IS(2)
  To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 2.24.1.1, Remote: 2.24.1.2
  IPv6 Local: 2001:db8:24::1, IPv6 Remote: 2001:db8:24::2
  Local interface index: 364, Remote interface index: 361
  To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 2.23.1.1, Remote: 2.23.1.2
  IPv6 Local: 2001:db8:23::1, IPv6 Remote: 2001:db8:23::2
  Local interface index: 363, Remote interface index: 361
ID          Type Age(s) LnkIn LnkOut Protocol
P1.00(3.3.3.3)(2001:db8:3:255:3::3) Rtr 217 5    5 IS-IS(2)
  To: PE1.00(1.1.1.1)(2001:db8:1:255:1::1), Local: 1.13.1.2, Remote: 1.13.1.1
  IPv6 Local: 2001:db8:13::2, IPv6 Remote: 2001:db8:13::1
  Local interface index: 360, Remote interface index: 362
  To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 3.34.1.1, Remote: 3.34.1.2
  IPv6 Local: 2001:db8:34::1, IPv6 Remote: 2001:db8:34::2
  Local interface index: 357, Remote interface index: 357
  To: PE2.00(2.2.2.2)(2001:db8:2:255:2::2), Local: 2.23.1.2, Remote: 2.23.1.1
  IPv6 Local: 2001:db8:23::2, IPv6 Remote: 2001:db8:23::1
  Local interface index: 361, Remote interface index: 363
  To: P4.00(6.6.6.6)(2001:db8:6:255:6::6), Local: 3.36.1.1, Remote: 3.36.1.2
  IPv6 Local: 2001:db8:36::1, IPv6 Remote: 2001:db8:36::2
  Local interface index: 359, Remote interface index: 357
  To: P3.00(5.5.5.5)(2001:db8:5:255:5::5), Local: 3.35.1.1, Remote: 3.35.1.2
  IPv6 Local: 2001:db8:35::1, IPv6 Remote: 2001:db8:35::2
  Local interface index: 358, Remote interface index: 357
ID          Type Age(s) LnkIn LnkOut Protocol
P2.00(4.4.4.4)(2001:db8:4:255:4::4) Rtr 263 5    5 IS-IS(2)
  To: PE1.00(1.1.1.1)(2001:db8:1:255:1::1), Local: 1.14.1.2, Remote: 1.14.1.1
  IPv6 Local: 2001:db8:14::2, IPv6 Remote: 2001:db8:14::1
  Local interface index: 360, Remote interface index: 363
  To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 3.34.1.2, Remote: 3.34.1.1
  IPv6 Local: 2001:db8:34::2, IPv6 Remote: 2001:db8:34::1
  Local interface index: 357, Remote interface index: 357
  To: PE2.00(2.2.2.2)(2001:db8:2:255:2::2), Local: 2.24.1.2, Remote: 2.24.1.1
  IPv6 Local: 2001:db8:24::2, IPv6 Remote: 2001:db8:24::1
  Local interface index: 361, Remote interface index: 364
  To: P4.00(6.6.6.6)(2001:db8:6:255:6::6), Local: 4.46.1.1, Remote: 4.46.1.2
  IPv6 Local: 2001:db8:46::1, IPv6 Remote: 2001:db8:46::2
  Local interface index: 359, Remote interface index: 358
  To: P3.00(5.5.5.5)(2001:db8:5:255:5::5), Local: 4.45.1.1, Remote: 4.45.1.2
  IPv6 Local: 2001:db8:45::1, IPv6 Remote: 2001:db8:45::2
```

```

Local interface index: 358, Remote interface index: 358
ID          Type Age(s) LnkIn LnkOut Protocol
P3.00(5.5.5.5)(2001:db8:5:255:5::5) Rtr 263 5 5 IS-IS(2)
To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 4.45.1.2, Remote: 4.45.1.1
IPv6 Local: 2001:db8:45::2, IPv6 Remote: 2001:db8:45::1
Local interface index: 358, Remote interface index: 358
To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 3.35.1.2, Remote: 3.35.1.1
IPv6 Local: 2001:db8:35::2, IPv6 Remote: 2001:db8:35::1
Local interface index: 357, Remote interface index: 358
To: P4.00(6.6.6.6)(2001:db8:6:255:6::6), Local: 5.56.1.1, Remote: 5.56.1.2
IPv6 Local: 2001:db8:56::1, IPv6 Remote: 2001:db8:56::2
Local interface index: 359, Remote interface index: 359
To: PE4.00(8.8.8.8)(2001:db8:8:255:8::8), Local: 5.58.1.1, Remote: 5.58.1.2
IPv6 Local: 2001:db8:58::1, IPv6 Remote: 2001:db8:58::2
Local interface index: 361, Remote interface index: 364
To: PE3.00(7.7.7.7)(2001:db8:7:255:7::7), Local: 5.57.1.1, Remote: 5.57.1.2
IPv6 Local: 2001:db8:57::1, IPv6 Remote: 2001:db8:57::2
Local interface index: 360, Remote interface index: 364
ID          Type Age(s) LnkIn LnkOut Protocol
P4.00(6.6.6.6)(2001:db8:6:255:6::6) Rtr 48 5 5 IS-IS(2)
To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 4.46.1.2, Remote: 4.46.1.1
IPv6 Local: 2001:db8:46::2, IPv6 Remote: 2001:db8:46::1
Local interface index: 358, Remote interface index: 359
To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 3.36.1.2, Remote: 3.36.1.1
IPv6 Local: 2001:db8:36::2, IPv6 Remote: 2001:db8:36::1
Local interface index: 357, Remote interface index: 359
To: P3.00(5.5.5.5)(2001:db8:5:255:5::5), Local: 5.56.1.2, Remote: 5.56.1.1
IPv6 Local: 2001:db8:56::2, IPv6 Remote: 2001:db8:56::1
Local interface index: 359, Remote interface index: 359
To: PE4.00(8.8.8.8)(2001:db8:8:255:8::8), Local: 6.68.1.1, Remote: 6.68.1.2
IPv6 Local: 2001:db8:68::1, IPv6 Remote: 2001:db8:68::2
Local interface index: 361, Remote interface index: 365
To: PE3.00(7.7.7.7)(2001:db8:7:255:7::7), Local: 6.67.1.1, Remote: 6.67.1.2
IPv6 Local: 2001:db8:67::1, IPv6 Remote: 2001:db8:67::2
Local interface index: 360, Remote interface index: 365
ID          Type Age(s) LnkIn LnkOut Protocol
PE3.00(7.7.7.7)(2001:db8:7:255:7::7) Rtr 263 2 2 IS-IS(2)
To: P4.00(6.6.6.6)(2001:db8:6:255:6::6), Local: 6.67.1.2, Remote: 6.67.1.1
IPv6 Local: 2001:db8:67::2, IPv6 Remote: 2001:db8:67::1
Local interface index: 365, Remote interface index: 360
To: P3.00(5.5.5.5)(2001:db8:5:255:5::5), Local: 5.57.1.2, Remote: 5.57.1.1
IPv6 Local: 2001:db8:57::2, IPv6 Remote: 2001:db8:57::1
Local interface index: 364, Remote interface index: 360
ID          Type Age(s) LnkIn LnkOut Protocol
PE4.00(8.8.8.8)(2001:db8:8:255:8::8) Rtr 263 2 2 IS-IS(2)
To: P4.00(6.6.6.6)(2001:db8:6:255:6::6), Local: 6.68.1.2, Remote: 6.68.1.1
IPv6 Local: 2001:db8:68::2, IPv6 Remote: 2001:db8:68::1
Local interface index: 365, Remote interface index: 361
To: P3.00(5.5.5.5)(2001:db8:5:255:5::5), Local: 5.58.1.2, Remote: 5.58.1.1
IPv6 Local: 2001:db8:58::2, IPv6 Remote: 2001:db8:58::1
Local interface index: 364, Remote interface index: 361

```

You can verify the SRv6 End.X SIDs are present in the links, SRv6 locators and the End SIDs are present in TED database by using an extensive command:

```

regress@PE1# run show ted database topology-type l3-unicast PE1 extensive
TED database: 8 ISIS nodes 8 INET nodes 8 INET6 nodes
NodeID: PE1.00(1.1.1.1)(2001:db8:1:255:1::1)

```

```

Type: Rtr, Age: 311 secs, LinkIn: 2, LinkOut: 2
Protocol: IS-IS(2)
1.1.1.1, 128.207.209.58
  To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 1.14.1.1, Remote: 1.14.1.2
  IPv6 Local: 2001:db8:14::1, IPv6 Remote: 2001:db8:14::2
  Local interface index: 363, Remote interface index: 360
  Color: 0 <none>
  Metric: 1
  IGP metric: 1
  Static BW: 1000Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 0bps      [1] 0bps      [2] 0bps      [3] 0bps
      [4] 0bps      [5] 0bps      [6] 0bps      [7] 0bps
    P2P Adjacency-SID:
      IPV6, SID: 299904, Flags: 0xb0, Weight: 0
      IPV4, SID: 299888, Flags: 0x30, Weight: 0
SRV6 End.X SID:
  P2P SRV6 End-X-SID:, SID: 2001:db8:a1:1:214::0, Flags: --P, Weight: 0, Algorithm: 0,
Flavor: PSP, USD
  To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 1.13.1.1, Remote: 1.13.1.2
  IPv6 Local: 2001:db8:13::1, IPv6 Remote: 2001:db8:13::2
  Local interface index: 362, Remote interface index: 360
  Color: 0 <none>
  Metric: 1
  IGP metric: 1
  Static BW: 1000Mbps
  Interface Switching Capability Descriptor(1):
    Switching type: Packet
    Encoding type: Packet
    Maximum LSP BW [priority] bps:
      [0] 0bps      [1] 0bps      [2] 0bps      [3] 0bps
      [4] 0bps      [5] 0bps      [6] 0bps      [7] 0bps
    P2P Adjacency-SID:
      IPV6, SID: 299936, Flags: 0xb0, Weight: 0
      IPV4, SID: 299920, Flags: 0x30, Weight: 0
SRV6 End.X SID:
  P2P SRV6 End-X-SID:, SID: 2001:db8:a1:1:213::0, Flags: --P, Weight: 0, Algorithm: 0,
Flavor: PSP, USD
  Prefixes:
    1.1.1.1/32
    IGP Flags: 0x00
    128.207.209.58/32
    IGP Flags: 0x00
    2001:db8:1:255:1::1/128
    IGP Flags: 0x00
    2001:db8:13::/64
    Metric: 1, IGP Flags: 0x00
    2001:db8:14::/64
    Metric: 1, IGP Flags: 0x00
    2001:db8:a1:1::/64
    IGP Flags: 0x00
SRV6 Locators:
  2001:db8:a1:1::/64
  2001:db8:a1:1:11::, 0x00, USD
  2001:db8:a1:1:12::, 0x00, PSP

```

```

SPRING-Algorithms:
  Algo: 0
  Algo: 1
SRV6 Capable: - Flags: 0
SRV6 Node MSD:
  - Type: 41, Value: 6
  - Type: 42, Value: 7
  - Type: 43, Value: 5
  - Type: 44, Value: 6
  - Type: 45, Value: 6

```

6.1.4 Enabling TED Import

In order to create the lsdist tables, enable TED import policies and apply on the TED.
 This will feed the lsdist.0 and lsdist.1 tables:

```
set policy-options policy-statement ted_import then accept
set protocols mpls traffic-engineering database import policy ted_import
```

In order to push l3-unicast topology into lsdist.0 use this configuration:

```
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state
```

Verify the lsdist.0 table has all the NLRI's including SRv6 information:

```

egress@PE3# run show route table lsdist.0
lsdist.1: 84 destinations, 84 routes (84 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
NODE { AS:64512 ISO:0019.2168.1000.00 ISIS-L2:0 }/1216
  *[IS-IS/18] 00:00:20
    Fictitious
NODE { AS:64512 ISO:0019.2168.2000.00 ISIS-L2:0 }/1216
  *[IS-IS/18] 00:00:20
    Fictitious
NODE { AS:64512 ISO:0019.2168.3000.00 ISIS-L2:0 }/1216
  *[IS-IS/18] 00:00:20
    Fictitious
NODE { AS:64512 ISO:0019.2168.4000.00 ISIS-L2:0 }/1216
  *[IS-IS/18] 00:00:20
    Fictitious
NODE { AS:64512 ISO:0019.2168.5000.00 ISIS-L2:0 }/1216
  *[IS-IS/18] 00:00:20
    Fictitious
NODE { AS:64512 ISO:0019.2168.6000.00 ISIS-L2:0 }/1216

...
LINK { Local { AS:64512 ISO:0019.2168.1000.00 }.{ IPv4:1.13.1.1 } Remote { AS:64512
ISO:0019.2168.3000.00 }.{ IPv4:1.13.1.2 } ISIS-L2:0 }/1216
  *[IS-IS/18] 00:13:18
    Fictitious
LINK { Local { AS:64512 ISO:0019.2168.1000.00 }.{ IPv4:1.14.1.1 } Remote { AS:64512
ISO:0019.2168.4000.00 }.{ IPv4:1.14.1.2 } ISIS-L2:0 }/1216
  *[IS-IS/18] 00:13:18
    Fictitious
LINK { Local { AS:64512 ISO:0019.2168.2000.00 }.{ IPv4:2.23.1.1 } Remote { AS:64512
ISO:0019.2168.3000.00 }.{ IPv4:2.23.1.2 } ISIS-L2:0 }/1216

```

```

*[IS-IS/18] 00:13:18
  Fictitious
LINK { Local { AS:64512 ISO:0019.2168.2000.00 }.{ IPv4:2.24.1.1 } Remote { AS:64512
ISO:0019.2168.4000.00 }.{ IPv4:2.24.1.2 } ISIS-L2:0 }/1216
*[IS-IS/18] 00:13:18
  Fictitious
LINK { Local { AS:64512 ISO:0019.2168.3000.00 }.{ IPv4:1.13.1.2 } Remote { AS:64512
ISO:0019.2168.1000.00 }.{ IPv4:1.13.1.1 } ISIS-L2:0 }/1216
*[IS-IS/18] 00:13:18
  Fictitious
...
PREFIX { Node { AS:64512 ISO:0019.2168.1000.00 } { IPv4:1.1.1.1/32 } ISIS-L2:0 }/1216
*[IS-IS/18] 00:13:51
  Fictitious
PREFIX { Node { AS:64512 ISO:0019.2168.1000.00 } { IPv4:128.207.209.58/32 } ISIS-L2:0 }/1216
*[IS-IS/18] 00:13:51
  Fictitious
PREFIX { Node { AS:64512 ISO:0019.2168.2000.00 } { IPv4:2.2.2.2/32 } ISIS-L2:0 }/1216
...
IPV6 PREFIX { Node { AS:64512 ISO:0019.2168.1000.00 } { IPv6:2001:db8:1:255:1::1/128 } ISIS-L2:0
}/1216
*[IS-IS/18] 00:14:17
  Fictitious
IPV6 PREFIX { Node { AS:64512 ISO:0019.2168.1000.00 } { IPv6:2001:db8:13::/64 } ISIS-L2:0 }/1216
...
SRV6 End SID { Node { AS:64512 ISO:0019.2168.1000.00 } { End SID:2001:db8:a1:1:11:: } ISIS-L2:0 }/1216
*[IS-IS/18] 00:15:30
  Fictitious
SRV6 End SID { Node { AS:64512 ISO:0019.2168.1000.00 } { End SID:2001:db8:a1:1:12:: } ISIS-L2:0 }/1216
*[IS-IS/18] 00:15:30
  Fictitious

```

SRv6 End.X SIDs will be attributes in link NLRI. SRv6 End SIDs will appear as separate NLRI.

As the SRV6 SIDs are separate NLRI, they allow for more granular SID updates. If they were node attributes, then all node attributes along with all SIDs had to be updated if one SID changes. The SRV6 SID NLRI structure defined by the draft [draft-ietf-idr-bgpls-srv6-ext](#) allows for more granular and compact updates.

6.1.5 Enabling BGP-LS Family

Enable BGP-LS family on the BGP session.

On PE1:

```
set protocols bgp group ibgp family traffic-engineering unicast
```

On PE3:

```
set protocols bgp group ibgp family traffic-engineering unicast
set protocols bgp group ibgp export nlri2bgp
```

PE3 is a BGP-LS producer. Configure export policies to export NLRI from the lsdist.0 table:

```
regress@PE3# run show route table lsdist.0 extensive
lsdist.1: 84 destinations, 84 routes (84 active, 0 holddown, 0 hidden)
NODE { AS:64512 ISO:0019.2168.1000.00 ISIS-L2:0 }/1216 (1 entry, 0 announced)
  *IS-IS Preference: 18
  Level: 2
  Next hop type: Fictitious, Next hop index: 0
  Address: 0x747c634
  Next-hop reference count: 84, key opaque handle: 0x0, non-key opaque handle: 0x0
  Next hop:
    State: <Active NotInstall>
    Local AS: 64512
    Age: 5:51
    Validation State: unverified
    Task: IS-IS
    AS path: I
    IPv4 Router-ids:
      1.1.1.1
    Area border router: No
    External router: No
    Attached: No
    Overload: No
    Hostname: PE1
    Area membership:
      47 00 05
      47 00 05 80 ff f8 00 00 00 01 08 00 01
    SPRING-Algorithms:
      - Algo: 0
      - Algo: 1
    SRV6 Capable: - Flags: 0
    SRV6 Node MSD:
      - Type: 41, Value: 6
      - Type: 42, Value: 7
      - Type: 43, Value: 5
      - Type: 44, Value: 6
      - Type: 45, Value: 6
    Thread: junos-main

SRV6 End SID { Node { AS:64512 ISO:0019.2168.1000.00 } { End SID:2001:db8:a1:1:11:: } ISIS-L2:0 }/1216 (1
entry, 0 announced)
  *IS-IS Preference: 18
  Level: 2
  Next hop type: Fictitious, Next hop index: 0
  Address: 0x747c634
  Next-hop reference count: 84, key opaque handle: 0x0, non-key opaque handle: 0x0
  Next hop:
    State: <Active NotInstall>
    Local AS: 64512
    Age: 50
    Validation State: unverified
    Task: IS-IS
    AS path: I
```

```

Flavor: USD, Flags: 0, Algo: 0
Thread: junos-main

SRV6 End SID { Node { AS:64512 ISO:0019.2168.1000.00 } { End SID:2001:db8:a1:1:12:: } ISIS-L2:0 }/1216
(1 entry, 0 announced)
  *IS-IS Preference: 18
  Level: 2
  Next hop type: Fictitious, Next hop index: 0
  Address: 0x747c634
  Next-hop reference count: 84, key opaque handle: 0x0, non-key opaque handle: 0x0
  Next hop:
  State: <Active NotInstall>
  Local AS: 64512
  Age: 50
  Validation State: unverified
  Task: IS-IS
  AS path: I
  Flavor: PSP, Flags: 0, Algo: 0
  Thread: junos-main

```

Verify BGP-LS advertisements received and installed in lsdist.0 on PE1:

```

regress@PE1# run show route table lsdist.0
lsdist.0: 136 destinations, 136 routes (136 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
NODE { AS:64512 ISO:0019.2168.1000.00 ISIS-L2:0 }/1216
  *[BGP/170] 00:21:06, localpref 100, from 7.7.7.7
    AS path: I, validation-state: unverified
    > to 1.13.1.2 via ge-0/0/2.0
      to 1.14.1.2 via ge-0/0/3.0
NODE { AS:64512 ISO:0019.2168.2000.00 ISIS-L2:0 }/1216
  *[BGP/170] 00:21:06, localpref 100, from 7.7.7.7
    AS path: I, validation-state: unverified
    > to 1.13.1.2 via ge-0/0/2.0
      to 1.14.1.2 via ge-0/0/3.0
NODE { AS:64512 ISO:0019.2168.3000.00 ISIS-L2:0 }/1216
  *[BGP/170] 00:21:06, localpref 100, from 7.7.7.7
    AS path: I, validation-state: unverified
    > to 1.13.1.2 via ge-0/0/2.0
      to 1.14.1.2 via ge-0/0/3.0
NODE { AS:64512 ISO:0019.2168.4000.00 ISIS-L2:0 }/1216
  *[BGP/170] 00:21:06, localpref 100, from 7.7.7.7
    AS path: I, validation-state: unverified
    > to 1.13.1.2 via ge-0/0/2.0
.....

```

NOTE BGP-LS family can only be enabled on an IPv4 BGP session. BGP-LS family on IPv6 session will be supported in the near future.

Chapter 7

Configuring SRv6 Flex Algo

7.1 Introduction

Flex Algo is a mechanism that allows network operators to influence how the IGP calculates the least cost path for each prefix segment. This chapter will discuss how you can configure flex-algo with the SRv6 data plane and how services can be mapped to those.

The basic concepts of flex-algo remain the same for SRv6. There is a flex-algo definition which defines the constraints that need to be used for flex-algo path computation. There is flex-algo participation when each node is configured with flex-algo in ISIS and it advertises the flex-algo in algorithm sub-TLV.

The flex-algo definition and the flex-algo participation information is used to derive the flex-algo topology. SPF computation is done on this topology to get the least cost paths. The main difference between SR-MPLS flex-algo and SRv6 flex-algo is that the SR-MPLS flex-algo would have multiple Prefix-SIDs associated with same prefix on a per algorithm basis. In SRv6, you will have to configure a per flex-algo locator. You will also have to configure a per flex-algo End SID and End.X SIDs. The flex-algo End and End.X SIDs will have to be derived from the flex-algo locator. Junos supports all the constraints supported for SR-MPLS such as delay metric, te-metric, include/exclude admin groups, etc., in SRv6 as well. Note that admin group configurations are tightly coupled with MPLS and will require protocol `mpls` to be enabled.

This chapter concentrates on a flex algo based on delay metric for illustration purposes.

7.2 Delay Metric

There are two ways a delay metric can be derived in IGP. One via static configuration on the interface and another via dynamic latency measurement using the TWAMP protocol. The dynamic latency measurement method uses periodic latency measurements and can account for optical layer re-routing.

7.2.1 Deriving the Metric Dynamically Using TWAMP

The network performance is measured using TWAMP-Light. TWAMP-Light provides capabilities for the measurement of various performance metrics in IP networks using probe messages.

End-to-end minimum delay metric paths can be built by defining an ISIS flex algorithm having Latency metric as the metric-type which minimizes the latency to the endpoint.

Junos supports point-to-point links only for this feature. Ethernet links that are set to point-to-point interface type are also supported.

Enabling/Disabling Link Delay Measurement for an ISIS Interface

```
set protocols isis interface <interface_name> delay-measurement
```

When this knob is configured, delay measurement will be started for this link and the following new sub-TLVs will be generated:

- Unidirectional Link Delay
- Min/Max Unidirectional Link Delay
- Unidirectional Link Delay Variation

```
delete protocols isis interface <interface_name> delay-measurement
```

When this knob is unconfigured, delay measurement will be stopped for this link and delay metrics are unpublished by removing sub-TLVs from ISIS.

Link delay measurement probe interval configuration, unconfiguration

```
set protocols isis interface <interface_name> delay-measurement probe-interval <1-255>
delete protocols isis interface <interface_name> delay-measurement probe-interval
```

When the probe-interval is set to a time interval the interface delay is measured at the configured interval and reports it to ISIS. Longer time intervals mean slower response. Smaller intervals may be too heavy on IGP advertisement and flooding.

The probe interval default value is 3 seconds.

Link delay measurement probe count configuration, unconfiguration

```
set protocols isis interface <interface_name> delay-measurement probe-count <1-15>
delete protocols isis interface <interface_name> delay-measurement probe-count
```

The probe count default value is 10. The probing can be periodic or accelerated. Periodic means the delay is measured periodically and if it's greater than the threshold, then the measured value is reported via ISIS. In case of accelerated mode, the probing is at an accelerated rate and as soon as the change is more than the accelerated threshold the value is reported in ISIS.

Periodic interval configuration/unconfiguration

By default, periodic advertisement is enabled.

```
set protocols isis interface <interface_name> delay-measurement advertisement periodic interval
<30-3600>
delete protocols isis interface <interface_name> delay-measurement advertisement periodic interval
```

The periodic interval default value is 120 seconds

Periodic threshold configuration/unconfiguration

```
set protocols isis interface <interface_name> delay-measurement advertisement periodic threshold <
0-100>
delete protocols isis interface <interface_name> delay-measurement advertisement periodic threshold
```

Periodic threshold default value is 10% (percentage).

Accelerated Advertisement enable/disable

By default, accelerated advertisement is disabled. This is to enable accelerated advertisement:

```
set protocols isis interface <interface_name> delay-measurement advertisement accelerated
```

This is for disabling accelerated advertisement:

```
delete protocols isis interface <interface_name> delay-measurement advertisement accelerated
```

Accelerated threshold configuration/unconfiguration

```
set protocols isis interface <interface_name> delay-measurement advertisement accelerated threshold <
0-100>
delete protocols isis interface <interface_name> delay-measurement advertisement accelerated threshold
```

Accelerated threshold default value is 20% (percentage).

7.2.2 Configuring Static Delay Metric

Delay Metric Config/Unconfig for an interface

When `delay-metric` is configured for an interface, minimum, maximum, and average delays for the associated link are set to the `delay-metric` value (microseconds) and the delay variance for the link is set to zero:

```
set protocols isis interface <interface_name> delay-metric <0-16777215>
```

When `delay-metric` is unconfigured for an interface, delay metrics are unpublished by removing sub-TLVs from ISIS:

```
delete protocols isis interface <interface_name> delay-metric
```

7.3 Flex Algo with Delay Metric

Configuring/Unconfiguring flex-algorithm definition with metric-type delay

When this knob is configured, flex algo SPF computation uses delay metric. Flex algo definition needs to be configured only on one or two nodes. Configuring only on two nodes has the advantage that there is less chance of mis-configurations at the same time you get the minimum reliability:

```
set routing-options flex-algorithm <id> definition metric-type delay
```

When this knob is unconfigured, triggers flex-algo spf and compute routes based on IGP metric:

```
delete routing-options flex-algorithm <id> definition metric-type delay
```

It is difficult to get different dynamic delay measurements on the links in a virtual lab setup, so we will use a statically configured delay metric for this exercise (see Figure 7.1). However, the dynamic delay measurement works similar to the static configured delay.

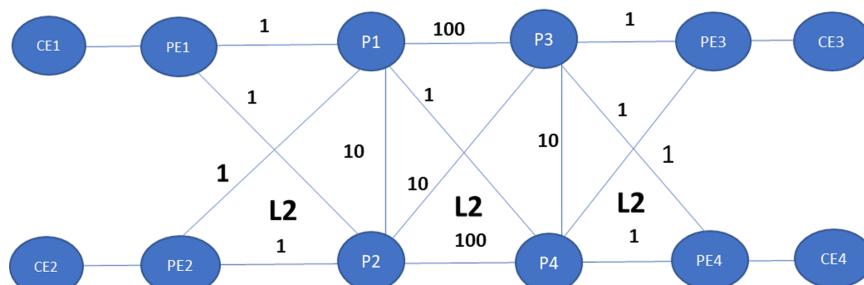


Fig 7.1

Flex Algo Topology with Delay Metric Configuration on Links

This topology that has a flat L2 (Level 2) domain. All the PEs and P routers are part of normal algo 0 as well as algo 128 flex algo. The flex algo 128 is defined to be based on the delay metric. The algo 0 has all the default metrics and the delay metric for flex algo 128 is shown in Figure 7.1. The flex algo definition is configured on PE3 and P3.

There is an SRv6 locator per flex algo configured on all participating routers. Also two End SIDs with the `usd` and `psp` flavor are configured on those locators. Transport class infrastructure is enabled on these routers. Transport class infrastructure is explained in detail in Chapter 10.4.2.

When the flex-algo locators and the FAD is configured, every router computes the flex algo path to the corresponding locator and downloads the ingress route in `inet6.3` color table and the transit locator routes in `inet6.0` table.

7.3.1 Locator Design for Flex Algo

Table 7.1 Locator design for flex algo lab topology

Locator Block (48 bits)				Node id (16 bits)	
Prefix	Reserved	Flex-algo	Sub-region	Reserved	NE ID
32 bits	8 bits	4 bits	4 bits	8 bits	8 bits
2001:db8	0	b	1	0	1

As listed in Table 7.1, the flex algo locator naming follows the paradigm defined in Chapter 2. Here's the configuration required at PE1. Refer to the Appendices for the configurations on all other routers.

PE1:

Configure flex-algo specific locator loc2:

```
set routing-options source-packet-routing srv6 locator loc2 2001:db8:b1:1::/64 algorithm 128 >> This creates a algo 128 locator
```

Configure delay metric on the links:

```
set protocols isis interface ge-0/0/2.0 delay-metric 1 >> static delay metric
set protocols isis interface ge-0/0/3.0 delay-metric 1
```

Configure flex algo specific End.x sids:

```
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2 end-x-sid 2001:db8:b1:1:213:0:0:0
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2 end-x-sid 2001:db8:b1:1:214:0:0:0
```

Enable flex-algo 128 on ISIS:

```
set protocols isis source-packet-routing srv6 locator loc2
set protocols isis source-packet-routing flex-algorithm 128
```

Configure flex-algo specific End SIDs:

```
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b1:1:11::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b1:1:12::0 flavor psp
```

Flex-algo definition configuration:

```
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class
```

7.3.2 Flex Algo Routes

You can see that the remote flex algo usd-flavored sids and remote algo 128 locator routes are present in the 128color transport class table:

```
root@PE1# run show route table junos-rti-tc-128.inet6.3
junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 10:17:25, metric 1
    > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:1::
2001:db8:b2:1:11::/128
    *[SRV6-ISIS/14] 10:17:25, metric 1
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:11::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 10:17:25, metric 1
        > to fe80::5604:dff:fe00:6d7 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2::
2001:db8:b2:2:11::/128
        *[SRV6-ISIS/14] 10:17:25, metric 1
        > to fe80::5604:dff:fe00:6d7 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:11::
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 10:17:25, metric 4
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3::
2001:db8:b2:3:11::/128
        *[SRV6-ISIS/14] 10:17:25, metric 4
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:11::
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 10:17:25, metric 2
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:::
2001:db8:b2:4:11::/128
        *[SRV6-ISIS/14] 10:17:25, metric 2
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:11::
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 10:17:25, metric 3
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:::
2001:db8:b3:1:11::/128
        *[SRV6-ISIS/14] 10:17:25, metric 3
        > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:11::
```

7.4. Ping and Traceroute for Verifying SRv6 Flex-algo Transport

Once the locators and SIDs are configured, you can use SRv6 ping and traceroute to validate the connectivity to the SIDs:

```
root@PE1> ping srv6 isis 2001:db8:b3:1:11::0
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.104/112.483/523.693/205.633 ms
root@PE1> traceroute srv6 isis 2001:db8:b3:1:11::0
traceroute to destination 2001:db8:b3:1:11::1, from source 2001:db8:1:255:1::1, 64 hops max, retries 3
 1  2001:db8:13::2  2.762 ms  2.564 ms  2.346 ms
    DA: 2001:db8:b3:1:11::
 2  2001:db8:36::2  3.764 ms  3.512 ms  3.301 ms
    DA: 2001:db8:b3:1:11::
 3  2001:db8:57::2  5.685 ms  4.857 ms  5.137 ms
    DA: 2001:db8:b3:1:11::
Traceroute status: Success
```

7.5 Configuring Services to Use SRv6 Flex Algo

In Chapter 3, normal (algo 0) DT4/DT6 SIDs were used for services. If you prefer to use a flex algo-based path for a service, then the service needs to use the SIDs from the algo locator. For explanation purposes, let's assume the IPv4 traffic users prefer to use the low latency path through the network.

You can see that the CE1 to CE2 ping is working in the topology:

```
root@PE1# run ping 77.77.77.77 logical-system CE1
PING 77.77.77.77 (77.77.77.77): 56 data bytes
64 bytes from 77.77.77.77: icmp_seq=0 ttl=62 time=9.547 ms
64 bytes from 77.77.77.77: icmp_seq=1 ttl=62 time=7.570 ms
^C
```

There are two ways to map services onto the SRv6 flex algo transport, using extended color community and using without it.

7.5.1 Without Color Community Service Mapping

In SRv6, each flex algo will have a separate locator configured. Since the locators do not conflict with the default algorithm or with other flex algos, the flex algo locator routes can be installed in inet6.3

The service prefix next hop can be updated to advertise the flex algo locator. In this method, ISIS will install the SRv6 flex algo locators into inet6.3 (with the help of a knob). The service prefixes with flex algo service SIDs can resolve over inet6.3 (using longest prefix match to the flex algo locators).

At egress PE, PE3, the DT4 SID, is configured out of the flex algo locator. The next hop is set to the algo 128 locator by using policies:

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc2 end-dt4-sid
2001:db8:b3:1:3777:0:0:0
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 2001:db8:b3:1::
deactivate routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1
end-dt4-sid
```

At ingress PE, PE1, first the flex algo locators are leaked into inet6.3 using the transport class sub knob:

```
root@PE1# set routing-options flex-algorithm 128 use-transport-class inet3-install
```

You can see that the route is present in both the 128transport class table and inet.3:

```
[edit]
root@PE1# run show route 2001:db8:b3:1::
...
inet6.3: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:01:18, metric 3
    > to fe80::5604:dff:fe00:107f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1::
junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:01:18, metric 3
    > to fe80::5604:dff:fe00:107f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1::
```

Here you can see that the service prefix got resolved over the helper flex algo 128 locator route present in inet6.3:

```
[edit]
root@PE1# run show route 77.77.77.77
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
77.77.77.77/32      *[BGP/170] 00:02:49, localpref 100, from 2001:db8:7:255:7::7
    AS path: 100 I, validation-state: unverified
        > to fe80::5604:dff:fe00:107f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1::
root@PE1# run ping 77.77.77.77 logical-system CE1
PING 77.77.77.77 (77.77.77.77): 56 data bytes
64 bytes from 77.77.77.77: icmp_seq=0 ttl=62 time=6.013 ms
64 bytes from 77.77.77.77: icmp_seq=1 ttl=62 time=5.883 ms
64 bytes from 77.77.77.77: icmp_seq=2 ttl=62 time=6.716 ms
```

Starting with Junos 22.3, the setting of next hop at egress PE to flex algo locator is not required. Without that configuration itself, the ingress will be able to resolve the SRv6 Service SID using the helper locator route.

At PE3:

```
delete policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop
```

7.5.2 With Color Community Service Mapping

Here, at egress PE, PE3 will advertise the color 128 along with the payload prefix advertisement. At ingress, the SRv6 SID will be resolved over the locator present in the 128transport class table.

At ingress PE, PE1 there is no need to leak the locator to inet6.3:

```
delete routing-options flex-algorithm 128 use-transport-class inet3-install
```

At egress PE3, the color community is attached along with the payload prefix advertisement:

```
set policy-options community com1 members color:0:128
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add com1
```

You can see that the payload prefix got resolved over the helper locator route present in the 128transport class:

```
root@PE1# run show route 77.77.77.77 extensive | match 128
  Transport class: junos-tc-128
    Communities: target:64512:1 color:0:128
      2001:db8:b3:1::/64 Originating RIB: junos-rti-tc-128.inet6.3
[edit]
root@PE1# run ping 77.77.77.77 logical-system CE1
PING 77.77.77.77 (77.77.77.77): 56 data bytes
64 bytes from 77.77.77.77: icmp_seq=0 ttl=62 time=7.915 ms
64 bytes from 77.77.77.77: icmp_seq=1 ttl=62 time=6.819 ms
```

Starting with Junos 22.3, the setting of next hop at the egress PE to flex algo locator is not required. The ingress itself will be able to resolve the SRv6 Service SID using the helper locator route. At PE3:

```
delete policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop
```

7.6 TILFA with SRv6 Flex Algo

Let's look at the TILFA backup path at P1 for the transit SRv6 flex algo plain traffic to PE3. You can see the PE3 flex algo 128 locator is 2001:db8:b3:1::/64. The primary path is P1-P4-PE3. At PE3:

```
set routing-options source-packet-routing srv6 locator loc2 2001:db8:b3:1::/64
```

Enable the transit TILFA at P1. At P1:

```
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
set protocols isis backup-spf-options use-post-convergence-lfa
set protocols isis backup-spf-options use-source-packet-routing
```

You can see in Figure 7.2, that the backup path is programmed via the post-convergence path P1-PE1-P2-P3-PE3.

- Primary path
- Post convergence/tilfa path

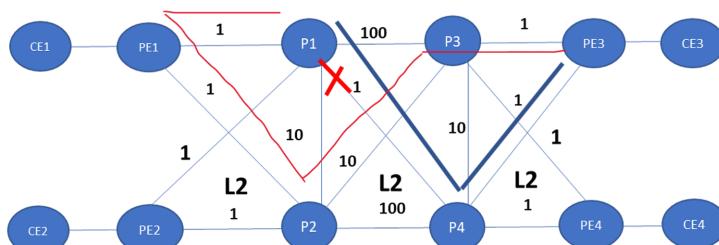


Fig 7.2

SRv6 Flex Algo TILFA Computation in Flex-algorithm Topology

The anchor points chosen for this path is the P2 End SID followed by the P2-P3 End.X Sid. As mentioned earlier, the P2-P3 End.X itself is sufficient, but more optimization is pending in Junos and will be addressed soon.

```
root@P1# run show route 2001:db8:b3:1::/64 extensive table inet6.0
inet6.0: 63 destinations, 63 routes (63 active, 0 holddown, 0 hidden)
2001:db8:b3:1::/64 (1 entry, 1 announced)
TSI:
KRT in-kernel 2001:db8:b3:1::/64 -> {fe80::5604:17ff:fe00:b586}
IS-IS level 2, SRv6 Loc Sid 0x93c9e60
 *IS-IS Preference: 18
   Level: 2
   Downbit: 256
   Next hop type: List, Next hop index: 0
   Address: 0x7da3c24
   Next-hop reference count: 1, key opaque handle: 0x0, non-key opaque handle: 0x0
   Next hop: ELNH Address 0x787d434 weight 0x1, selected
   .....
   Next hop: fe80::5604:17ff:fe00:b586 via ge-0/0/2.0 weight 0x1
   Next hop: ELNH Address 0x787ed94 weight 0xf000
   .....
   SRV6-Tunnel: Reduced-SRH Insert-mode
   Src: 2001:db8:3:255:3::3 Dest: 2001:db8:b3:1::
   Segment-list[0] 2001:db8:b2:2:11::
   Segment-list[1] 2001:db8:b2:2:245::
     Next hop type: Router, Next hop index: 0
     Address: 0x787ce14
     Next-hop reference count: 25, key opaque handle: 0x0, non-key opaque handle: 0x0
     Next hop: fe80::5604:17ff:fe00:68fc via ge-0/0/3.0 weight 0xf000
   .....
```

The different flavors and protection at ingress and transit are same as described in the TILFA chapter; except that the SIDs chosen are from the corresponding flex algo locators in the failure path.

7.7 Troubleshooting Flex Algo

Day One: Migrating to Segment Routing (https://www.juniper.net/documentation/en_US/day-one-books/DO_MigrateSR.pdf) covers the basic procedures to troubleshoot flex algo related issues in its Chapter 9. For SRv6, the procedures mentioned under “Misconfiguration of Flex-Algo Definition”, “SPF Trigger Problems”, “Route Changes”, and “Internal ISIS Route Table Display” are applicable. The traffic counters however, are yet to be supported for SRv6.

Flex algo advertisements can be verified on a router by looking at the ISIS database. The ISIS database shows the advertisements from all nodes in an IGP domain and is a quick way to observe flex algo specific locator advertisements.

With flex algo, the locator TLV gets advertised with an algorithm field set to a flex algo specific number. The SRv6 End SIDs corresponding to the algorithm are placed under this locator TLV. The ISIS database output is shown below with the relevant TLV information highlighted:

```

root@PE1# run show isis database PE1 level 2 extensive
IS-IS level 2 link-state database:
PE1.00-00 Sequence: 0x42, Checksum: 0xe19, Lifetime: 1130 secs
.....
TLVs:
.....
SRv6 Locator: 2001:db8:b1:1::/64, Metric: 0, MTID: 0, Flags: 0x0, Algorithm: 128
SRv6 SID: 2001:db8:b1:1:11::, Flavor: USD
SRv6 SID: 2001:db8:b1:1:12::, Flavor: PSP
.....
Extended IS Reachability TLV, Type: 22, Length: 127
IS extended neighbor: P2.00, Metric: default 1 SubTLV len: 116
.....
Local interface index: 364, Remote interface index: 359
Unidirectional link delay: 1
Min unidirectional link delay: 1
Max unidirectional link delay: 1
Unidirectional delay variation: 0
P2P SRV6 End-X-SID:2001:db8:b1:1:214:: , Flags:--P, Weight:0, Algorithm:128
Flags:0x20(B:0,S:0,P:1), Flavor: None
.....
Extended IS Reachability TLV, Type: 22, Length: 127
IS extended neighbor: P1.00, Metric: default 1 SubTLV len: 116
.....
Local interface index: 363, Remote interface index: 359
Unidirectional link delay: 1
Min unidirectional link delay: 1
Max unidirectional link delay: 1
Unidirectional delay variation: 0
P2P SRV6 End-X-SID:2001:db8:b1:1:213:: , Flags:--P, Weight:0, Algorithm:128
Flags:0x20(B:0,S:0,P:1), Flavor: None
.....
Router Capability: Router ID 128.51.157.169, Flags: 0x00
SPRING Algorithm - Algo: 0
SPRING Algorithm - Algo: 1
SPRING Algorithm - Algo: 128
Flex Algo: 128, Len: 4, Metric: 1, Calc: 0, Prio: 0
.....
No queued transmissions

```

Only the locator TLV is sent in case of flex algo locators. The locator address is not sent in extended IP reach TLV unlike algo 0. Also observe the flex algo specific End.X SID advertisements in TLV 22.

The `show isis spring flex-algorithm` output is shown below. This output provides details about whether the node on which the command is executed is participating in the flexible algorithm. In case it is not participating, the reason will also be displayed. The `winner FAD` definition the node is using is also displayed. These details are useful in trouble shooting the flex algo feature. The details of how many SPFs were triggered on flex algo topology and the reason why it was triggered is also captured in the outputs:

```

root@PE1# run show isis spring flex-algorithm flex-algorithm-id 128
Flex Algo: 128
Level: 2, Color: 128, Participating, FAD supported
  Winner: PE1, Metric: 1, Calc: 0, Prio: 0, FAD supported
  Spf Version: 49

```

```

Participation toggles: 1
Last Route add: 0
Last Route rem: 0
Last Route chg: 0
Last Route fail: 0
Last Route def: 0
Total Route add: 18
Total Route rem: 0
Total Route chg: 3
Total Route fail: 0
Total Route def: 0
Topo refresh count: 2
Full SPFs: 49, Partial SPFs: 0
IS-IS Flex Algo 128 level 2 SPF log:
Start time      Elapsed (secs) Count Reason
Wed Jun  8 16:45:47    0.000149   1 Periodic SPF
Wed Jun  8 16:57:04    0.000141   1 Periodic SPF
.....

```

The `show isis route flex-algorithm-id` output is shown below. This command provides details of the flex-algo routes in ISIS module.

```

root@PE1# run show isis route flex-algorithm-id 128
IS-IS routing table          Current version: L1: 0 L2: 133
SRV6 Routes
-----
Prefix      L Version  Metric Type Interface      NH     Via
2001:db8:b1:1::/64 2        49      0 int
2001:db8:b1:1:11::/128 2    49      0 int
2001:db8:b1:1:12::/128 2    49      0 int
2001:db8:b2:1::/64 2        49      1 int
2001:db8:b2:2::/64 2        49      1 int
2001:db8:b2:3::/64 2        49      4 int
2001:db8:b2:4::/64 2        49      2 int
2001:db8:b3:1::/64 2        49      3 int
SRv6->INET6.3 Routes
-----
Prefix      L Version  Metric Type Interface      NH     Via          Backup Score
2001:db8:b2:1::/64 2        49      1 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b2:1:11::/128 2    49      1 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b2:2::/64 2        49      1 int ge-0/0/3.0    SRv6 TCNH
2001:db8:b2:2:11::/128 2    49      1 int ge-0/0/3.0    SRv6 TCNH
2001:db8:b2:3::/64 2        49      4 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b2:3:11::/128 2    49      4 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b2:4::/64 2        49      2 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b2:4:11::/128 2    49      2 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b3:1::/64 2        49      3 int ge-0/0/2.0    SRv6 TCNH
2001:db8:b3:1:11::/128 2    49      3 int ge-0/0/2.0    SRv6 TCNH

```

7.8 BGP-LS Updates with Flex Algo

In Chapter 6, we enabled a BGP-LS session on PE3 and imported TED to lsdist.0 table. Let's verify that with a flex algo configuration. The flex algo locators and SIDs get updated in TED and also get exported via BGP-LS.

Let's look at which router is advertising the flex algo definition with a show command to show the details of `flex-algorithm` as follows:

```
regress@PE3# run show isis spring flex-algorithm flex-algorithm-id 128
Flex Algo: 128
  Level: 2, Color: 128, Participating, FAD supported
    Winner: PE2, Metric: 1, Calc: 0, Prio: 0, FAD supported
    Spf Version: 68
    Participation toggles: 1
    Last Route add: 0
    Last Route rem: 0
    Last Route chg: 0
    Last Route fail: 0
    Last Route def: 0
    Total Route add: 24
    Total Route rem: 0
    Total Route chg: 0
    Total Route fail: 0
    Total Route def: 0
    Topo refresh count: 1
    PE1, Metric: 1, Calc: 0, Prio: 0, FAD supported
    Full SPFs: 68, Partial SPFs: 0
IS-IS Flex Algo 128 level 2 SPF log:
Start time      Elapsed (secs) Count Reason
Wed May 4 09:24:33      0.000158 3 Reconfig
Wed May 4 09:37:34      0.000261 3 Reconfig
Wed May 4 09:38:10      0.000273 3 Reconfig
```

The output shows PE2 as the node advertising FAD and also the winner among the FAD advertising router. Now let's look at the TED outputs of PE2 on the router PE3:

```
regress@PE3# run show ted database topology-type l3-unicast PE2 extensive
TED database: 8 ISIS nodes 8 INET nodes 8 INET6 nodes
NodeID: PE2.00(2.2.2.2)(2001:db8:2:255:2::2)
  Type: Rtr, Age: 410 secs, LinkIn: 2, LinkOut: 2
  Protocol: IS-IS(2)
  2.2.2.2, 128.207.207.140
    To: P2.00(4.4.4.4)(2001:db8:4:255:4::4), Local: 2.24.1.1, Remote: 2.24.1.2
    IPv6 Local: 2001:db8:24::1, IPv6 Remote: 2001:db8:24::2
    Local interface index: 364, Remote interface index: 361
    Color: 0 <none>
    Metric: 1
    IGP metric: 1
    Average delay: 1
    Minimum delay: 1
    Maximum delay: 1
    Delay variation: 0
    Static BW: 1000Mbps
    Interface Switching Capability Descriptor(1):
      Switching type: Packet
      Encoding type: Packet
      Maximum LSP BW [priority] bps:
        [0] 0bps      [1] 0bps      [2] 0bps      [3] 0bps
        [4] 0bps      [5] 0bps      [6] 0bps      [7] 0bps
      P2P Adjacency-SID:
        IPV6, SID: 299904, Flags: 0xb0, Weight: 0
        IPV4, SID: 299888, Flags: 0x30, Weight: 0
        SRV6 End.X SID:
```

```

P2P SRV6 End-X-SID:, SID: 2001:db8:b1:2:224::, Flags: --P, Weight: 0, Algorithm: 128,
Flavor: PSP, USP, USD
P2P SRV6 End-X-SID:, SID: 2001:db8:a1:2:223::, Flags: --P, Weight: 0, Algorithm: 0, Flavor:
PSP, USD
To: P1.00(3.3.3.3)(2001:db8:3:255:3::3), Local: 2.23.1.1, Remote: 2.23.1.2
IPv6 Local: 2001:db8:23::1, IPv6 Remote: 2001:db8:23::2
Local interface index: 363, Remote interface index: 361
Color: 0 <none>
Metric: 1
IGP metric: 1
Average delay: 1
Minimum delay: 1
Maximum delay: 1
Delay variation: 0
Static BW: 1000Mbps
Interface Switching Capability Descriptor(1):
Switching type: Packet
Encoding type: Packet
Maximum LSP BW [priority] bps:
[0] 0bps      [1] 0bps      [2] 0bps      [3] 0bps
[4] 0bps      [5] 0bps      [6] 0bps      [7] 0bps
P2P Adjacency-SID:
IPV6, SID: 299936, Flags: 0xb0, Weight: 0
IPV4, SID: 299920, Flags: 0x30, Weight: 0
SRV6 End.X SID:
P2P SRV6 End-X-SID:, SID: 2001:db8:b1:2:223::, Flags: --P, Weight: 0, Algorithm: 128,
Flavor: PSP, USP, USD
P2P SRV6 End-X-SID:, SID: 2001:db8:a1:2:224::, Flags: --P, Weight: 0, Algorithm: 0, Flavor:
PSP, USD
Prefixes:
2.2.2.2/32
IGP Flags: 0x00
128.207.207.140/32
IGP Flags: 0x00
2001:d00::/24
Metric: 1, IGP Flags: 0x00
2001:db8:2:255:2::128
IGP Flags: 0x00
2001:db8:23::/64
Metric: 1, IGP Flags: 0x00
2001:db8:a1:2::/64
IGP Flags: 0x00
SRV6 Locators:
2001:db8:a1:2::/64
2001:db8:a1:2:1:1::, 0x00, USD
2001:db8:a1:2:1:2::, 0x00, PSP
2001:db8:b1:2::/64
2001:db8:b1:2:11::, 0x00, USD
2001:db8:b1:2:12::, 0x00, PSP
SPRING-Algorithms:
Algo: 0
Algo: 1
Algo: 128
SPRING Flex-Algorithms:
Flex-Algo: 128
Metric: 1, Calc: 0, Prio: 0
Flags: 0x00

```

```
SRV6 Capable: - Flags: 0
SRV6 Node MSD:
  - Type: 41, Value: 6
  - Type: 42, Value: 7
  - Type: 43, Value: 5
  - Type: 44, Value: 6
  - Type: 45, Value: 6
```

You can see that the PE2 router extensive TED database output lists the SRv6 End.X SIDs for flex-algo 128, End SIDs for flex-algo 128 and also algorithm advertisement as well as FAD advertisement.

Let's also verify the lsdist.0 table for updated flex algo information:

```
regress@PE3# run show route table lsdist.0 extensive
lsdist.0: 136 destinations, 136 routes (136 active, 0 holddown, 0 hidden)
NODE { AS:64512 ISO:0019.2168.1000.00 ISIS-L2:0 }/1216 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group ibgp type Internal) Type 1 val 0x94ef078 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [64512] I
    Communities:
      Advertise: 00000001
Path NODE { AS:64512 ISO:0019.2168.1000.00 ISIS-L2:0 }
Vector len 4. Val: 0
  *IS-IS Preference: 18
  Level: 2
  Next hop type: Fictitious, Next hop index: 0
  Address: 0x747c634
  Next-hop reference count: 244, key opaque handle: 0x0, non-key opaque handle: 0x0
  Next hop:
    State: <Active NotInstall>
    Local AS: 64512
    Age: 16:24
    Validation State: unverified
    Task: IS-IS
    Announcement bits (1): 1-BGP_RT_Background
    AS path: I
    IPv4 Router-ids:
      1.1.1.1
    Area border router: No
    External router: No
    Attached: No
    Overload: No
    Hostname: PE1
    Area membership:
      47 00 05
      47 00 05 80 ff f8 00 00 00 01 08 00 01
    SPRING-Algorithms:
      - Algo: 0
      - Algo: 1
      - Algo: 128
    SPRING Flex-Algorithms Definition:
      - Flex-Algo: 128
```

```

Metric: 1, Calc: 0, Priority: 0
  - Flags: 0x00
SRV6 Capable: - Flags: 0
SRV6 Node MSD:
  - Type: 41, Value: 6
  - Type: 42, Value: 7
  - Type: 43, Value: 5
  - Type: 44, Value: 6
  - Type: 45, Value: 6
Thread: junos-main
LINK { Local { AS:64512 ISO:0019.2168.1000.00 }.{ IPv4:1.13.1.1 } Remote { AS:64512
ISO:0019.2168.3000.00 }.{ IPv4:1.13.1.2 } ISIS-L2:0 }/1216 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group ibgp type Internal) Type 1 val 0x94eef44 (adv_entry)
  Advertised metrics:
    Flags: Nexthop Change
    Nexthop: Self
    Localpref: 100
    AS path: [64512] I
    Communities:
    Advertise: 00000001
Path LINK { Local { AS:64512 ISO:0019.2168.1000.00 }.{ IPv4:1.13.1.1 } Remote { AS:64512
ISO:0019.2168.3000.00 }.{ IPv4:1.13.1.2 } ISIS-L2:0 }
Vector len 4. Val: 0
  *IS-IS Preference: 18
  Level: 2
  Next hop type: Fictitious, Next hop index: 0
  Address: 0x747c634
  Next-hop reference count: 244, key opaque handle: 0x0, non-key opaque handle: 0x0
  Next hop:
    State: <Active NotInstall>
    Local AS: 64512
    Age: 18:12
    Validation State: unverified
    Task: IS-IS
    Announcement bits (1): 1-BGP_RT_Background
    AS path: I
    Color: 0
    Maximum bandwidth: 1000Mbps
    Metric: 1
    TE Metric: 1
    Average delay: 1
    Minimum delay: 1
    Maximum delay: 1
    Delay variation: 0
    Label: 299936, Flags: 0xb0, Weight: 0
    Label: 299920, Flags: 0x30, Weight: 0
    P2P SRV6 End.X SID: SID: 2001:db8:b1:1:213::, Flags: --P, Weight: 0, Algorithm: 128, Flavor:
PSP, USP, USD
    P2P SRV6 End.X SID: SID: 2001:db8:a1:1:213::, Flags: --P, Weight: 0, Algorithm: 0, Flavor:
PSP, USD
    Thread: junos-main
...
SRV6 End SID { Node { AS:64512 ISO:0019.2168.2000.00 } { End SID:2001:db8:a1:2:11:: } ISIS-L2:0 }/1216
  *[IS-IS/18] 00:19:22
  Fictitious
SRV6 End SID { Node { AS:64512 ISO:0019.2168.2000.00 } { End SID:2001:db8:a1:2:12:: } ISIS-L2:0 }/1216

```

```
*[IS-IS/18] 00:19:22
  Fictitious
SRV6 End SID { Node { AS:64512 ISO:0019.2168.2000.00 } { End SID:2001:db8:b1:2:11:: } ISIS-L2:0 }/1216

*[IS-IS/18] 00:19:22
  Fictitious
SRV6 End SID { Node { AS:64512 ISO:0019.2168.2000.00 } { End SID:2001:db8:b1:2:12:: } ISIS-L2:0 }/1216

*[IS-IS/18] 00:19:22
  Fictitious
```

Note that the SRv6 flex algo End.X SIDs as well as new NLRIIs for flex algo End SIDs are being advertised via BGP-LS.

Chapter 8

IPv6 Only Networks

8.1 Introduction

SRv6 technology is a purely IPv6 technology. The network does not need to have any IPv4 address for SRv6 to run. IPv4 addresses are precious resources and there may be a need to remove the IPv4 addresses if not needed. In previous chapters we have enabled the network with SRv6 and switched the services to use SRv6 tunnels. In this chapter, let's remove LDP from core and also release the IPv4 addresses. Note that Junos does not support BGP-LS family on a pure IPv6 BGP session.

8.1.1 Remove LDP From Core

In preparation to remove LDP from the network, an operator should ensure the traffic is completely drained out from LDP. This can be verified by checking `ldp` statistics or telemetry streaming of counter data for the LSP labels.

Run this on all routers:

```
delete protocols ldp
delete protocols mpls
```

8.1.2 Remove All IPv4 Addresses and Remove eBGP and iBGP v4 Sessions

Since we are migrating towards an IPv6 only network we can now get rid of our eBGP and iBGP v4 sessions. Also, we can now remove all IPv4 addresses in our network, except the IPv4 addresses on CE to PE links. We will keep those to showcase IPv4 services over IPv6-only network.

For example, in PE1:

```
delete interfaces ge-0/0/2 unit 0 family inet
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
delete routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv4
delete logical-systems CE1 protocols bgp group eBGPv4
delete protocols bgp group ibgp
```

Now we need to carry this exercise on all routers. After this step only IPv6 iBGP and eBGP sessions will be left in our network. For example, on PE1:

```
[edit]
root@PE1# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 4 Down peers: 0
Table      Tot Paths  Act Paths Suppressed History Damp State Pending
inet.0
          0        0        0        0        0        0
bgp.l3vpn.0
          6        3        0        0        0        0
bgp.l3vpn-inet6.0
          9        3        0        0        0        0
inet6.0
          0        0        0        0        0        0
Peer          AS  InPkt  OutPkt  OutQ  Flaps Last Up/Dwn State|#Active/Received/Accepted/
Damped...
2001:db8:2:255:2::2      64512    44     43     0     1   15:48 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 1/2/2/0
  bgp.l3vpn-inet6.0: 1/3/3/0
  L3VPN_PE1_CE1.inet.0: 1/2/2/0
  L3VPN_PE1_CE1.inet6.0: 1/3/3/0
  inet6.0: 0/0/0/0
2001:db8:7:255:7::7      64512    44     43     0     1   15:43 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 1/2/2/0
  bgp.l3vpn-inet6.0: 1/3/3/0
  L3VPN_PE1_CE1.inet.0: 1/2/2/0
  L3VPN_PE1_CE1.inet6.0: 1/3/3/0
  inet6.0: 0/0/0/0
2001:db8:8:255:8::8      64512    43     42     0     1   15:38 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 1/2/2/0
  bgp.l3vpn-inet6.0: 1/3/3/0
  L3VPN_PE1_CE1.inet.0: 1/2/2/0
  L3VPN_PE1_CE1.inet6.0: 1/3/3/0
  inet6.0: 0/0/0/0
2001:db8:11::1      100      58     59     0     0   23:44 Establ
  L3VPN_PE1_CE1.inet.0: 1/2/2/0
  L3VPN_PE1_CE1.inet6.0: 2/3/3/0
```

Our ISIS adjacency also shows that no IPv4 adjacencies are present:

```
root@PE1# run show isis adjacency detail
P1
  Interface: ge-0/0/2.0, Level: 2, State: Up, Expires in 26 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 00:48:10 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
IPv6 addresses: fe80::5603:7ff:fe00:3ff0
IPv6 Global Interface Address: 2001:db8:13::2
  Level 2 SRv6 unprotected End-X-SID: 2001:db8:a1:1:213::
    Flavor: None, Flags: --P, Algorithm: 0

P2
  Interface: ge-0/0/3.0, Level: 2, State: Up, Expires in 21 secs
  Priority: 0, Up/Down transitions: 1, Last transition: 00:47:55 ago
  Circuit type: 2, Speaks: IP, IPv6
  Topologies: Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
IPv6 addresses: fe80::5603:7ff:fe00:3fe4
IPv6 Global Interface Address: 2001:db8:14::2
  Level 2 SRv6 unprotected End-X-SID: 2001:db8:a1:1:214::
    Flavor: None, Flags: --P, Algorithm: 0
```

Chapter 9

Micro-loop Avoidance

9.1 Introduction

When there is a change in network topology due either to a link going down or a new link addition, all the nodes in the network need to get the complete view of the network and re-compute the routes. There will generally be a small time window when the forwarding state of each of the nodes is not synchronized. This can result in transient loops in the network, leading to dropped traffic due to over-subscription of links or due to TTL expiry. Micro-looping is generally more harmful than simply dropping traffic on failed links because it can cause control traffic to be dropped due to congestion on an otherwise healthy link involved in micro-loop.

[Draft-bashandy-rtgwg-segment-routing-uloop](#) provides a solution to avoid micro-loops during IGP convergence. The post convergence path is computed consisting of a SID stack. Each SID in the stack represents an anchor point in the path that is used to avoid micro-loops. The micro-loop avoidance path consisting of the SID stack is used for the post convergence path for a time interval called the *micro-loop avoidance delay*. Micro-loop avoidance delay is a configurable timer and varies from network to network based on the device convergence capability as well as the scale of the network. The micro-loop scenario is called local micro-loop when the micro-loop involves the node that is point of local repair (PLR) for the failure. The micro-loop scenario is called *remote micro-loop* when the micro-loop is formed on remote nodes, not involving the node on which the failure occurred.

The Junos micro-loop avoidance solution provides protection from both local and remote micro-loops. The failure events such as link up/down, node up/down, and metric change can cause micro-loops. The micro-loop avoidance procedures work well for single failure in the network.

If there are multiple failures\ within the time interval of micro-loop avoidance delay, then the micro-loop avoidance procedures are aborted and normal convergence procedures are followed.

Junos support for various failure events and the multiple failure events is described in detail in Section 9.7.

9.2 Micro-loop Scenario

Let's create a micro-loop scenario in the lab setup and then configure the micro-loop avoidance to prevent the micro-loop. The topology is shown in Figure 9.1.

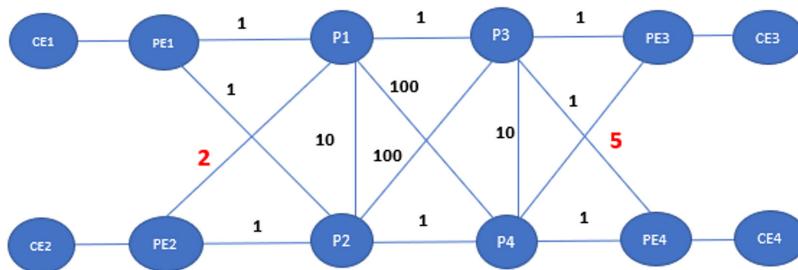


Figure 9.1 Topology for M-loop Avoidance

We modified the link metrics on a few links to create a micro-loop scenario in the lab set-up. The link metric on the P4 to PE3 is set to 5 and the link metric from PE2 to P1 is set to 2.

All metrics on the links are symmetric and the metric change was done on each link in both directions. Micro-loops happen in the network due to FIB convergence delay on different nodes.

On a virtual setup, all nodes have the same processing capability and same convergence speed and it is very hard to create micro-loops. Therefore, we set the SPF trigger delay on PE1 to 1 second. During any failure event, SPF will be triggered on PE1 after 1 second from the time the event was received. This simulates slow convergence on PE1 compared to other nodes in the network and hence creates micro-loops:

```
regress@PE1# set protocols isis spf-options delay 1000
```

We use the ping inet6 facility available on the router to send traffic. The ping utility sends traffic at a very slow rate and we may not be able to see traffic loss due to micro-loop in this virtual setup.

So we configure a firewall filter to detect if the micro-loops are seen. In our setup, we use PE3 End-SID address as the destination address for the ping traffic. We originate ping packets from PE2 destined to 2001:db8:a3:1:1:1::0. On PE1, we configure an output

firewall filter which counts the traffic destined to 2001:db8:a3:1:11::0 and is applied on the PE1 to P1 interface. In normal scenarios we should not see any packet destined to 2001:db8:a3:1:11::0 on the PE1->P1 interface as we are originating ping packets from PE1. In case of micro-loop between PE1 and P1, we can see traffic on PE1->P1 interface:

```
set firewall family inet6 filter filter1 term 1 from destination-address 2001:db8:a3:1:11::0/128
set firewall family inet6 filter filter1 term 1 then count cnt
set firewall family inet6 filter filter1 term 1 then accept
set firewall family inet6 filter filter1 term 2 then accept
set interfaces ge-0/0/2 unit 0 family inet6 filter output filter1'
```

9.2.1 Local Micro-loop

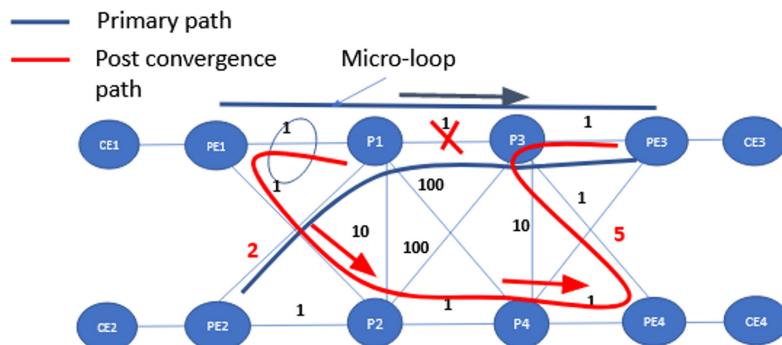


Figure 9.2 Local M-loop

Figure 9.2 illustrates a local micro-loop scenario. The ping traffic sent from PE2 goes via PE2->P1->P3->PE3. When the P1->P3 link goes down, the post convergence path is P1->PE1->P2->P4->PE3. Pre-failure PE1 points toward P1 for the destination PE3. As PE1 is made to converge slower we can see micro-loop between PE1 and P1. This is a local micro-loop as the PLR itself is involved in the micro-loop.

Let's start the ping traffic on PE2 towards destination 2001:db8:a3:1:11::0/128:

```
regress@PE2# run ping rapid inet6 2001:db8:A3:1:11::0 count 1000
PING6(56=40+8+8 bytes) 2001:db8:23::1 --> 2001:db8:a3:1:11::
**
--- 2001:db8:A3:1:11::0 ping6 statistics ---
1000 packets transmitted, 999 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 3.276/4.544/39.990/2.748 ms
```

You can see that when the network is stable there is no traffic loss and all packets successfully reach PE3.

Now let's bring down the interface P1->P3:

```
regress@P1# set interfaces ge-0/0/1.0 disable
regress@P1# commit
commit complete
```

On PE1, you can observe that there are outgoing packets on the PE1->P1 link. These packets are looping between PE1 and P1:

```
regress@PE1# run show firewall filter filter1 counter cnt
Filter: fil1
Counters:
Name          Bytes      Packets
cnt          1736        31
```

9.2.2 Remote Micro-loop

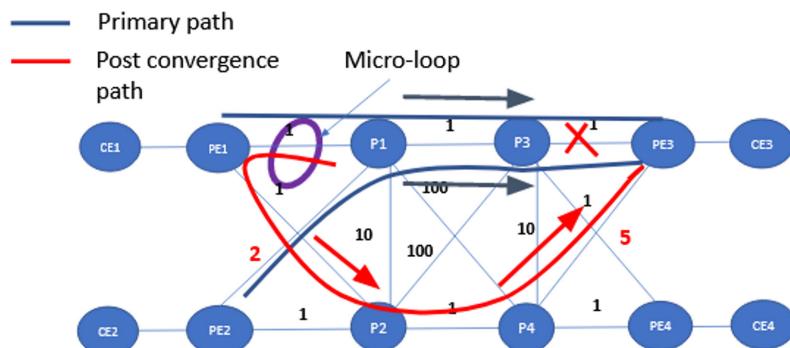


Figure 9.3 Remote Micro-loop

Figure 9.3 illustrates a remote micro-loop scenario. The ping traffic sent from PE2 goes via PE2->P1->P3->PE3. When the P3->PE3 link goes down, the post convergence path is P3->P1->PE1->P2->P4->PE4->P3->PE3. Pre-failure PE1 points toward P1 for the destination PE3. As PE1 is made to converge slower we can see the micro-loop between PE1 and P1. This is a remote micro-loop as the PLR itself is not affected in the micro-loop. Let's start the ping traffic on PE2 towards destination 2001:db8:a3:1:11::0/128:

```
regress@PE2# run ping rapid inet6 2001:db8:A3:1:11::0 count 1000
PING6(56=40+8+8 bytes) 2001:db8:23::1 --> 2001:db8:a3:1:11::
```
--- 2001:db8:A3:1:11::0 ping6 statistics ---
1000 packets transmitted, 999 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 3.276/4.544/39.990/2.748 ms
```

Let's bring down the interface P3->PE3:

```
regress@P3# set interfaces ge-0/0/3.0 disable
regress@P3# commit
commit complete
```

We can check the firewall filter to check the packets destined to 2001:db8:a3:1:11::0/128 are being seen on PE1->P1 interface in outgoing direction which are looping between PE1 and P1:

```
egress@PE1# run show firewall filter fil1 counter cnt
Filter: fil1
Counters:
Name Bytes Packets
cnt 2184 39
```

## 9.3 Enabling Micro-loop Avoidance

The micro-loop avoidance feature uses local computations and programs the non-looping path into the forwarding plane. Micro-loop avoidance features can be selectively enabled on the nodes that support the feature. The best outcome of avoiding micro-loops is possible only when all nodes in the network support the feature and have been enabled. Partial support in the network would still provide the benefit for the traffic that is subjected to a micro-loop and passing through the particular node on which the feature is enabled.

Micro-loop avoidance feature requires a delay timer to be configured. The route would use the micro-loop avoiding paths until the delay timer expires. Once the delay timer expires the converged paths would be programmed into the forwarding plane. The delay timer configured on every node in the network should be same. The delay timer value should be based on the time the slowest router in the IGP domain takes to completely download all the IGP prefixes into the forwarding plane.

Let's enable micro-loop avoidance on all nodes and verify that the failures demonstrated in Section 9.2.1 and Section 9.2.2 do not cause micro-loops.

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

### P1->P3 link failure

Let's start the ping traffic from PE2 and fail the P1->P3 link and verify there is no micro-loop:

```
regress@P1# set interfaces ge-0/0/1.0 disable
regress@P1# commit
commit complete
```

You can see that there are no packets destined to 2001:db8:a3:1:11::0/128 on the PE1->P1 link in outgoing direction which implies that the micro-loop has been avoided:

```
regress@PE1# run show firewall filter fil1 counter cnt
Filter: fil1
Counters:
Name Bytes Packets
cnt 0 0
```

### P3->PE3 link failure

Now let's start the ping traffic from PE2 and fail the P3->PE3 link and verify there is no micro-loop:

```
regress@P3# set interfaces ge-0/0/3.0 disable
[edit]
regress@P3# commit
commit complete

regress@PE1# run show firewall filter fil1 counter cnt
Filter: fil1
Counters:
Name Bytes Packets
cnt 0 0
```

## 9.4 Micro-loop Avoiding Paths

This section reviews how the micro-loop avoiding paths are computed. Micro-loop avoidance paths select one or more anchor nodes based on the topology. A node which has the same pre-failure path and post failure path is selected as an anchor node. In other words, the path to the anchor node is unaffected by the failure. See Figure 9.4.

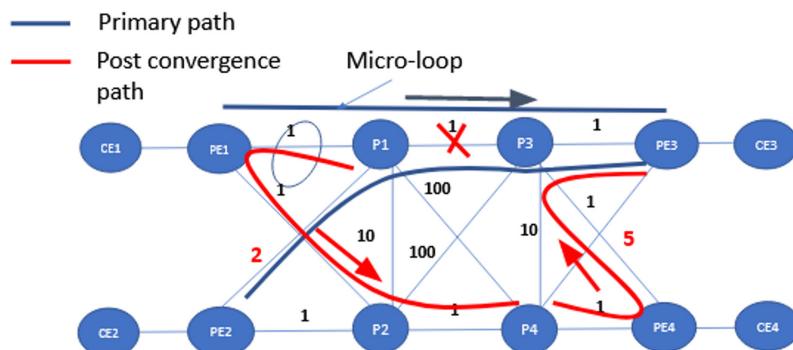


Figure 9.4 Computation of Micro-loop Avoiding Paths

In Figure 9.3, the failure of P1->P3 link is illustrated. Path to P4 from P1 is P1->PE1->P2->P4. This path is unaffected by the failure and remains the same even after failure.

P4 is chosen as the anchor node and is the top SID in the segment list. In this topology and for this prefix there is only one anchor node. From P4 to PE3 the path is P4->PE4->P3->PE3 and this path is unaffected by failure. The post convergence path for the failure consists of one SID which is P4's End SID `2001:db8:a2:4:11::`.

The post convergence path can be observed on P1 during convergence for the duration of configured micro-loop delay time. It's a very short-lived route and next hops can be observed with a `show route <x> extensive` command:

```

regress@P1# run show route 2001:db8:A3:1:11::0 extensive
inet6.0: 49 destinations, 49 routes (49 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
 State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {Chain 0x710c57c 691}
IS-IS level 2, SRV6 Loc Sid 0x88dc5f8
 *IS-IS Preference: 18
 Level: 2
 Next hop type: Chain, Next hop index: 691
 Address: 0x710c57c
 Next-hop reference count: 2
 Next hop: ELNH Address 0x710c98c, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1::
 Segment-list[0] 2001:db8:a2:4:11::
 Next hop type: Router, Next hop index: 660
 Address: 0x710c98c
 Next-hop reference count: 54
 Next hop: fe80::5603:7ff:fe00:4097 via ge-0/0/3.0
 State: <Active Int>
 Age: 3 Metric: 6
 Validation State: unverified
 ORR Generation-ID: 0
 Task: IS-IS
 Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
 AS path: I
 Thread: junos-main

inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)

2001:db8:a3:1:11::/128 (1 entry, 1 announced)
 *SRV6-ISIS Preference: 14
 Level: 2
 Next hop type: Chain, Next hop index: 0
 Address: 0x7108414
 Next-hop reference count: 1
 Next hop: ELNH Address 0x710c98c, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1:11::
 Segment-list[0] 2001:db8:a2:4:11::
 Segment-list[1] 2001:db8:a3:1:11::
 Next hop type: Router, Next hop index: 660
 Address: 0x710c98c
 Next-hop reference count: 54
 Next hop: fe80::5603:7ff:fe00:4097 via ge-0/0/3.0
 State: <Active NoReadvrt Int>
 Age: 3 Metric: 6
 Validation State: unverified
 ORR Generation-ID: 0
 Task: IS-IS
 Announcement bits (1): 1-Resolve tree 2
 AS path: I
 Thread: junos-main

```

Note that the segment lists in inet.0 and inet.3 look slightly different. The Inet.0 table stores IP routes and these routes are directly downloaded into the forwarding plane.

Traffic that is directly being sent to the address `2001:db8:a3:1:11::0` will use this route. Additional tunnel encapsulation is not applicable to these routes and hence you can see that just the anchor node's IP address is seen in the next hop. Inet.3 routes are tunnel routes which do not directly get downloaded in the forwarding plane. Some service routes resolves on these tunnel routes and as part of the service routes these next hops get downloaded into the forwarding plane. The service prefixes that resolve on these inet.3 routes will have an additional IPv6 encapsulation header and the segment list will be part of that encapsulation header. Tunnel routes in inet.3 always contain the last destination SID also in the segment list which gets used as a destination address in the last segment of the path. If reduced mode of operation is used, the last SID gets replaced by the service SID.

Let's consider another example in Figure 9.5 and observe the micro-loop avoidance path.

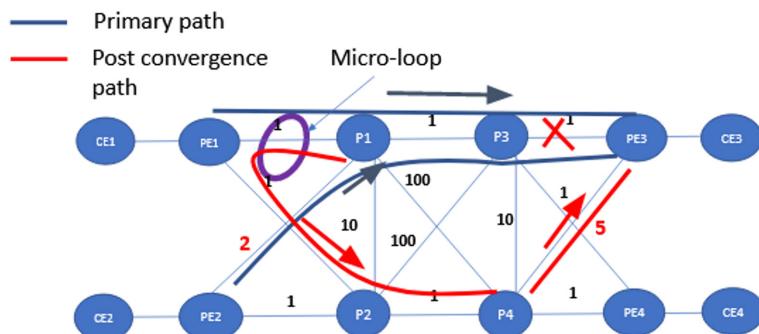


Figure 9.5 Computation of Remote M-loop Avoidance Path

In this example, let's review P3->PE3 link failure. The post-convergence path is P1->PE1->P2->P4->PE3. P4 is the anchor node and an additional End.x SID for the link P4->PE3 is used to get the micro-loop avoiding path. Let's look at the `show route` outputs on P1 towards PE3 on our lab setup. Route to `2001:db8:a3:1:11::0` as seen before the failure:

```
regress@P1# run show route 2001:db8:a3:1:11::0 extensive
inet6.0: 48 destinations, 48 routes (48 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
 State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {fe80::5603:7ff:fe00:40ca}
IS-IS level 2, SRv6 Loc Sid 0x933397c
 *IS-IS Preference: 18
 Level: 2
 Next hop type: Router, Next hop index: 675
 Address: 0x7107164
 Next-hop reference count: 24
 Next hop: fe80::5603:7ff:fe00:40ca via ge-0/0/1.0, selected
 Session Id: 0x14b
```

```

State: <Active Int>
Age: 2:01 Metric: 2
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
AS path: I
Thread: junos-main

```

Note that the route has a plain route next hop and forwarding traffic towards P3.

```
regress@P1# set interfaces ge-0/0/1.0 disable
```

```
regress@P1# commit
commit complete
```

Observe the route during micro-loop avoidance delay duration after the interface has been brought down. The next hop contains additional SIDs which help avoid the micro-loops during the convergence:

```

regress@P1# run show route 2001:db8:a3:1:11::0 extensive
inet6.0: 47 destinations, 47 routes (47 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
 State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {Chain 0x710b87c 689}
IS-IS level 2, SRV6 Loc Sid 0x933397c
 *IS-IS Preference: 18
 Level: 2
 Next hop type: Chain, Next hop index: 689
 Address: 0x710b87c
 Next-hop reference count: 2
 Next hop: ELNH Address 0x710c98c, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1::
 Segment-list[0] 2001:db8:a2:4:11::
 Segment-list[1] 2001:db8:a2:4:267::0
 Next hop type: Router, Next hop index: 660
 Address: 0x710c98c
 Next-hop reference count: 48
 Next hop: fe80::5603:7ff:fe00:4097 via ge-0/0/3.0
 State: <Active Int>
 Age: 2 Metric: 8
 Validation State: unverified
 ORR Generation-ID: 0
 Task: IS-IS
 Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
 AS path: I
 Thread: junos-main

inet6.3: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)

2001:db8:a3:1:11::/128 (1 entry, 1 announced)
 *SRV6-ISIS Preference: 14
 Level: 2
 Next hop type: Chain, Next hop index: 0
 Address: 0x710b5a4
 Next-hop reference count: 1
 Next hop: ELNH Address 0x710c98c, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode

```

```

Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1:11::
Segment-list[0] 2001:db8:a2:4:11::
Segment-list[1] 2001:db8:a2:4:267:0
Segment-list[2] 2001:db8:a3:1:11::
 Next hop type: Router, Next hop index: 660
 Address: 0x710c98c
 Next-hop reference count: 48
 Next hop: fe80::5603:7ff:fe00:4097 via ge-0/0/3.0
State: <Active NoReadvrt Int>
Age: 2 Metric: 8
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
Announcement bits (1): 1-Resolve tree 2
AS path: I
Thread: junos-main

```

## 9.5 Insert Mode and Encap Mode

The micro-loop avoidance feature can be deployed in two different modes.

The default mode of operation is encapsulation mode. In this mode the node that is applying the micro-loop prevention mechanism for the traffic will encapsulate another IPv6 header and include the SRH consisting of anchor nodes SID list as part of the new header. Insert mode does not use additional header and so saves bytes on the wire whereas encap mode uses an additional header but is more compliant with the IPv6 architecture by not inserting the extension headers in the middle of the network (see Figure 9.6).

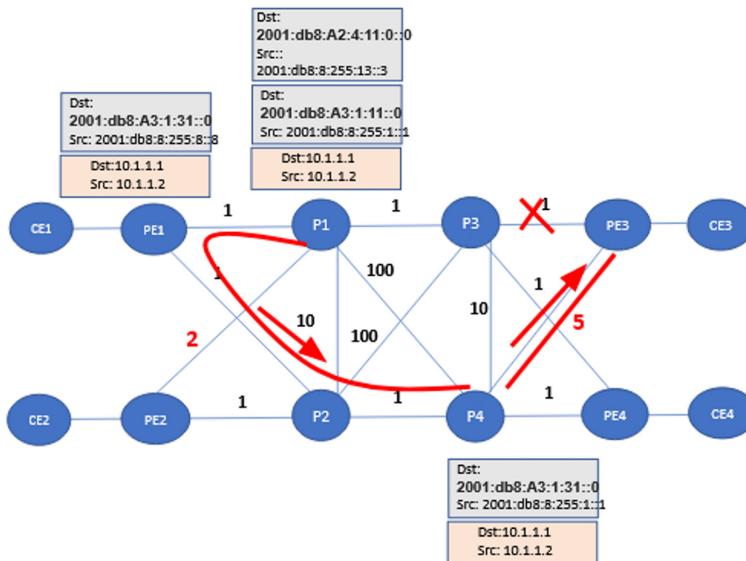


Figure 9.6 Packet Forwarding on M-loop Avoiding Path

Figure 9.6 depicts the packet forwarding details on each of the nodes. Let's consider a service prefix 10.1.1.1 which is advertised with a End.DT4 SID from PE3. The End.DT4 SID is 2001:db8:A3:1:3:1::0. Let's assume the service is using the best effort path from PE1 to PE3. On PE1, the packet gets encapsulated in an IPv6 header. There is no SRH since the traffic is using best effort path. In the reduced-srh mode, the destination address will be the service SID advertised by PE3.

On reaching P1, there is a failure and the traffic uses post-convergence micro-loop avoidance paths. In the encapsulation mode, there is a new header imposed on P1 which uses the P1 loopback address as source address and the P4's End SID as destination address. Note that the End SID of P4 used in the encapsulation is of usd flavor. On reaching P4, the additional encapsulation gets removed and the inner header is exposed and traffic forwarded to PE3.

In the encapsulation mode, it is important that the nodes advertise End and End.X SIDs of flavor usd. If they don't advertise required flavor SIDs, the micro-loop avoidance paths are not created:

```
set routing-options source-packet-routing srv6 transit-srh-insert
```

Okay let's observe the route when insert mode is enabled. Note that the next hop has the insert mode set. When insert mode is set on the next hop, an additional header is not imposed and the segment list is copied into SRH and inserted into the existing IPv6 packet header:

```
regress@P1# run show route 2001:db8:A3:1:11::0 extensive
inet6.0: 49 destinations, 49 routes (49 active, 0 holddown, 0 hidden)
2001:db8:a3:1::/64 (1 entry, 1 announced)
 State: <FlashAll>
TSI:
KRT in-kernel 2001:db8:a3:1::/64 -> {Chain 0x710bc24 691}
IS-IS level 2, SRv6 Loc Sid 0x8e446f4
 *IS-IS Preference: 18
 Level: 2
 Next hop type: Chain, Next hop index: 691
 Address: 0x710bc24
 Next-hop reference count: 2
 Next hop: ELNH Address 0x710c98c, selected
 SRV6-Tunnel: Reduced-SRH Insert-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:a3:1::
 Segment-List[0] 2001:db8:a2:4:12::
 Next hop type: Router, Next hop index: 660
 Address: 0x710c98c
 Next-hop reference count: 54
 Next hop: fe80::5603:7ff:fe00:4097 via ge-0/0/3.0
State: <Active Int>
Age: 2 Metric: 6
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
Announcement bits (3): 0-KRT 1-LDP 2-IS-IS
AS path: I
Thread: junos-main
```

## 9.6 Micro-loops Avoidance in Flex Algo Planes

As described in Chapter 7, flexible -algorithms (flex algo) is a mechanism to create multiple routing planes. Micro-loop avoidance can be applied in a flex algo plane to prevent micro-loops that can happen in a flex algo topology (see Figure 9.7).

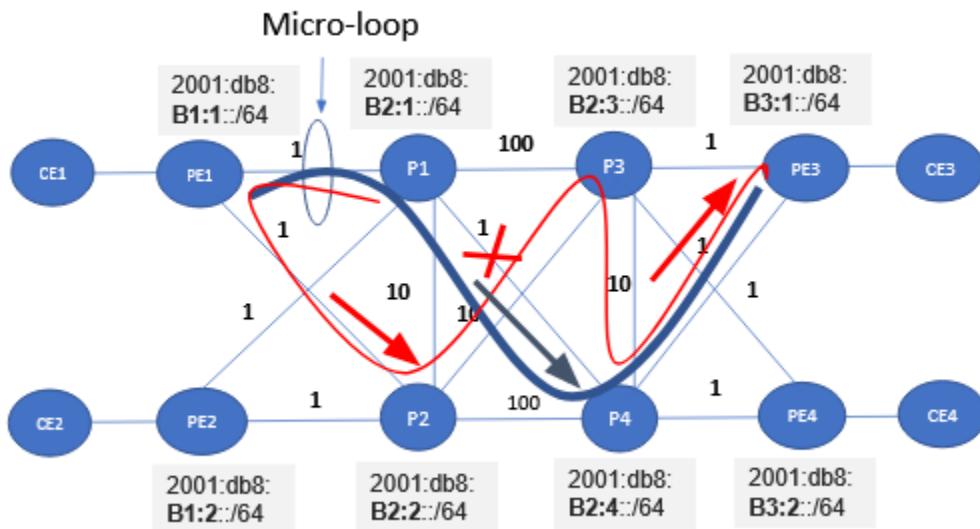


Figure 9.7 M-loop Avoidance in Flex Algo Planes

Let's demonstrate a micro-loop in a flex algo topology. Figure 9.7 shows a flex algo 128 topology. The flex algo definition is based on delay-metric and the numbers above the links in Figure 9.7 represent a delay-metric. The flex algo specific locators have been specified on the nodes in the diagram.

In order to detect the micro-loops, the firewall filter configuration is done on the PE1->P1 interface. The ping traffic for destination 2001:db8:b3:1:11:0/128 is started on PE2. The path for the traffic is PE2->P1->P4->PE3. When the P1->P4 interface is brought down, the traffic follows the path P1->PE1->P2->P3->P4->PE3. As the pre-failure path from PE1 to destination 2001:db8:b3:1:11:0/128 points to P1, there is a micro-loop between P1 and PE1.

You can observe the micro-loop by configuring a firewall filter that counts the outgoing traffic on PE1:

```
regress@PE1# run show configuration firewall |display set
set firewall family inet6 filter filter1 term 2 from destination-address 2001:db8:b3:1:11::0/128
set firewall family inet6 filter filter1 term 2 then count cnt2
set firewall family inet6 filter filter1 term 2 then accept
set firewall family inet6 filter filter1 term 3 then accept
set interfaces ge-0/0/2 unit 0 family inet6 filter output filter1

regress@PE2# run ping rapid inet6 2001:db8:b3:1:11:: count 1000
PING6(56=40+8+8 bytes) 2001:db8:23::1 --> 2001:db8:b3:1:11::
!!
!!
!!
!!
!!
!!
!!
!!
!!
!!
2001:db8:13::1: Time to live exceeded
Vr TC Flow Plen Nxt Hlim
6 00 00000 0010 3a 01
2001:db8:24::1->2001:db8:b3:1:11::
ICMP6: type = 128, code = 0
.!!
!!
!!
--- 2001:db8:b3:1:11::0 ping6 statistics ---
1000 packets transmitted, 999 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 3.424/6.276/73.223/4.584 ms
```

Start the ping traffic and then bring down the interface between P1->P4. Here are the filter counters on PE2 going out on PE1->P1 interface for the destination 2001:db8:b3:1:1:0/128:

```
regress@PE1# run show firewall filter filter1 counter cnt2
Filter: filter1
Counters:
Name Bytes Packets
cnt2 23072 412
```

Enable micro-loop avoidance on P1 and repeat the test. The post-convergence micro-loop avoidance path goes via P1->PE1->P2->P3->PE3.

During the micro-loop avoidance period, you can observe that the route has a SID stack consisting of P2's End-SID and P2->P3 link's End.X SID in the stack. You can also observe the locator route in inet6.0 as well as in the color table junos-rti-tc-128.inet6.3 for the flex algo locator of PE3:

```
regress@P1# run show route 2001:db8:b3:1:11::0 extensive
inet6.0: 63 destinations, 63 routes (63 active, 0 holddown, 0 hidden)
2001:db8:b3:1::/64 (1 entry, 1 announced)
TSI:
KRT in-kernel 2001:db8:b3:1::/64 -> {Chain 0x7112c44 661}
IS-IS level 2, SRv6 Loc Sid 0x89c1de4
```

```

*IS-IS Preference: 18
Level: 2
Downbit: 256
Next hop type: List, Next hop index: 0
Address: 0x9006d94
Next-hop reference count: 1
Next hop: ELNH Address 0x7112c44 weight 0x1, selected
 Next hop type: Chain, Next hop index: 661
 Address: 0x7112c44
 Next-hop reference count: 2
 Next hop: ELNH Address 0x71127cc
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:b3:1:::
 Segment-list[0] 2001:db8:b2:2:11:::
 Segment-list[1] 2001:db8:b2:2:245::0
 Next hop type: Router, Next hop index: 657
 Address: 0x71127cc
 Next-hop reference count: 47
 Next hop: fe80::5603:7ff:fe00:6081 via ge-0/0/3.0 weight 0x1
Next hop: ELNH Address 0x7107164 weight 0x1
 Next hop type: Chain, Next hop index: 0
 Address: 0x7107164
 Next-hop reference count: 1
 Next hop: ELNH Address 0x7112764
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:b3:1:::
 Segment-list[0] 2001:db8:b2:2:11:::
 Segment-list[1] 2001:db8:b2:2:245::0
 Next hop type: Router, Next hop index: 656
 Address: 0x7112764
 Next-hop reference count: 25
 Next hop: fe80::5603:7ff:fe00:6a96 via ge-0/0/4.0 weight 0x1
State: <Active Int>
Age: 1 Metric: 13
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
Announcement bits (2): 0-KRT 2-IS-IS
AS path: I
Thread: junos-main

junos-rti-tc-128.inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)

2001:db8:b3:1:11::/128 (1 entry, 0 announced)
*SRV6-ISIS Preference: 14
Level: 2
Next hop type: List, Next hop index: 0
Address: 0x9006a34
Next-hop reference count: 2
Next hop: ELNH Address 0x71095f4 weight 0x1, selected
 Next hop type: Chain, Next hop index: 0
 Address: 0x71095f4
 Next-hop reference count: 1
 Next hop: ELNH Address 0x71127cc
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:b3:1:11:::
 Segment-list[0] 2001:db8:b2:2:11:::
 Segment-list[1] 2001:db8:b2:2:245::0

```

```

Segment-list[2] 2001:db8:b3:1:11::
 Next hop type: Router, Next hop index: 657
 Address: 0x71127cc
 Next-hop reference count: 47
 Next hop: fe80::5603:7ff:fe00:6081 via ge-0/0/3.0 weight 0x1
Next hop: ELNH Address 0x7107d94 weight 0x1
 Next hop type: Chain, Next hop index: 0
 Address: 0x7107d94
 Next-hop reference count: 1
 Next hop: ELNH Address 0x7112764
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:3:255:3::3 Dest: 2001:db8:b3:1:11::
Segment-list[0] 2001:db8:b2:2:11::
Segment-list[1] 2001:db8:b2:2:245::0
Segment-list[2] 2001:db8:b3:1:11::
 Next hop type: Router, Next hop index: 656
 Address: 0x7112764
 Next-hop reference count: 25
 Next hop: fe80::5603:7ff:fe00:6a96 via ge-0/0/4.0 weight 0x1
State: <Secondary Active NoReadvrt Int>
Age: 1 Metric: 13
Validation State: unverified
ORR Generation-ID: 0
Task: IS-IS
AS path: I
Primary Routing Table: inet6color.0
Thread: junos-main

```

We can observe the filter counters to verify that there has been no out packets on the PE1->Pl link for the destination 2001:db8:b3:1:11::0/128:

```

regress@PE1# run show firewall filter filter1 counter cnt2
Filter: filter1
Counters:
Name Bytes Packets
cnt2 0 0

```

## 9.7 Supported Events

Micro-loops can be caused by a number of network events:

- Link up/down
- Node down/up
- Metric increase/decrease

Junos implementation supports micro-loop avoidance for link down/link up as well as metric change event. Node down event is recognized as a series of link-down events and node-up scenario is recognized as a series of link-up events. Micro-loop avoidance for node-down and node-up events are supported only when the subsequent link-down/link-up events do not impact the same prefixes. In case of simple topologies such as ring topologies and node-down, the series of link-down events do not impact the prefixes and hence will continue to hold the micro-loop avoidance paths.

## 9.7.1 Multiple Events

When a network experiences an event that changes the topology of the network, the nodes in the network computes micro-loop avoiding path for the prefixes that were impacted by the network event. The micro-loop avoidance path will stay for a small duration which is called *micro-loop avoidance delay timer*. When the micro-loop avoidance delay timer is running, if another network event occurs it is expensive to do computations for micro-loop avoidance that accounts for various states of the network devices related to the multiple network events. In such a scenario, micro-loop avoidance is aborted and normal an IGP post convergence path is programmed.

Junos implementation allows continuing micro-loop avoidance paths for prefixes that are not impacted by multiple network events. Junos behavior can be best explained with an example.

Let's assume A1, A2, A3 and B1, B2, B3 are prefixes in a network. N1 is a network event. N2 is another network event that occurs consecutively during the micro-loop delay time window. Let's us assume, that A1, A2, A3 impacted by N1. There are two scenarios for network event N2.

*Scenario 1:* A1, A2, A3 and B1, B2, B3 are impacted by N2. In this case, for A1, A2, A3 prefixes micro-loop avoidance path is aborted while B1, B2, B3 prefixes also will not have any micro-loop avoidance path. The micro-loop computation is not done for N2 event as it occurred within delay time.

*Scenario 2:* B1, B2, B3 are impacted by N2. In this case the A1, A2, A3 prefixes will continue to have micro-loop avoidance path. B1, B2, B3 prefixes will not have any micro-loop avoidance path. The micro-loop computation is not done for N2 event as it occurred within delay time.

## 9.8 Troubleshooting Micro-loop Avoidance Feature

Micro-loop avoidance paths exist for a very short time and are transient in nature. It is quite difficult to troubleshoot problems related to the micro-loop avoidance feature. One easy approach is to observe the TI-LFA backup path for the destination on the node. The failure impacting the destination will have the exact similar micro-loop avoidance path as the TI-LFA backup path before failure. The exact same anchor End SIDs and End.X SIDs that show up in TI-LFA backup path will also be seen in micro-loop avoidance path.

The most common problems associated with TI-LFA backup path would also be the reasons for missing micro-loop avoidance. The appropriate flavors have to be associated with the End SID and End.X SID for them to be used in TI-LFA/micro-loop avoidance. The MX and ACX router is capable of simultaneously supporting all flavors on the same SID and it is recommended to enable all flavors on a single SID for MX.

The ability to support all flavors on a single SID is a hardware capability and note on some models it may not be supported. When all flavors cannot be supported on single SID, multiple SIDs should be configured with required flavor.

## 9.9 Restoring the Config Changes

Let's restore the metric changes back to our original lab configurations so we can continue with further chapters. Please refer to the Appendices for the exact configs.

# Chapter 10

## Configuring SRv6-TE

### 10.1 Introduction to SRv6 TE

By now you should be familiar with SRv6 so it's time to show you how you can use SRv6 SIDs in a TE policy to steer the traffic in a particular path from headend in a SRv6 network.

In Junos/Evo today, a static SRv6 TE policy can be configured on the headend device to enable services to follow the TE path instead of a best effort IGP SRv6 Path. The static SRv6 TE policy can have multiple (at max 128) primary ECMP legs and is protected by a secondary path.

The service next hop resolution in BGP for SRv6 TE until Junos 22.2R1 release is over IPv6 lo0 address of the egress peers along with color resolution map.

There are two flavors of SR-TE. When a color is associated with the SR-TE it is called a *colored SR-TE*. When there is no color associated with the tunnel and the name of the LSP is used to identify the SR-TE LSP, it is called *uncolored SR-TE*. In this book we will focus more on colored SR-TE for SRv6. Colored SR-TE is in alignment with the SR-TE policy architecture defined in [RFC 9256](#).

Figure 10.1 depicts the egress and an ingress for SRv6 TE policy.

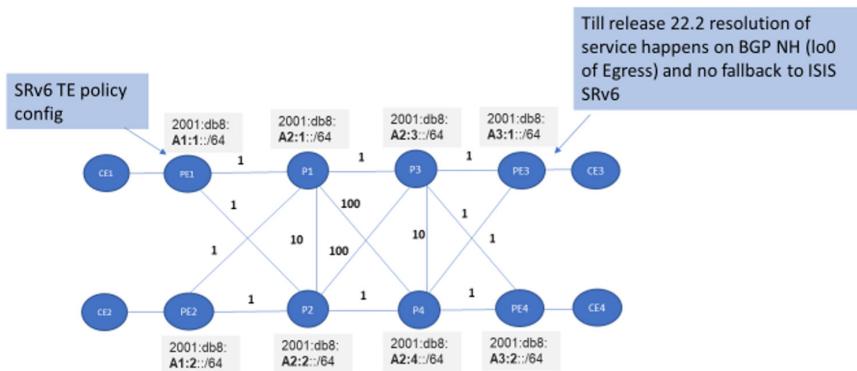


Figure 10.1 A Simplified View of SRv6-TE Policy Headend and Egress PE

In our lab example we will take PE1 as our ingress and PE3 as our egress. On ingress a SRv6 TE policy with endpoint as PE3's IPv6 loopback address and color 3 will be configured statically. The said policy has two primary paths that we want to configure:

PE1→P1→P3→PE3

PE1→P2→P3→PE3

And the above two paths are protected by a secondary path:

PE1→P2→P4→PE3

That means when both P1 and P3 are down only then the traffic is steered through P4 to PE3 as shown in Figure 10.2 and Figure 10.3.

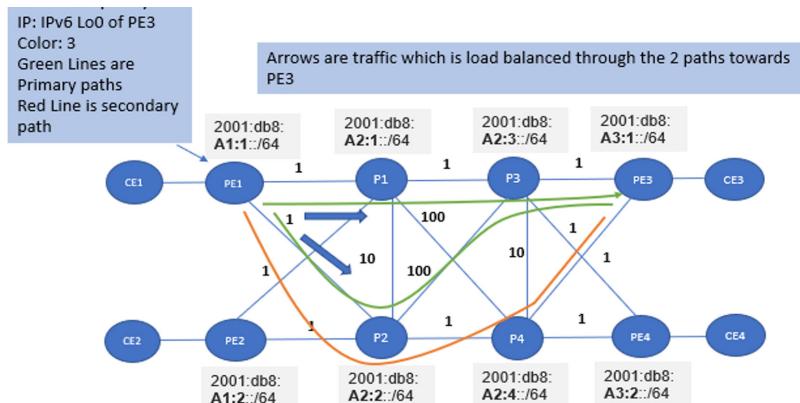


Figure 10.2 Three Segment-Lists of SRv6 TE policy in discussion

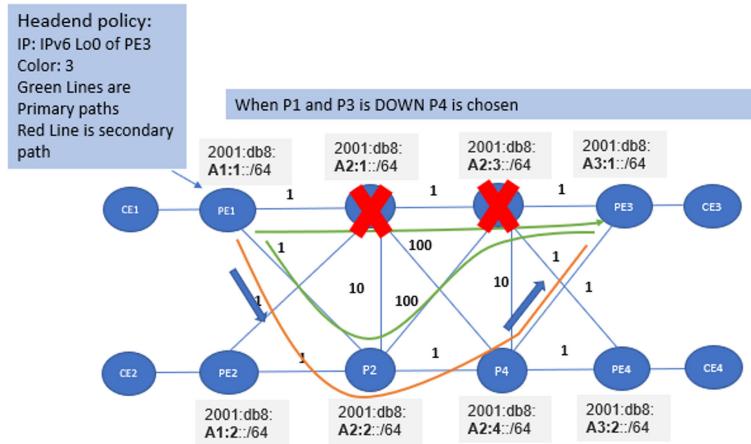


Figure 10.3 Services Transported over Backup SRv6 TE PATH on Primary Node Failure

At the time of this writing, SRv6 TE is supported on the MX Platform. SRv6 TE will be coming to the ACX platform in future releases.

### 10.1.2 Configuration on PE1

PE1 will have a colored SRv6 TE policy towards PE3 which will have three segment-lists. Two of those are primary and one is secondary:

```
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 srv6
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop1 srv6-sid 2001:db8:a2:1:11::0
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 srv6
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 srv6
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop2 srv6-sid 2001:db8:a2:4:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing srv6
set protocols source-packet-routing source-routing-path to_PE3 srv6
set protocols source-packet-routing source-routing-path to_PE3 to 2001:db8:7:255:7::7
set protocols source-packet-routing source-routing-path to_PE3 color 3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P1_P3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P2_P3
set protocols source-packet-routing source-routing-path to_PE3 secondary to_PE3_via_P2_P4
```

In this SRv6 TE policy you can see that the endpoint is the IPv6 loopback address of the iBGP peer PE3 and the color is 3 (the number 3 is chosen just for simplicity to address the policy is directed to PE3).

At the segment list part, the first primary segment list is comprised of three SIDs:

- Node SID of P1
- Node SID of P3
- Node SID of PE3

The second primary segment list is comprised of three SIDs:

- Node SID of P2
- Node SID of P3
- Node SID of PE3

The secondary segment list is also comprised of three SIDs:

- Node SID of P2
- Node SID of P4
- Node SID of PE3

Note that the segment list can also have adjacency SIDs if an explicit path through a particular link has to be used.

Two additional configurations in PE1 that can enable SRv6 TE:

For enabling multipath resolution so that our service can have ECMP legs this is an optional config. If you wish to have only a single path and not to do load balancing you can skip this config:

```
set routing-options resolution rib bgp.13vpn-inet6.0 inet6color-import mpath-resolve
set routing-options resolution rib bgp.13vpn.0 inet6color-import mpath-resolve
```

Note that this is a mandatory config meaning SRv6 TE is required to merge its own SID stack with IGP's stack:

```
set routing-options forwarding-table srv6-chain-merge
```

### 10.1.3 Verification of Static SRv6-TE Routes:

```
root@PE1# run show route protocol spring-te table inet6color.0
inet6color.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:7:255:7::7-3<c6>/160
 *[SPRING-TE/8] 01:32:22, metric 1, metric2 3
 > to fe80::5603:7ff:fe00:3729 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:6970 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:6970 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
```

The colored prefix has three parts:

- `2001:db8:7:255:7::7` : The endpoint of the policy. This is egress PE's loopback address
- `3` : This is the transport color.
- `<c6>` : This means it is an IPv6 colored policy and thus it resides in `inet6color`
- (Note: The above output is from a Junos release prior to 22.3 without Transport Class)

## 10.2 PCEP SRv6 TE

From Junos 21.4R1 release, delegated and PCE-initiated SRv6 TE policies are supported. But local computation of SRv6 TE policies using distributed CSPF is not supported. PCE controller computation of SRv6 TE Policies are not yet supported. The controller today has static CLI commands to provision SRv6 TE policies on the headend.

### 10.2.1 Delegated PCEPbased SRv6 Policies

This works the same way as SRMPLS works today. The SRv6-TE policy is delegated to the controller. The controller, if unable to compute the policy forwarding states, can today return the control to the headend to locally compute or configure the policy and form forwarding states of the policy. In this case the local configuration will take effect.

To configure delegation of SRv6 TE policies here are the incremental configs and any explanation.

Enable PCCD under protocols `mpls`. This enables the PCCD (PCC daemon in Junos) infrastructure:

```
set protocols mpls lsp-external-controller pccd
```

Enable external controller and the PCC client under SRTE:

```
set protocols source-packet-routing lsp-external-controller pccd
```

Configure the SRv6 TE policy:

```
set protocols source-packet-routing source-routing-path to_PE3 srv6
set protocols source-packet-routing source-routing-path to_PE3 to 2001:db8:7:255:7::7
set protocols source-packet-routing source-routing-path to_PE3 color 3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P1_P3
```

For delegating the policy to a PCE controller enables the external controller PCC client for this policy:

```
set protocols source-packet-routing source-routing-path to_PE3 lsp-external-controller pccd
```

Enable PCEP on the router and configure a stateful PCE. When the controller comes and tries to establish a connection with the PCCD it can now do so.

```
set protocols pcep pce 1 pce-type active
set protocols pcep pce 1 pce-type stateful
```

You can configure only one primary path for the SRv6 TE policy at the time of this book's writing. The SRv6 TE policy will take on delegated paths once a PCE update is received from the controller. If the SRv6 TE policy becomes locally controlled, it will download the statically configured path in RIB.

The controller can take back control of this candidate path of the policy anytime and update the segment list. Also the above policy is equipped to carry global internet traffic as well as L3VPN traffic.

## 10.2.2 PCE Initiated SRv6 Policies

The PCE can also provision a SRv6 TE policy candidate path on the headend node. Today Junos has the PCEP capabilities to exchange SRv6 TE policies with the PCE, therefore, a Junos router today supports PCE extensions of reporting SRv6 TE policies, getting updates from a PCE controller for a SRv6 TE policy, and delegations of SRv6 TE policies to a PCE controller (See Figure 10.4). As we don't have a controller in our setup we will not demonstrate the PCEP provisioning from the controller.

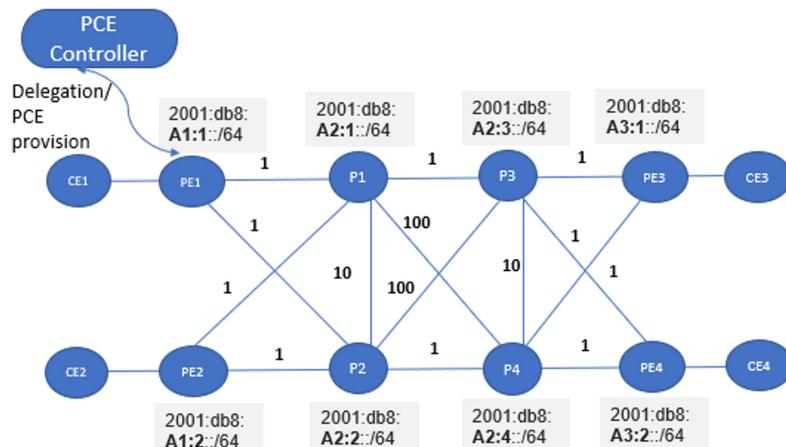


Figure 10.4 A PCE Controller Programming SRv6-TE Policy on Headend Node

## 10.3 Maximum SID Depth

In SRv6 TE policy the segment list can contain one or multiple SIDs. Due to some hardware or forwarding limitations, the SID depth currently cannot exceed more than six in the MX Platform. The depth “6” includes SRv6 TE SIDs plus ISIS TI-LFA SIDs. Let’s take a tour over some use cases to see how this SID depth works.

### 10.3.1 SID Depth when SRv6 TE policy has first segment as ISIS SRv6 SID which only has primary legs

Using our existing example let’s take our first segment list:

```
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 srv6
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop1 srv6-sid 2001:db8:a2:1:1::0
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop2 srv6-sid 2001:db8:a2:3:1:1::0
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop3 srv6-sid 2001:db8:a3:1:1:1::0
```

The first segment points to `2001:db8:a2:1:11::0` which is the End SID of P1. The SRv6 ISIS route towards this first segment:

```
[edit]
root@PE1# run show route 2001:db8:a2:1:11::0 table inet6.3
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a2:1:1::/128
 *[SRV6-ISIS/14] 06:18:46, metric 1
 > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:11::
```

You can see the route has only one primary leg which has only one segment:

```
root@PE1# run show route 2001:db8:a2:1:11::0 table inet6.3 extensive | match seg
Segment-list[0] 2001:db8:a2:1:11::
```

So SRv6 TE has to merge this one segment from ISIS SRv6 SID List with its hop2 and hop3. Thus, the final SID list will be:

```
[edit]
root@PE1# run show route protocol spring-te extensive
inet6color.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
2001:db8:7:255:7::7-3<c6>/160 (1 entry, 1 announced)
*SPRING-TE Preference: 8
 Next hop type: Indirect, Next hop index: 0
 Address: 0x7f6d114
 Next-hop reference count: 1, key opaque handle: 0x0, non-key opaque handle: 0x0
 Next hop type: Chain, Next hop index: 0
 Next hop: via Chain Tunnel Composite, SRv6
 Next hop: ELNH Address 0x787d124, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:1:255:1::1 Dest: 2001:db8:7:255:7::7-3<c6>
 Segment-list[1] 2001:db8:a2:3:11::
 Segment-list[2] 2001:db8:a3:1:11::
... Segment-list[0] 2001:db8:a2:1:11::
```

In this scenario a total of six SIDs can be configured in the segment list.

### 10.3.2 SID Depth when SRv6 TE policy has first segment as ISIS SRv6 SID which has both primary and TI-LFA legs:

Chapter 4 explained TI-LFA in detail. So let's dive directly into how the SRv6 TE maximum SID depth is calculated when the first segment of the SRv6 TE segment list has both primary and TI-LFA legs.

When the TI-LFA configuration is enabled on PE1, the End SID and locator of P1 will have two paths on PE1. Let's check the End SID route of P1 on PE1 because that is our first segment in the SRv6 TE segment list to\_Pe3\_via\_P1\_P3:

```
[edit]
root@PE1# run show route 2001:db8:a2:1:11::
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a2:1:11::/128
 *[SRV6-ISIS/14] 00:00:01, metric 1
 > to fe80::5603:7ff:fe00:609e via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:11::
 to fe80::5603:7ff:fe00:2418 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:11::
```

This brief output shows that there are two paths. Now, let's go through the extensive output to see which is the backup FRR leg:

```
[edit]
root@PE1# run show route 2001:db8:a2:1:11:: table inet6.3 extensive
 Segment-list[0] 2001:db8:a2:1:11:: >>> Primary leg
Next hop: fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0 weight 0x1
...
...
 Segment-list[0] 2001:db8:a1:2:11:: >>> TI-LFA leg
 Segment-list[1] 2001:db8:a2:1:11::
Next hop: fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0 weight 0xf000
```

Notice the primary leg has one SID and that's the End SID of P1. The outgoing interface is ge-0/0/2.0.

The second leg has two SIDs in the segment list. The first SID corresponds to PE2's End SID. Once the segment routing header is processed in PE2, the next destination will be P1's End SID. The outgoing interface for the second leg is ge-0/0/3.0 with weight 0xf000. This means that when ge-0/0/2 is DOWN, only then will this TI-LFA path will be used.

Now, from SRv6 TE's point of view, the first segment list has to do two merges:

- With Primary leg, which is same as section 10.3.1.
- With TI-LFA leg, which itself has two SIDs. So, in this case, SRv6 TE will merge its own SID list with the TI-LFA SID list and the final SID list will contain four SIDs.

Since this merged SID list output is the final resultant SID list, which goes to the FIB, this SID list can only be seen when a service route resolves over SRv6 TE policy and the next hop is programmed in the FIB.

Just to visualize how the SID list looks like, here is one such output taken from a service route which is resolving over a SRv6 TE policy and has the same segment-list we have talked about in this section:

```
SRV6-Tunnel: Reduced-SRH Encap-mode
Src: 2001:db8:1:255:1::1 Dest: 2001:db8:7:255:7::7-3<c6>
Segment-list[0] 2001:db8:a1:2:11:: >>> PE2
Segment-list[1] 2001:db8:a2:1:11:: >>> P1
Segment-list[2] 2001:db8:a2:3:11:: >>> P3
Segment-list[3] 2001:db8:a3:1:11:: >>> PE3
```

Now, in this case, the SRv6 TE segment list cannot have more than five SIDs because the first SID or segment has a TI-LFA path which already has two SIDs.

First SID TI-LFA PATH results in two SIDs in the PATH plus four SIDs in the SRv6 TE stack which results in six SIDs in the final stack. The TI-LFA backup path and the SID depth for backup path varies dynamically based on changing topology when the links go up/down or on metric change. We recommend you estimate the maximum depth of TI-LFA paths in the network. From our observation it is generally two or three. If a budget of two is considered for TI-LFA then SRv6 TE policy can have up to four SIDs.

The default SID depth as mentioned is six in Junos. But this depth can be altered by a configuration such as:

```
[edit]
root@PE1# set protocols source-packet-routing maximum-srv6-te-segment-list-depth ?
Possible completions:
<maximum-srv6-te-segment-list-depth> Maximum segment list depth for SRv6-TE policies (1..6)
```

Any change in the SID depth will trigger a route recalculation.

## 10.4 BGP L3 VPN Services Over SRv6 TE Policies

In earlier chapters you have become familiar with services over SRv6, and the path to egress is a best cost SRv6 path populated by IGP. But networks can have TE-attributes defining the type of services that it can carry. For instance, let's consider the bandwidth or delay metric for a moment where a service requires higher bandwidth rather than best cost or shortest path towards the destination. Here comes the role of services over SRv6 TE policies.

BGP at egress can choose which service to resolve on what SRv6 TE static-colored policies on ingress.

Before diving into the service resolution schemes, one must remember what we have mentioned in Chapter 3, that the last SID in a SRH will be the SRv6 Service SID or the VPN SID. In Junos, different platforms achieve this using different mechanisms which are abstract and have no impact on actual functionality. But the last segment in SRH *must always* be the Service or VPN SID which has the VPN context at the Egress PE.

So, this holds true in case of services resolving over SRv6 TE policies as well. Let's divide the resolution scheme in two parts.

## 10.4.1 BGP L3VPN over SRv6 TE Resolution with Next Hop as BGP-NH or Configured NH at PE3

### 10.4.1.1 When the Next-hop is BGP NH

BGP at ingress will always use the `next-hop` attribute to resolve service prefixes. At egress, you can manipulate this `next-hop` attribute advertised in BGP with `policy-option` configurations.

Until the 22.2R1 Junos release, without any next hop manipulation in egress, there is no fallback mechanism if the colored SRv6-TE policy doesn't exist.

To take an example let's take our scenario discussed in Section 10.1 where we have a statically configured SRv6 TE policy at PE1 (our ingress):

```
set protocols source-packet-routing source-routing-path to_PE3 srv6
set protocols source-packet-routing source-routing-path to_PE3 to 2001:db8:7:255:7::7
set protocols source-packet-routing source-routing-path to_PE3 color 3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P1_P3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P2_P3
set protocols source-packet-routing source-routing-path to_PE3 secondary to_PE3_via_P2_P4
```

The policy and its segment lists are explained in detail in section 10.1.2.

Now let's start with the policy configuration in egress. For the sake of simplicity, and to depict the idea as simply as possible, the policy configuration is a broad one that includes both `inet` and `inet6` services:

```
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 from family inet
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COM_COLOR
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then accept
set policy-options policy-statement L3VPN_PE3_CE3_export term 1 from family inet6
set policy-options policy-statement L3VPN_PE3_CE3_export term 1 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE3_CE3_export term 1 then community add L3VPN_1_COM_COLOR
set policy-options policy-statement L3VPN_PE3_CE3_export term 1 then accept
```

**NOTE** The `inet` and `inet6` policy options can be combined into one. It is separated out just to give you an overview of how different policies can be made for different services. You can add more filters and granularity to the policy.

In both of these export policies no `next-hop` is configured.

The BGP now advertises all service prefixes with a next hop as the IPv6 loopback address of PE3, along with `End.DT4` SID as the service SID for IPv4 services and `End.DT6` SID as the service SID for IPv6 services.

To achieve colored resolution a `resolution-map` policy has to be enabled in PE1 and imported in the iBGP config:

```
set policy-options policy-statement color-resolve-policy then accept
set policy-options policy-statement color-resolve-policy then resolution-map color-resolve
set policy-options resolution-map color-resolve mode ip-color
set protocols bgp group ibgpv6 import color-resolve-policy
```

In egress PE, PE3, a color community must be attached with the export policy so that BGP can advertise the color community for services to ingress PE1:

```
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COLOR_COM
set policy-options community L3VPN_1_COLOR_COM members color:0:3
```

Let's now validate the BGP services and what it means.

At PE3:

```
[edit]
root@PE3# run show route advertising-protocol bgp 2001:db8:1:255:1::1 extensive
L3VPN_PE3_CE3.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (2 entries, 1 announced) >> IPv4 services with Next-Hop as SELF and Service SID as End.
DT4:
BGP group ibgpv6 type Internal
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
Nexthop: Self
 Flags: Nexthop Change
 Localpref: 100
 AS path: [64512] I
 Communities: target:64512:1 color:0:3
 SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
...
...
* 2001:db8:77::/48 (2 entries, 1 announced) >>IPv6 services with Next-Hop as SELF and Service SID as End.DT6:
BGP group ibgpv6 type Internal
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
Nexthop: Self
 Flags: Nexthop Change
 Localpref: 100
 AS path: [64512] I
 Communities: target:64512:1 color:0:3
 SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

At PE1, verify BGP advertisement from PE3:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
*= 7.77.1.0/24 (1 entry, 1 announced) >> IPv4 services
 Import Accepted
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
Nexthop: 2001:db8:7:255:7::7
 Localpref: 100
 AS path: I
 Communities: target:64512:1 color:0:3
 SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
...
...
* 2001:db8:77::/48 (1 entry, 1 announced) >> IPv6 services
```

```

Import Accepted
Route Distinguisher: 7.7.7.7:1
VPN Label: 3
Nexthop: 2001:db8:7:255:7::7
Localpref: 100
AS path: I
Communities: target:64512:1 color:0:3
SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0

```

Validating BGP IPv4 service route over SRv6 TE:

```

[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x747d744) 2001:db8:7:255:7::7-3<c6> >>> SRv6 TE Policy
Krt_cnh (0x6fcfd48) Index:750
 Krt_inh (0x718b6cc) Index:1048583 PNH: 2001:db8:7:255:7::7-3<c6>
 List (0x7a57414) Index:1048577
 Frr_inh (0x747d0b4) Index:1048579 PNH: 2001:db8:a2:1:11:: Session-ID: 379 >> 1st Primary Leg
 Chain Fully resolved tunnel (0x747d824) Index:747 SRv6
 Router (0x747e1c4) Index:744 fe80::5603:7ff:fe00:3729 Session-ID: 332 via ge-0/0/2.0
 Frr_inh (0x747df94) Index:1048575 PNH: 2001:db8:a2:2:11:: Session-ID: 381 >> 2nd Primary leg
 Chain Fully resolved tunnel (0x747d894) Index:660 SRv6
 Router (0x747e384) Index:745 fe80::5603:7ff:fe00:6970 Session-ID: 333 via ge-0/0/3.0
 Frr_inh (0x747df24) Index:1048576 PNH: 2001:db8:a2:2:11:: Session-ID: 381 >> secondary Leg
 Chain Fully resolved tunnel (0x747da54) Index:661 SRv6
 Router (0x747e384) Index:745 fe80::5603:7ff:fe00:6970 Session-ID: 333 via ge-0/0/3.0
...
SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid >> Reduced mode
...
SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0 >>> End.DT4 SID imposed
in the SRH.
...
...
...
Indirect next hop: 0x9300284 - INH Session ID: 0 Weight 0x1
...
Indirect next hop: 0x9300444 - INH Session ID: 0 Weight 0x1
...
Indirect next hop: 0x9300604 - INH Session ID: 0 Weight 0xff >> signifies secondary
...

```

Validating BGP IPv6 service route over SRv6 TE:

```

[edit]
root@PE1# run show route 2001:db8:77:: extensive expanded-nh
2001:db8:77::/48 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x747d124) 2001:db8:7:255:7::7-3<c6>
 Krt_cnh (0x6fcfcf4) Index:655
 Krt_inh (0x718b39c) Index:1048574 PNH: 2001:db8:7:255:7::7-3<c6>
 List (0x7a57414) Index:1048577
 Frr_inh (0x747d0b4) Index:1048579 PNH: 2001:db8:a2:1:11:: Session-ID: 379
 Chain Fully resolved tunnel (0x747d824) Index:747 SRv6
 Router (0x747e1c4) Index:744 fe80::5603:7ff:fe00:3729 Session-ID: 332 via ge-0/0/2.0
 Frr_inh (0x747df94) Index:1048575 PNH: 2001:db8:a2:2:11:: Session-ID: 381
 Chain Fully resolved tunnel (0x747d894) Index:660 SRv6
 Router (0x747e384) Index:745 fe80::5603:7ff:fe00:6970 Session-ID: 333 via ge-0/0/3.0
 Frr_inh (0x747df24) Index:1048576 PNH: 2001:db8:a2:2:11:: Session-ID: 381
 Chain Fully resolved tunnel (0x747da54) Index:661 SRv6

```

```
Router (0x747e384) Index:745 fe80::5603:7ff:fe00:6970 Session-ID: 333 via ge-0/0/3.0
```

```
...
...
SRV6-Tunnel: Reduced-SRH Encap-mode Remove-Last-Sid
...
SRV6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
>>> End.DT6 SID imposed on SRH
...
Indirect next hop: 0x9300284 - INH Session ID: 0 Weight 0x1
...
Indirect next hop: 0x9300444 - INH Session ID: 0 Weight 0x1
...
Indirect next hop: 0x9300604 - INH Session ID: 0 Weight 0xff
```

Validating ping from CE1 to CE3 (both IPv4 and IPv6):

```
root@PE1# run ping 2001:db8:77:: logical-system CE1 rapid count 100000 interval 0.1 size 30 ttl 20
PING6(78=40+8+30 bytes) 2001:db8:11::1 --> 2001:db8:77::
!!!!!!^C
--- 2001:db8:77:: ping6 statistics ---
227 packets transmitted, 226 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 3.773/4.930/37.657/2.265 ms
[edit]
root@PE1# run ping 7.77.1.0 logical-system CE1 rapid count 100000 interval 0.1 size 30
PING 7.77.1.0 (7.77.1.0): 30 data bytes
!!!!!!^C
--- 7.77.1.0 ping statistics ---
217 packets transmitted, 216 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.641/6.099/52.351/4.204 ms
```

Load balancing.

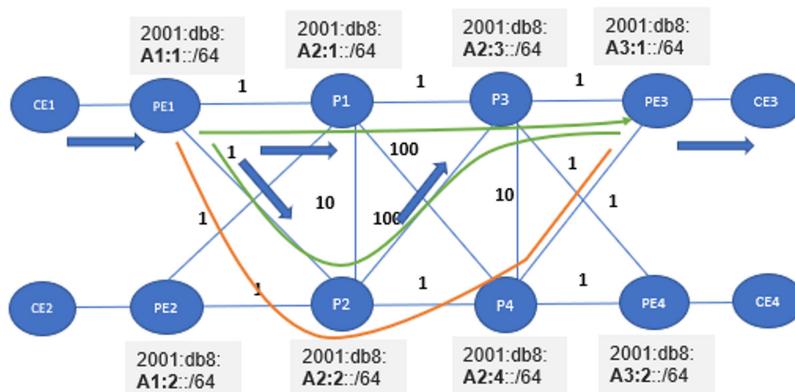


Figure 10.5 Load-balancing with SRv6-TE Transport

Sending traffic all at once and you can see that it is load balanced between P1 and P2 from PE1:

| PE1       |      | Seconds: 41   |       | Time: 04:04:34 |           |
|-----------|------|---------------|-------|----------------|-----------|
| Interface | Link | Input packets | (pps) | Output packets | (pps)     |
| ge-0/0/0  | Up   | 271505        | (373) | 271692         | (372)     |
| lc-0/0/0  | Up   | 0             | 0     | 0              | 0         |
| pfh-0/0/0 | Up   | 0             | 0     | 0              | 0         |
| ge-0/0/1  | Up   | 271497        | (372) | 271505         | (372)     |
| ge-0/0/2  | Up   | 762971        | (171) | 967218         | (202)>>>  |
| ge-0/0/3  | Up   | 981272        | (203) | 1085206        | (170) >>> |

If we disable the link between PE1 and P1 (ge-0/0/2) then we will see traffic only flowing on ge-0/0/3, as in Figure 10.6:

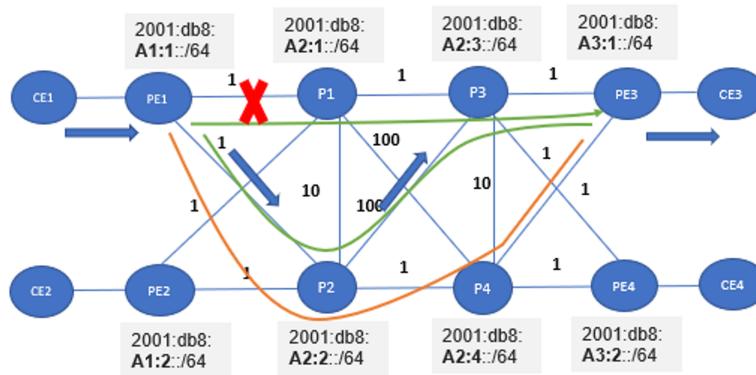


Figure 10.6 Transition of Traffic When One Primary Leg of SRv6-TE Tunnel is Down

| root@PE1# set interfaces ge-0/0/2 disable |      |               |       |                |           |
|-------------------------------------------|------|---------------|-------|----------------|-----------|
| Interface                                 | Link | Input packets | (pps) | Output packets | (pps)     |
| ge-0/0/0                                  | Up   | 326525        | (338) | 326521         | (332)     |
| lc-0/0/0                                  | Up   | 0             | 0     | 0              | 0         |
| pfh-0/0/0                                 | Up   | 0             | 0     | 0              | 0         |
| ge-0/0/1                                  | Up   | 326520        | (338) | 326525         | (338)     |
| ge-0/0/2                                  | Down | 786346        | (0)   | 995037         | (0)       |
| ge-0/0/3                                  | Up   | 1013006       | (338) | 1112522        | (339) >>> |

You can also use uncolored SRv6-TE policy to transport service. In that case the resolution map configured at ingress can be removed and the SRv6 TE policy color configuration can be removed:

```
delete protocols bgp group ibgpv6 import resolution-map-policy
delete protocols source-packet-routing source-routing-path to_Pe3 color
```

**NOTE** With this approach the service cannot fall back to the best cost IGP SRv6 path when the static SRv6 TE policy is withdrawn or deleted from the configuration.

#### 10.4.1.2 When BGP Services over uncolored SRv6-TE/SRv6 with a fallback mechanism with configured next-hop in Egress PE:

In the above example we have seen how resolution over colored SRv6 TE policy works. But that comes with the cost of “no fallback resolution scheme” when the SRv6 TE policy is DOWN or not available.

This can be achieved by removing the color community and changing the next hop in egress to the locator of the egress PE. In our example we have to remove the color community from PE3:

```
delete policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COLOR_COM
delete policy-options community L3VPN_1_COLOR_COM members color:0:3
```

And the map from PE1:

```
delete protocols bgp group ibgpv6 import color-resolve-policy
```

Now configure the next-hop in PE3 explicitly to PE3’s locator address:

```
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 2001:db8:a3:1::
```

Change the SRv6 TE policy endpoint to the locator address of the egress PE and remove the color configuration:

```
set protocols source-packet-routing source-routing-path to_PE3 to 2001:db8:a3:1::0
delete protocols source-packet-routing source-routing-path to_PE3 color
```

Now our BGP advertisement on PE1, from PE3, looks like:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
 Import Accepted
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
 Nexthop: 2001:db8:a3:1:: >> Changed to the configured NH
 Localpref: 100
 AS path: I
 Communities: target:64512:1
 SRv6 SID: 2001:db8:a3:1:3777:: Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
...
...
L3VPN_PE1_CE1.inet6.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (1 entry, 1 announced)
 Import Accepted
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
 Nexthop: 2001:db8:a3:1:: >> Changed to the configured NH
 Localpref: 100
 AS path: I
 Communities: target:64512:1
 SRv6 SID: 2001:db8:a3:1:4777:: Behavior: 18 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
```

And the transport route looks like:

```
[edit]
root@PE1# run show route 2001:db8:a3:1::/64 table inet6.3
inet6.3: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 00:32:34, metric 3
 > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
2001:db8:a3:1::/128*[SPRING-TE/8] 00:31:16, metric 1, metric2 3
 > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
2001:db8:a3:1:11::/128
 *[SRV6-ISIS/14] 00:32:34, metric 3
 > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11::
```

You can see that there are three routes to the 64 bit prefix. The first one comes from ISIS and it's the 64 bit locator SRv6 route. The second one is downloaded by SPRING-TE as a 128 bit route over three segment lists. And the third one is the End SID route of egress PE3 downloaded by ISIS.

Since resolution happens with LPM (longest prefix match), the second route will be chosen. So here the service will resolve over SRv6 TE policy route. At this point our service route looks like the following with three ECMP legs:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x747dcf4) 2001:db8:a3:1:: Session-ID: 360
 Krt_cnh (0x6fd1440) Index:710
 Krt_inh (0x934f404) Index:1048607 PNH: 2001:db8:a3:1::
 List (0x7adf414) Index:1048606
 Frr_inh (0x747dac4) Index:1048600 PNH: 2001:db8:a2:1:11:: Session-ID: 361
 Chain Fully resolved tunnel (0x747ddd4) Index:706 SRv6
 Router (0x747c7f4) Index:747 fe80::5603:1ff:fe00:59fa Session-ID: 332 via ge-0/0/2.0
 Frr_inh (0x747dba4) Index:1048601 PNH: 2001:db8:a2:2:11:: Session-ID: 362
 Chain Fully resolved tunnel (0x747e1c4) Index:707 SRv6
 Router (0x747c8d4) Index:748 fe80::5603:1ff:fe00:6248 Session-ID: 333 via ge-0/0/3.0
 Frr_inh (0x747e704) Index:1048603 PNH: 2001:db8:a2:2:11:: Session-ID: 362
 Chain Fully resolved tunnel (0x747ee74) Index:708 SRv6
 Router (0x747c8d4) Index:748 fe80::5603:1ff:fe00:6248 Session-ID: 333 via ge-0/0/3.0
...
 Indirect next hop: 0x9350044 - INH Session ID: 0 Weight 0x1
...
 Indirect next hop: 0x9350204 - INH Session ID: 0 Weight 0x1
...
 Indirect next hop: 0x9351a84 - INH Session ID: 0 Weight 0xffff
```

And traffic validation:

```
[edit]
root@PE1# run ping 7.77.1.0 logical-system CE1
PING 7.77.1.0 (7.77.1.0): 56 data bytes
64 bytes from 7.77.1.0: icmp_seq=0 ttl=63 time=5.528 ms
```

```
64 bytes from 7.77.1.0: icmp_seq=1 ttl=63 time=5.022 ms
^C
--- 7.77.1.0 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.022/5.275/5.528/0.253 ms
```

Now, if we remove the SRv6 TE policy from configuration, the LPM will be over and ISIS SRv6 locator route and the service can still be resolved:

```
[edit]
root@PE1# deactivate protocols source-packet-routing source-routing-path to_Pe3
[edit]
root@PE1# commit
commit complete
```

Now the transport looks like:

```
root@PE1# run show route 2001:db8:a3:1::/64 table inet6.3
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 00:40:59, metric 3
 > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
2001:db8:a3:1:11::/128
 *[SRV6-ISIS/14] 00:40:59, metric 3
 > to fe80::5603:1ff:fe00:59fa via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11::
 to fe80::5603:1ff:fe00:6248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11::
```

With no SRv6-TE route our service looks like:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x747dd64) 2001:db8:a3:1:: Session-ID: 363
Krt_cnh (0x6fcece0) Index:657
Krt_inh (0x934d484) Index:1048574 PNH: 2001:db8:a3:1::
List (0x79a1ddc) Index:1048595
Chain Fully resolved tunnel (0x747e624) Index:701 SRv6
 Router (0x747c7f4) Index:747 fe80::5603:1ff:fe00:59fa Session-ID: 332 via ge-0/0/2.0
Chain Fully resolved tunnel (0x747ca94) Index:788 SRv6
 Router (0x747c8d4) Index:748 fe80::5603:1ff:fe00:6248 Session-ID: 333 via ge-0/0/3.0
```

If you notice, the service now has only two ECMP legs that are populated by ISIS SRv6.

Here's the traffic validation:

```
[edit]
root@PE1# run ping 7.77.1.0 logical-system CE1
PING 7.77.1.0 (7.77.1.0): 56 data bytes
64 bytes from 7.77.1.0: icmp_seq=0 ttl=63 time=5.556 ms
64 bytes from 7.77.1.0: icmp_seq=1 ttl=63 time=5.227 ms
^C
--- 7.77.1.0 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.227/5.392/5.556/0.164 ms
```

## 10.4.2 BGP L3VPN Over SRv6-TE Policy and Fallback to SRv6 Locator

Things get easier and simpler after Junos 22.3R1. These next configurations are not required.

Ingress:

```
set policy-options policy-statement color-resolve-policy then accept
set policy-options policy-statement color-resolve-policy then resolution-map color-resolve
set policy-options resolution-map color-resolve mode ip-color
set protocols bgp group ibgpv6 import color-resolve-policy
```

From Junos 22.3R1 we recommend you use Transport Class to achieve simpler resolution schemes. Transport Class defines a set of characteristics that the path through the network satisfies. All the paths that satisfy these characteristics will be placed in a RIB specific to the Transport Class and these RIBs are called Transport Class RIBs. Transport Class is associated with a color. The services that intend to use a specific path belonging to a particular Transport Class carries a mapping community in the service advertisement. This mapping community may be a regular BGP community or an extended color community. When the service carries an extended color community, the service will automatically resolve on the Transport RIB of the matching color of the associated Transport Class.

Transport Classes are defined under `routing-options`. In order to simplify configurations, automatic creation of transport classes based on the color of the transport tunnel is supported.

When `transport-class` is auto-created, they will not have a name but will be identified by a color. In this book, we will use auto-created transport classes where the color in SR-TE LSP configuration will automatically create the Transport Class and the corresponding Transport RIB. To achieve the full advantage of the service resolutions in 22.3R1 these configs are recommended strongly:

```
set routing-options transport-class auto-create
set protocols source-packet-routing use-transport-class
```

The first is required for the underlying Classful Transport infrastructure to become active and to ensure `auto-create` the transport RIBs. The second is to enable SRv6 TE route to be downloaded in the auto-created transport RIB. The SRv6 TE route also gets downloaded to `inet6color` table as well but that `inet6color` is discouraged by Juniper and may get discontinued in future. Transport Class architecture is the new future.

The BGP service route also carries another property which is called *SRv6 SID transposition*. It tries to transpose the SID in such a way that it can resolve over SRv6 locators so that different VPN SIDs can be packed in a single BGP update. More info on this can be found in the below draft: <https://datatracker.ietf.org/doc/html/draft-ietf-bess-srv6-services-03#section-3.2.1>

Let's discuss the resolution schemes for services in the next two sections.

#### 10.4.2.1 Resolution schemes for services not advertised with BGP COLOR community:

In the egress PE when the color community is not configured, the BGP received route in the ingress PE looks like the following. Here our egress PE is PE3 and our ingress PE is PE1:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
 Import Accepted MultiNexthop RecvNextHopIgnored
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 14199
 Nexthop: 2001:db8:7:255:7::7
 Localpref: 100
 AS path: I
 Communities: target:64512:1
 SRv6 SID: 2001:db8:a3:1:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64
L3VPN_PE1_CE1.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (1 entry, 1 announced)
 Import Accepted MultiNexthop RecvNextHopIgnored
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 18295
 Nexthop: 2001:db8:7:255:7::7
 Localpref: 100
 AS path: I
 Communities: target:64512:1
 SRv6 SID: 2001:db8:a3:1:: Behavior: 18 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64
```

The SRv6 SID in this advertise SRv6 SID: 2001:db8:a3:1::. If you look closely at the bits behavior: 18 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64, where:

- The TL: 16 is the number of bits that has to be transposed from the service/VPN SID.
- The TO: 64 is the offset from where the TL number of bits has to be transposed.

So, in this case, from the 64th bit, 16 bits need to be transposed.

Our VPN SIDs from egress PE3 are:

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3777:0:0:0
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:1:4777:0:0:0
```

You can clearly see that “3777” and “4777” are transposed to “0”. Until 2001:db8:a3:1 is 64 bits and it belongs to the locator loc1 configured in egress PE3, after that “3777” and “4777” comprises 16 bits. BGP service will use this transposed SID for resolution if BGP NH is not resolved.

For services without color community the locator route in inet6.3 table will always be chosen for resolution. In other words, these service route will always resolve over ISIS SRv6 locators and not any SRv6 TE policies.

The color of any TE policies somewhat distinguishes between different policies with different TE attributes. Without color the TE notion fades away.

The locator route is already populated in inet6.3 table by ISIS SRv6.

The inet6.3 table for the locator route looks like:

```
[edit]
root@PE1# run show route table inet6.3 2001:db8:a3:1::
inet6.3: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[SRV6-ISIS/14] 03:45:05, metric 3
 > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

And the service prefix after resolution looks like:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x787e3f4) 2001:db8:a3:1:: Session-ID: 360
 Krt_cnh (0x73cfdf0) Index:558
 Krt_inh (0x7edbac4) Index:1048578 PNH: 2001:db8:a3:1::
 List (0x7da2924) Index:1048584
 Chain Fully resolved tunnel (0x787e0e4) Index:616 SRv6
 Router (0x787cf64) Index:612 fe80::5603:7ff:fe00:2d9f Session-ID: 344 via ge-0/0/2.0
 Chain Fully resolved tunnel (0x787e1c4) Index:617 SRv6
 Router (0x787d044) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
```

#### 10.4.2.2 Resolution Schemes for Services Advertised with BGP COLOR Community

At egress PE, PE3, the color community is configured as:

```
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COLOR_COM
set policy-options community L3VPN_1_COLOR_COM members color:0:3
```

Once this is set, the receiving BGP route at PE1 looks like:

```
[edit]
root@PE1# run show route receive-protocol bgp 2001:db8:7:255:7::7 extensive
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
* 7.77.1.0/24 (1 entry, 1 announced)
 Import Accepted MultiNexthop
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
 Nexthop: 2001:db8:7:255:7::7
 Localpref: 100
 AS path: I
 Communities: target:64512:1 color:0:3
 SRv6 SID: 2001:db8:a3:1:: Behavior: 19 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64
L3VPN_PE1_CE1.inet6.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
* 2001:db8:77::/48 (1 entry, 1 announced)
 Import Accepted MultiNexthop
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 4
```

```

Nexthop: 2001:db8:7:255:7::7
Localpref: 100
AS path: I
Communities: target:64512:1 color:0:3
SRv6 SID: 2001:db8:a3:1:: Behavior: 18 BL: 64 NL: 0 FL: 16 AL: 0 TL: 16 TO: 64

```

Here also is the transposition logic applied the same as described in section 10.3.2.2. What is different is the resolution scheme. Let's first dive into the SRv6 TE colored transport.

With Transport Class configuration enabled, SR-TE will now download routes into Transport Class RIBs. Here's the SRv6 TE configuration in PE1 (the segment list is same as described in section 10.1.2):

```

set protocols source-packet-routing source-routing-path to_PE3_test srv6
set protocols source-packet-routing source-routing-path to_PE3_test to 2001:db8:7:255:7::8
set protocols source-packet-routing source-routing-path to_PE3_test color 3
set protocols source-packet-routing source-routing-path to_PE3_test primary to_PE3_via_P1_P3
set protocols source-packet-routing source-routing-path to_PE3_test primary to_PE3_via_P2_P3
set protocols source-packet-routing source-routing-path to_PE3_test secondary to_PE3_via_P2_P4

```

You can see now that the transport RIB is populated:

```

root@PE1# run show route table ?
Possible completions:
<table> Name of routing table
L3VPN_PE1_CE1.inet.0
L3VPN_PE1_CE1.inet6.0
bgp.13vpn-inet6.0
bgp.13vpn.0
inet.0
inet6.0
inet6.3
inet6color.0
iso.0
junos-rti-tc-3.inet6.3
mpls.0

```

Notice `junos-rti-tc-3.inet6.3`. The name of this RIB starts with Junos, “tc” means Transport Class, “3” in tc-3 identifies the color for which this transport is created, and `inet6` notifies the family of Endpoint Address. Thus, all Flex Algo and SRv6 TE routes can come under its own Transport carrying its own TE attributes.

The SRv6 TE route in this table looks like:

```

[edit]
root@PE1# run show route table junos-rti-tc-3.inet6.3
junos-rti-tc-3.inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:7:255:7::7/128
 *[SPRING-TE/8] 00:04:26, metric 1, metric2 3
 > to fe80::5603:7ff:fe00:2d9f via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:22e9 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>

```

Since the color is already defined at the route table level, only the IP prefix route can reside under this table.

Back to the service resolution, it first looks up the Transport Class RIB for the color community that is advertised with the service. Here the color community is 3 and our TC RIB is auto-created for color 3, and thus resolution happens on this SRv6 TE route. Hence our service looks like:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
List (0x7da2cb4) Index:1048590
 Indr Composite (0x787d744) 2001:db8:7:255:7::7 Session-ID: 373
 Krt_cnh (0x73d1830) Index:621
 Krt_inh (0x7edb904) Index:1048588 PNH: 2001:db8:7:255:7::7
 List (0x7f6b214) Index:1048583
 Frr_inh (0x787dd64) Index:1048580 PNH: 2001:db8:a2:1:11:: Session-ID: 370
 Chain Fully resolved tunnel (0x787e5b4) Index:615 SRv6
 Router (0x787cf64) Index:612 fe80::5603:7ff:fe00:2d9f Session-ID: 344 via ge-0/0/2.0
 Frr_inh (0x787d664) Index:1048581 PNH: 2001:db8:a2:2:11:: Session-ID: 371
 Chain Fully resolved tunnel (0x787e704) Index:618 SRv6
 Router (0x787d044) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
 Frr_inh (0x787e004) Index:1048582 PNH: 2001:db8:a2:2:11:: Session-ID: 371
 Chain Fully resolved tunnel (0x787e464) Index:619 SRv6
 Router (0x787d044) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
 Indr Composite (0x787de44) 2001:db8:a3:1:: Session-ID: 368
 Krt_cnh (0x73d1c74) Index:610
 Krt_inh (0x7edae84) Index:1048578 PNH: 2001:db8:a3:1:::
 List (0x7da2924) Index:1048584
 Chain Fully resolved tunnel (0x787e0e4) Index:616 SRv6
 Router (0x787cf64) Index:612 fe80::5603:7ff:fe00:2d9f Session-ID: 344 via ge-0/0/2.0
 Chain Fully resolved tunnel (0x787e1c4) Index:617 SRv6
 Router (0x787d044) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
```

In this output there are two main things to notice:

- **Indr Composite (0x787d744) 2001:db8:7:255:7::7:** This is the BGP NH which is used primarily for resolution. The originating RIB for this NH is our TC RIB. The weight for this Indirect Next Hop identifies that this is primary:

The originating RIB:

```
root@PE1# run show route 7.77.1.0 extensive expanded-nh | match rib
2001:db8:7:255:7::7/128 Originating RIB: junos-rti-tc-3.inet6.3
```

The weight:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
...
...
 Protocol next hop: 2001:db8:7:255:7::7 Metric: 3
 Composite next hop: 0x73d1830 621 INH Session ID: 373
 Indirect next hop: 0x7edb904 1048588 INH Session ID: 373 Weight 0x1
```

- This is the SRv6 (Transposed) SID which is used for fallback. The originating RIB for this route is inet6.3 where the ISIS SRv6 locator route resides. The weight for this Indirect Next Hop identifies that this next hop is backup and to be used when the primary is fully DOWN:

The originating RIB (same command as before):

```
2001:db8:a3:1::/64 Originating RIB: inet6.3
```

The weight:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
...
...
Protocol next hop: 2001:db8:a3:1:: Metric: 3
Composite next hop: 0x73d1c74 610 INH Session ID: 368
Indirect next hop: 0x7edae84 1048578 INH Session ID: 368 Weight 0x2
```

Now when the SRv6 TE policy is withdrawn or become unresolved, the service takes on only the locator route:

```
[edit]
root@PE1# deactivate protocols source-packet-routing source-routing-path to_PE3
[edit]
root@PE1# show | compare
[edit protocols source-packet-routing]
! inactive: source-routing-path to_PE3 { ... }
[edit]
root@PE1# commit
```

The service route looks like:

```
[edit]
root@PE1# run show route 7.77.1.0 extensive expanded-nh
L3VPN_PE1_CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
7.77.1.0/24 (1 entry, 1 announced)
Installed-nexthop:
Indr Composite (0x7da203c) 2001:db8:7:255:7::7
 Krt_inh (0x0)
 Krt_cnh (0x73d1c74) Index:610
 Krt_inh (0x7edae84) Index:1048578 PNH: 2001:db8:a3:1:::
 List (0x7da2924) Index:1048584
 Chain Fully resolved tunnel (0x787e0e4) Index:616 SRv6
 Router (0x787cf64) Index:612 fe80::5603:7ff:fe00:2d9f Session-ID: 344 via ge-0/0/2.0
 Chain Fully resolved tunnel (0x787e1c4) Index:617 SRv6
 Router (0x787d044) Index:613 fe80::5603:7ff:fe00:22e9 Session-ID: 345 via ge-0/0/3.0
```

As you can see, the first Krt\_inh is (0x0) which means our BGP NH in TC RIB for color 3 is not present or unresolvable and thus the second protocol next hop becomes the only resolvable next hop.

So now traffic can flow via the fallback Path.

# Chapter 11

## EVPN-VPWS

### 11.1 Introduction

EVPN VPWS provides point to point Layer 2 VPN service using BGP EVPN NLRI. EVPN VPWS is the latest form of L2VPN and enables functionality rich L2VPN service. It provides single homing and active-active/active-standby multihoming functionality. Traditionally EVPN has been supported with MPLS underlay, but from Junos 22.3, support is added for an SRv6 underlay. This chapter covers configuration and operational aspects of basic EVPN VPWS service over the SRv6 underlay, gradually explaining the active/active and active/standby functionality:

- Basic prerequisite to bring EVPN VPWS over SRv6
- Static end-dx2-sid provisioning
- Dynamic end-dx2-sid provisioning
- Active/Active Multihoming
- Active/Standby Multihoming

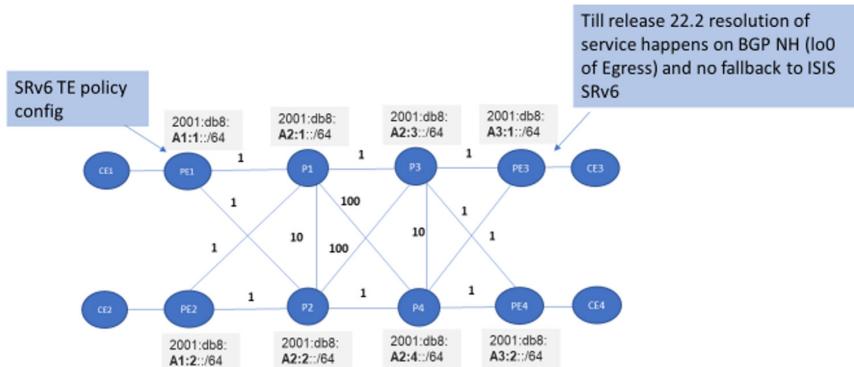


Figure 11.1 EVPN VPWS Over an SRv6 Topology

## 11.2 Basic Configuration Constructs to Enable EVPN VPWS Over SRv6

Let's configure EVPN VPWS on a setup which has a basic SRv6 underlay already configured. The set of configurations emphasize SRv6 specific configuration.

Migration from an MPLS underlay to SRv6 requires bringing up SRv6 underlay transport tunnels, enabling an IPv6 session for EVPN family under protocol BGP, then applying the SRv6 specific configuration to the EVPN VPWS config.

Here's the configuration to allow signaling SRv6 Service TLVs for EVPN NLRI. The configuration allows sending and processing of SRv6 L2 Service TLV's when EVPN VPWS over SRv6 control plane routes are present. The session won't bounce when the configuration is added or removed:

```
set protocols bgp group ibgpv6 family evpn signaling advertise-srv6-service
set protocols bgp group ibgpv6 family evpn signaling accept-srv6-service
```

This configuration on the BGP session allows advertising and receiving SRv6 SIDs in the EVPN family advertisements.

Configure the policy to set protocol nexthop on the egress to point to the locator address. This enables EVPN service to resolve over SRv6 Tunnel on ingress.

On PE1

```
set policy-options policy-statement vpws-nh-change term a from protocol evpn
set policy-options policy-statement vpws-nh-change term a then next-hop 2001:db8:a1:1
set policy-options policy-statement vpws-nh-change term a then accept
set protocols bgp group ibgpv6 export vpws-nh-change
set protocols bgp group ibgpv6 vpn-apply-export
```

On PE2

```
set policy-options policy-statement vpws-nh-change term a from protocol evpn
set policy-options policy-statement vpws-nh-change term a then next-hop 2001:db8:a1:2
set policy-options policy-statement vpws-nh-change term a then accept
set protocols bgp group ibgpv6 export vpws-nh-change
set protocols bgp group ibgpv6 vpn-apply-export
```

### On PE3

```
set policy-options policy-statement vpws-nh-change term a from protocol evpn
set policy-options policy-statement vpws-nh-change term a then next-hop 2001:db8:a3:1
set policy-options policy-statement vpws-nh-change term a then accept
set protocols bgp group ibgpv6 export vpws-nh-change
set protocols bgp group ibgpv6 vpn-apply-export
```

Configure encapsulation type as SRv6:

```
set routing-instances evpn-vpws protocols evpn encapsulation srv6
```

The change in encapsulation will lead to re-creation for EVPN VPWS instances.

Shown in Figure 11.2 is static end-dx2-sid provisioning: Where the ge-0/0/1.4 is the PE-CE interface on the PE which connects to the CE. vpws-service-id local and remote pair makes the pseudowire.

The commit will fail if locator loc1 is not configured at routing options as shown. Also the end-dx2-sid must have the matching prefix of the locator SID configured at routing option, else the commit will fail.

```
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing srv6 locator loc1
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing srv6 locator end-dx2-sid 2001:db8:a1:1:101:0:0:0
```

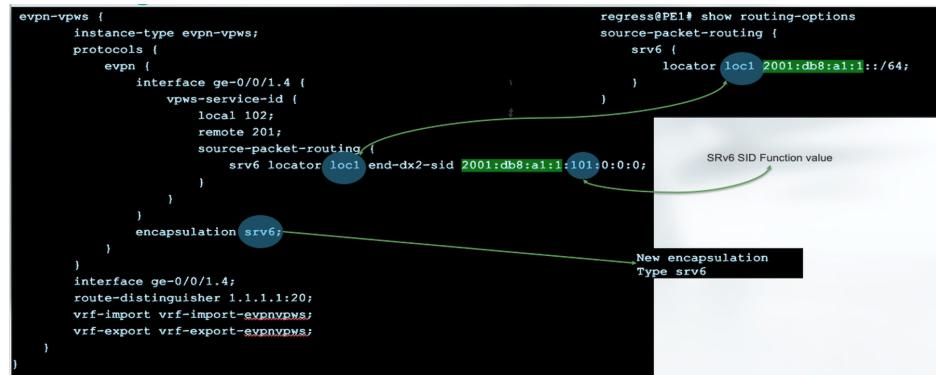


Figure 11.2 EVPN VPWS with Static End.Dx2 configuration

Next, you can see configured end-dx2-sid is shown in the show evpn vpws-instance output as Advertised End.Dx2 SID in the output. Show output format is inherited from the base EVPN-VPWS implementation over MPLS. Encapsulation type, Advertised End.Dx2 SID, and entry for End.Dx2 SID under remote SID are the new additions made with SRv6 underlay. Control-word, Flow-label-TX, and Flow-Label-Rx should be ignored in the SRv6 world. TS is the timestamp of the arrival of remote AD route. Status is Resolved when both AD per EVI and AD per ESI are received and processed in case of non-zero ESI (therefore, multihoming). And it's always resolved for single homing case.

## EVPN VPWS with Static End.Dx2 operational command output

```

regress@PE1# run show evpn vpws-instance
Instance: evpn-vpws, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 1.1.1.1:20
 Number of local interfaces: 1 (1 up)

Tx Interface name ESI Mode Role Status Control-Word Flow-Label-
 Flow-Label-Rx
 ge-0/0/1.4 00:00:00:00:00:00:00:00:00:00:00 single-homed Primary Up No o N
No
 Local SID: 102 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:1:101::
 Remote SID: 201
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00:00 3 2001:db8:a3:1:103:: single-homed Primary
2022-08-02 11:25:56.217 Resolved
 Number of protect interfaces: 0

```

Shown in Figure 11.3 is dynamic end-dx2-sid provisioning. You just configure the locator under `vpws-service-id` to enable dynamic provisioning of the `end-dx2-sid`. This ensures `en-dx2-sid` for the service is allocated dynamically corresponding to the locator prefix.

```
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing srv6 locator loc1
```

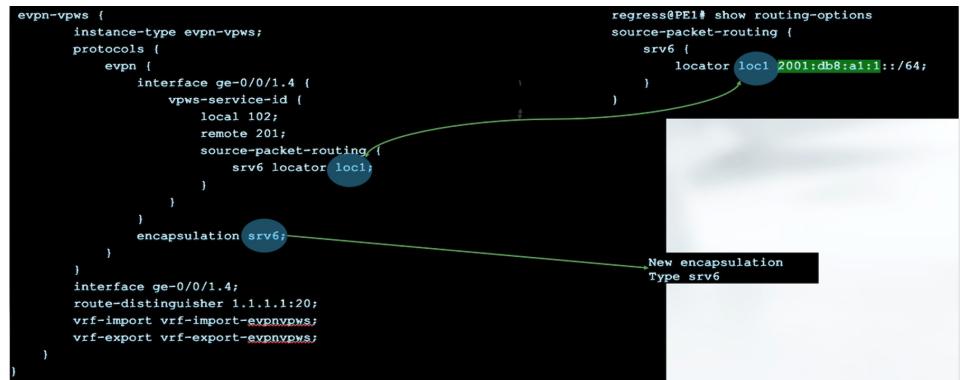


Figure 11.3 FVPN VPWS with Dynamic End-Dx2 Configuration

Let's observe the dynamically allocated SID as advertised End.Dx2 SID in the `show evpn vpwss-instance` output.

```
regress@PE1# run show evpn vpws-instance evpn-vpws-mh
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 1.1.1.12
 Number of local interfaces: 1 (1 up)
```

```
No Local SID: 103 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:1:8001::
PE addr ESI Label End.Dx2 SID Mode Role TS
Status
2001-08-02 2001:db8:a1:2:: 00:12:12:12:12:12:12:12:12:12:12:3 2001:db8:a1:2:8001:: all-active Primary
2022-08-02 20:54:48.897 Resolved
 Remote SID: 301
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
2001-08-02 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00:00:3 2001:db8:a3:1:8001:: single-homed Primary
2022-08-02 20:54:43.169 Resolved
 Number of protect interfaces: 0

Fast Convergence Information
ESI: 00:12:12:12:12:12:12:12:12:12:12:12 Number of PE nodes: 1
PE: 2001:db8:a1:2::
Advertised SID: 103
```

To debug and find allocation details of SID you can use `show srv6 locator <locator_name>`. Observe the dynamic SID range and dynamically allocated End.Dx2 SID value in this SRv6 SID locator detail:

```
regress@PE1# run show srv6 locator loc1
```

Warning: License key missing; requires 'srv6' license

```
Locator: loc1
 Locator prefix: 2001:db8:a1:1::, Locator length: 64
 Block length: 64, Node length: 0
 Function length: 16, Argument length: 0
 Static SID range: 0x1-0x7FFF, Dynamic SID range: 0x8000-0xFFFF
 Allocated static SID count: 5, Allocated dynamic SID count: 1
 Available static SID count: 32762, Available dynamic SID count: 32767
SID SID-Owner SID-Type SID-Behavior
2001:db8:a1:1:101:: EVPN STATIC End.DX2
2001:db8:a1:1:511:: BGP STATIC End.DT4
2001:db8:a1:1:611:: BGP STATIC End.DT6
2001:db8:a1:1:3111:: BGP STATIC End.DT4
2001:db8:a1:1:4111:: BGP STATIC End.DT6
2001:db8:a1:1:8001:: EVPN DYNAMIC End.DX2
```

## 11.2 EVPN VPWS Single Homed Over SRv6

Configure the encapsulation type as `srv6` and the `End.Dx2 SID` and EVPN VPWS config is ready for the SRv6 underlay. The basic constructs for SRv6 underlay and other basic constructs (explained in 11.1) are in place:

```
set routing-instances evpn-vpws protocols evpn encapsulation srv6
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing srv6 locator loc1
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing srv6 locator end-dx2-sid 2001:db8:a1:1:101::0:0:0
```

```
regress@PE1# run show evpn vpws-instance evpn=vpws
```

```

Instance: evpn-vpws, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 1.1.1.1:20
 Number of local interfaces: 1 (1 up)

 Interface name ESI Mode Role Status Control-Word Flow-
Label-Tx Flow-Label-Rx ge-0/0/1.4 00:00:00:00:00:00:00:00:00 single-homed Primary Up No o N
No
 Local SID: 102 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:1:101::
 Remote SID: 201
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00 3 2001:db8:a3:1:103:: single-homed Primary
2022-08-02 11:25:56.217 Resolved
 Number of protect interfaces: 0

regress@PE3# run show evpn vpws-instance evpn-vpws
Instance: evpn-vpws, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 7.7.7.7:20
 Number of local interfaces: 1 (1 up)

 Interface name ESI Mode Role Status Control-Word Flow-
Label-Tx Flow-Label-Rx ge-0/0/1.4 00:00:00:00:00:00:00:00:00:00 single-homed Primary Up No o N
No
 Local SID: 201 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a3:1:103::
 Remote SID: 102
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:1:: 00:00:00:00:00:00:00:00:00:00 3 2001:db8:a1:1:101:: single-homed Primary
2022-08-02 11:38:36.560 Resolved
 Number of protect interfaces: 0

```

Let's check EVPN VPWS control plane routes. Figure 11.8 is an AD route received from a remote peer. Check the BGP PNH and route in the inet6.3 table. Without the BGP PNH route for resolution in inet6.3, the route will remain in a hidden state.

SRv6 SID shows the end-dx2-sid of the remote peer. Notice also the value of behavior is set to 21, the value for L2 service SID attribute:

```

regress@PE1# run show route table evpn-vpws.evpn.0 extensive match-prefix 1:7.7.7.7:20::0::201

evpn-vpws.evpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
1:7.7.7.7:20::0::201/192 AD/EVI (1 entry, 1 announced)
 *BGP Preference: 170/-101
 Route Distinguisher: 7.7.7.7:20
 Next hop type: Indirect, Next hop index: 0
 Address: 0x7b42614
 Next-hop reference count: 2, key opaque handle: 0x0, non-key opaque handle: 0x0
 Source: 2001:db8:7:255:7::7
 Protocol next hop: 2001:db8:a3:1::
 Indirect next hop: 0x2 no-forward INH Session ID: 0
 State: <Secondary Active Int Ext>
 Local AS: 64512 Peer AS: 64512
 Age: 2:13 Metric2: 3
 Validation State: unverified
 ORR Generation-ID: 0

```

```
Task: BGP_64512.2001:db8:7:255:7::7
Announcement bits (1): 0-evpn-vpws-evpn
AS path: I
Communities: target:200:1 evpn-l2-info:0x0 (mtu 0)
Import Accepted
SRv6 SID: 2001:db8:a3:1:103:: Behavior: 21 BL: 64 NL: 0 FL: 16 AL: 0 TL: 0 TO: 0
Route Label: 3
Localpref: 100
Router ID: 7.7.7.7
Primary Routing Table: bgp.evpn.0
Thread: junos-main
Indirect next hops: 1
 Protocol next hop: 2001:db8:a3:1:: Metric: 3
 Indirect next hop: 0x2 no-forward INH Session ID: 0
 Indirect path forwarding next hops: 2
 Next hop type: List
 Next hop: fe80::5604:16ff:fe01:2d53 via ge-0/0/3.0
 Next hop: fe80::5604:16ff:fe00:561d via ge-0/0/4.0
 2001:db8:a3:1::/64 Originating RIB: inet.6.3
 Metric: 3 Node path count: 1
 Forwarding nexthops: 2
 Next hop type: List
 Next hop: fe80::5604:16ff:fe01:2d53 via ge-0/0/3.0
 Next hop: fe80::5604:16ff:fe00:561d via ge-0/0/4.0
```

Observe that the BGP next hop is set as the locator address and route to locator address is present in inet6.3:

```
regress@PE1# run show route table inet6.3 match-prefix 2001:db8:a3:1::
```

```
inet6.3: 29 destinations, 29 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:a3:1::/64 *[SRV6-ISIS/14] 01:41:09, metric 3
 > to fe80::5604:16ff:fe01:2d53 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5604:16ff:fe00:561d via ge-0/0/1.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

Now notice the advertised SRv6 SID in the `show evpn vpws-instance` and in the control plane message. Check the advertised route in `bgp.evpn.0` table. It captures the advertised SRv6 SID and the behavior value is set to 21:

```
regress@PE1# run show route table bgp.evpn.0 extensive protocol evpn
bgp.evpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
1:1.1.1.1:20::0::102/192 AD/EVI (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group ibgpv6 type Internal) Type 1 val 0x9b3a184 (adv_entry)
 Advertised metrics:
 Flags: Nexthop Change
 Nexthop: 2001:db8:a1:1::
 Localpref: 100
 AS path: [64512] I
 Communities: target:200:1 evpn-l2-info:0x0 (mtu 0)
 SRv6 SID: 2001:db8:a1:1:101:: Behavior: 21 BL: 64 NL: 0 FL: 16 AL: 0 TL: 0 T0: 0
 Advertise: 00000005
Path 1:1.1.1.1:20::0::102
Vector len 4. Val: 0
 *EVPN Preference: 170
 Next hop type: Indirect, Next hop index: 0
 Address: 0x7b3f124
 Next-hop reference count: 4, key opaque handle: 0x0, non-key opaque handle: 0x0
 Protocol next hop: abcd::128:53:34:209
 Indirect next hop: 0x0 - INH Session ID: 0
 State: <Secondary Active Int Ext>
 Age: 9:51:34
 Validation State: unverified
 Task: evpn-vpws-evpn
 Announcement bits (1): 1-BGP_RT_Background
 AS path: I
 Communities: target:200:1 evpn-l2-info:0x0 (mtu 0)
 Route Label: 3
 Primary Routing Table: evpn-vpws.evpn.0
 Thread: junos-main
```

Now let's verify ping from CE3 to CE1:

```
regress@PE3# run ping logical-system CE3 20.4.1.1 count 10 rapid
PING 20.4.1.1 (20.4.1.1): 56 data bytes
!!!!!!!
--- 20.4.1.1 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.379/6.258/8.410/0.829 ms
```

And verify ping from CE1 to CE3:

```
regress@PE1# run ping logical-system CE1 20.4.1.3 count 10 rapid
PING 20.4.1.3 (20.4.1.3): 56 data bytes
!!!!!!!
--- 20.4.1.3 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.891/8.419/20.171/4.132 ms
```

Here's the OAM ping to End.Dx2 SID of evpn-vpws service at PE1 from PE3:

```
regress@PE1# run ping srv6 isis 2001:db8:a3:1:103:: detail count 2
Warning: License key missing; requires 'sr' license
Request for icmp_seq=1, via interface 368, packet size 64
```

```

Reply for icmp_seq=1 from 2001:db8:57::2 via interface 368 hops-limit: 62, trip-time: 6.293 ms
 Local transmit time: 2022-08-02 20:47:48 PDT 195.824 ms
 Remote receive time: 2022-08-02 20:47:48 PDT 220.056 ms
Request for icmp_seq=2, via interface 368, packet size 64
Reply for icmp_seq=2 from 2001:db8:57::2 via interface 368 hops-limit: 62, trip-time: 13.282 ms
 Local transmit time: 2022-08-02 20:47:49 PDT 194.022 ms
 Remote receive time: 2022-08-02 20:47:49 PDT 220.733 ms

--- lsping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.293/9.787/13.282/3.495 ms

```

And the OAM Ping to End.Dx2 SID of evpn-vpws service at PE3 from PE1:

```
regress@PE3# run ping srv6 isis 2001:db8:a1:1:101:: detail count 2
```

```
Warning: License key missing; requires 'sr' license
```

```

Request for icmp_seq=1, via interface 366, packet size 64
Reply for icmp_seq=1 from 2001:db8:13::1 via interface 366 hops-limit: 62, trip-time: 7.341 ms
 Local transmit time: 2022-08-02 20:50:21 PDT 84.355 ms
 Remote receive time: 2022-08-02 20:50:21 PDT 97.607 ms
Request for icmp_seq=2, via interface 366, packet size 64
Reply for icmp_seq=2 from 2001:db8:13::1 via interface 366 hops-limit: 62, trip-time: 6.078 ms
 Local transmit time: 2022-08-02 20:50:22 PDT 81.960 ms
 Remote receive time: 2022-08-02 20:50:22 PDT 94.655 ms

--- lsping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.078/6.710/7.341/0.631 ms

```

## 11.3 EVPN VPWS Active-active Multihoming Over SRv6

```

regress@PE1# run show evpn vpws-instance evpn-vpws
Instance: evpn-vpws, Instance type: EVBN VPM, Encapsulation type: SRv6
Route Distinguisher: 1.1.1.1:20
Number of local interfaces: 1 (1 up)

Interface name ESI Mode Role Status Con
ge-0/0/1:4 00:00:00:00:00:00 single-homed Primary Up No
 Local SID: 102 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:1:101:
 Remote SID: 201
 PE addr ESI Label End.Dx2 SID Mode
 2001:db8:7:255:7:7 00:00:00:00:00:00:00:00:03 2001:db8:a1:1:103: singl
Number of protect interfaces: 0

regress@PE1# run show route table evpn-vpws.evpn.0 extensive match-prefix 1:7.7.7.7:20::0::201
evpn-vpws.evpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
1:7.7.7.7:20::0::201/192 AD/EVI (1 entry, 1 announced)
*BGPR Preference: 170/-101
 Route Distinguisher: 1:7.7.7.7:20
 Next-hop: 0x7a617b4, Indirect, Next hop index: 0
 Address: 0x7a617b4
 Next-hop reference count: 2, key opaque handle: 0x0, non-key opaque handle: 0x0
 Source: 2001:db8:7:255:7:7
 Protocol: next-hop 2001:db8:7:255:7:7
 Indirect next-hop 0x7a617b4, Indirect INH Session ID: 0
 State: Secondary, Old metric: 0, New metric: 0, Int Ext
 Local As: 64512 Peer As: 64512
 Age: 5:23:59 Metric:2:3
 Validation State: unverified
 ORR Generation-ID: 0
 Task: BGP_64512_2001:db8:7:255:7:7
 Assignment bits (1): 0-evpn-vpws-evpn
 AS path: 1
 Communities: target::2001: evpn-l2-info:0x0 (mtu:0)
 Import Accepted
 SRv6 sids 2001:db8:a1:1:103: Behavior: 2) BL: 64 NL: 0 FL: 16 AL: 0 TL: 0 TO: 0
 Route label: 3
 LocalLabel: 100
 Router ID: 7.7.7.7
 ...
 End behavior L2 Service (2)

regress@PE1# run show route table inet6.3 match-prefix 2001:db8:7:255:7::7
inet6.3: 23 destinations, 23 routes (23 active, 0 holddown, 0 hidden)
* = Active route, - = Last Active, * Both

2001:db8:7:255:7:7/128
 *[Static/5] 05:13:15, metric:2 3
 > to fe80::5e04:16ff:fe00:4a1c via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11:
 to fe80::5e04:16ff:fe00:fa56 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1:11:

[edit]
regress@PE1# show routing-options ldisplay set [match 2001:db8:7:255:7::7]
set routing-options rib inet6.3 static route 2001:db8:7:255:7:7/128 next-hop 2001:db8:a3:1:11::0
set routing-options rib inet6.3 static route 2001:db8:7:255:7:7/128 resolve

```

Figure 11.4

EVPN VPWS Active-active Multihoming Over SRv6 Topology

In Figure 11.4, all active multihoming provides redundancy for vpws service. Traffic from CE can go to either of the multihomed PEs. From remote peer traffic is load balanced toward multihomed PE. To enable multihoming, the CE is connected to more than one PE. Multiple PEs connect to a single CE device with the ethernet segment having the same ESI value. ESI can be configured in all-active or single-active mode. For active-active multihoming, mode is set to all-active for Ethernet segment on multihomed PE devices.

Let's configure the same `vpws-service-id local` and `vpws-service-id remote` on multihomed PE's. Here's the configured esi on ccc interface which connects to CE.

```
set interfaces ae0 unit 0 esi 00:12:12:12:12:12:12:12:12:12
set interfaces ae0 unit 0 esi all-active
```

In order to enable multihoming on PE1 and PE2:

- Configure identical `vpws-service-id local` and `vpws-service-id remote` on PE1 and PE2 .
- Configure identical ESI value and all-active mode on interface connecting to CE12 on both PE1 and PE2.

| PE1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | PE2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>ae0 {     unit 0 {         esi {             00:12:12:12:12:12:12:12:12;             all-active;         }     }     evpn-vpws-mh {         instance-type evpn-vpws;         protocols {             evpn {                 interface ae0.0 {                     vpws-service-id {                         local 103;                         remote 301;                         source-packet-routing {                             srv6 locator loc1;                         }                     }                 }                 encapsulation srv6;             }         }         interface ae0.0;         route-distinguisher 1.1.1.1:12;         vrf-import vrf-import-evpn-vpws-mh;         vrf-export vrf-export-evpn-vpws-mh;     } }</pre> | <pre>ae0 {     unit 0 {         esi {             00:12:12:12:12:12:12:12:12;             all-active;         }     }     evpn-vpws-mh {         instance-type evpn-vpws;         protocols {             evpn {                 interface ae0.0 {                     vpws-service-id {                         local 103;                         remote 301;                         source-packet-routing {                             srv6 locator loc1;                         }                     }                 }                 encapsulation srv6;             }         }         interface ae0.0;         route-distinguisher 2.2.2.2:12;         vrf-import vrf-import-evpn-vpws-mh;         vrf-export vrf-export-evpn-vpws-mh;     } }</pre> |

```

evpn-vpws-mh {
 instance-type evpn-vpws;
 protocols {
 evpn {
 interface ge-0/0/1.1 {
 vpwsservice-id {
 local 301;
 remote 103;
 source-packet-routing {
 srv6 locator loc1;
 }
 }
 }
 encapsulation svr6;
 }
 }
 interface ge-0/0/1.1;
 route-distinguisher 81.3.3.3:12;
 vrf-import vrf-import-evpn-vpws-mh;
 vrf-export vrf-export-evpn-vpws-mh;
}

```

Figure 11.5

Use `show evpn vpws-instance` to check vpws connection. Check advertised End.Dx2 SID has a dynamically allocated value of SRv6 SID. Check advertised mode is all-active and role is primary for both PE1 and PE2:

```

regress@PE1# run show evpn vpws-instance evpn-vpws-mh
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 1.1.1.12
 Number of local interfaces: 1 (1 up)

 Interface name ESI Mode Role Status Control-Word Flow-
Label-Tx Flow-Label-Rx
 ae0.0 00:12:12:12:12:12:12:12:12:12:12:12:12:12 all-active Primary Up No Nb
No
 Local SID: 103 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:1:8000::
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:2:: 00:12:12:12:12:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:2:8000:: all-active Primary
2022-08-02 14:44:38.242 Resolved
 Remote SID: 301
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00:00:00 3 2001:db8:a3:1:8000:: single-homed
Primary 2022-08-02 11:25:56.216 Resolved
 Number of protect interfaces: 0

 Fast Convergence Information
 ESI: 00:12:12:12:12:12:12:12:12:12:12:12:12:12 Number of PE nodes: 1
 PE: 2001:db8:a1:2::
 Advertised SID: 103

```

```
regress@PE2# run show evpn vpws-instance evpn-vpws-mh
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
```

```

Route Distinguisher: 2.2.2.2:12
Number of local interfaces: 1 (1 up)

 Interface name ESI Mode Role Status Control-Word Flow-
Label-Tx Flow-Label-Rx
 ae0.0 00:12:12:12:12:12:12:12:12 all-active Primary Up No No
No
 Local SID: 103 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:2:8000::
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:1:: 00:12:12:12:12:12:12:12:12 3 2001:db8:a1:1:8000:: all-active Primary
2022-08-02 14:45:47.307 Resolved
 Remote SID: 301
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00 3 2001:db8:a3:1:8000:: single-homed Primary
2022-08-02 11:25:56.209 Resolved
 Number of protect interfaces: 0

 Fast Convergence Information
 ESI: 00:12:12:12:12:12:12:12 Number of PE nodes: 1
 PE: 2001:db8:a1:1:
 Advertised SID: 103

regress@PE3# run show evpn vpws-instance evpn-vpws-mh
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 81.3.3.3:12
 Number of local interfaces: 1 (1 up)

 Interface name ESI Mode Role Status Control-Word Flow-
Label-Tx Flow-Label-Rx
 ge-0/0/1.1 00:00:00:00:00:00:00:00:00:00 single-homed Primary Up No o N
No
 Local SID: 301 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a3:1:8000::
 Remote SID: 103
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:2:: 00:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:2:8000:: all-active Primary
2022-08-02 14:44:38.241 Resolved
 2001:db8:a1:1:: 00:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:1:8000:: all-active Primary
2022-08-02 14:45:47.309 Resolved
 Number of protect interfaces: 0

 Fast Convergence Information
 ESI: 00:12:12:12:12:12:12:12 Number of PE nodes: 2
 PE: 2001:db8:a1:2:
 Advertised SID: 103
 PE: 2001:db8:a1:1:
 Advertised SID: 103

```

Show route table l2xc.0 ccc ae0 on PE1 and PE2 shows route resolve over SRv6 tunnel with dest End-SID located on PE3. Both can do SRv6 encapsulation and send traffic to PE3:

```

egress@PE1# run show route table l2xc.0 ccc ae0
l2xc.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

```

```
+ = Active Route, - = Last Active, * = Both

ae0.0 *[EVPN/7] 00:05:45
 > to fe80::5604:16ff:fe01:2d53 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5604:16ff:fe00:561d via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::

regress@PE2# run show route table l2xc.0 ccc ae0

l2xc.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ae0.0 *[EVPN/7] 00:07:10
 to fe80::5604:16ff:fe01:3e12 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 > to fe80::5604:16ff:fe01:125a via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::

Show route table l2xc.0 ccc ge-0/0/1.1 on PE3 shows PE3 can load balance and send
traffic to both PE1 and PE2:
```

```
regress@PE3# run show route table l2xc.0 ccc ge-0/0/1.1
```

```
l2xc.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ge-0/0/1.1 *[EVPN/7] 00:07:24
 to fe80::5604:16ff:fe00:ae70 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
 to fe80::5604:16ff:fe01:1ca6 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
 to fe80::5604:16ff:fe00:ae70 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
 > to fe80::5604:16ff:fe01:1ca6 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
```

CE12 to CE3 ping:

```
regress@CE12# run ping 10.3.1.3 count 10 rapid
PING 10.3.1.3 (10.3.1.3): 56 data bytes
!!!!!!!
--- 10.3.1.3 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.139/7.859/14.141/2.490 ms
```

CE3 to CE12 ping:

```
regress@PE3# run ping logical-system CE3 10.3.1.12 rapid count 10
PING 10.3.1.12 (10.3.1.12): 56 data bytes
!!!!!!!
--- 10.3.1.12 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.924/8.098/13.554/2.069 ms
```

And OAM ping verification:

```
regress@PE2# run ping srv6 isis 2001:db8:a1:1:8000:: count 2 detail
```

Warning: License key missing; requires 'sr' license

```
Request for icmp_seq=1, via interface 364, packet size 64
Reply for icmp_seq=1 from 2001:db8:13::1 via interface 364 hops-limit: 63, trip-time: 6.145 ms
 Local transmit time: 2022-08-02 14:56:20 PDT 999.040 ms
 Remote receive time: 2022-08-02 14:56:21 PDT 148.377 ms
```

```

Request for icmp_seq=2, via interface 364, packet size 64
Reply for icmp_seq=2 from 2001:db8:13::1 via interface 364 hops-limit: 63, trip-time: 4.340 ms
 Local transmit time: 2022-08-02 14:56:21 PDT 998.950 ms
 Remote receive time: 2022-08-02 14:56:22 PDT 147.479 ms

--- lsping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.340/5.242/6.145/0.903 ms

```

## 11.4 EVPN VPWS Single-active Multihoming Over SRv6

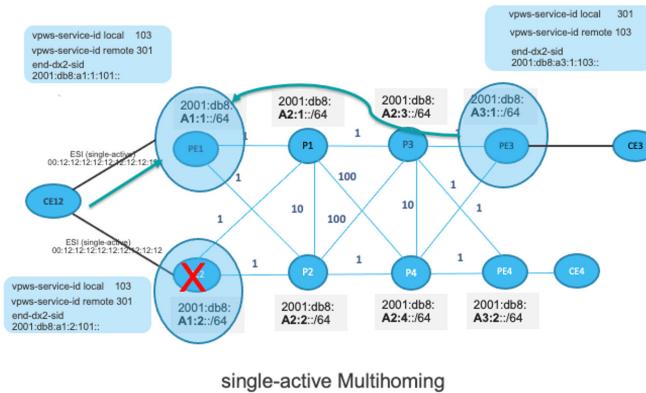


Figure 11.6 EVPN VPWS single-active Multihoming over SRv6

Single active multihoming also provides redundancy but only a single path is active at any given time. Based on the designated-election algorithm, one of the multihomed PEs is designated as primary or backup. Traffic will flow through the PE that gets elected as primary. The backup PE drops any packet sent on the CE-PE link.

In order to stop traffic from CE12 to backup PE, you need to enable LACP with `per-esi lacp-oos-on-ndf`. Also you need to have the same `system-id 00:00:00:00:00:12` configured on PE1 and PE2.

Enable LACP with the option on PE1, PE2:

```

set interfaces ae0 esi 00:12:12:12:12:12:12:12:12
set interfaces ae0 esi single-active
set interfaces ae0 esi df-election-granularity per-esi lacp-oos-on-ndf
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:00:00:12

```

On CE12 configure:

```

set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast

```

Configuration is identical to the all-active configuration. The only change will be with ESI mode set to single-active instead of all-active. These config changes over all-active is all it takes to make it single active.

PE1:

```
set interfaces ae0 unit 0 esi single-active
```

PE2:

```
set interfaces ae0 unit 0 esi single-active
```

The `show evpn vpws-instance` command captures mode and role. Check primary and backup status of multihomed PE's:

```
regress@PE2# run show evpn vpws-instance evpn-vpws-mh
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
 Route Distinguisher: 2.2.2.12
 Number of local interfaces: 1 (1 up)
```

```

Label-Tx Interface name ESI Mode Role Status Control-Word Flow-
 Flow-Label-Rx
 ae0.0 00:12:12:12:12:12:12:12:12 single-active Backup Up No o N
No
 Local SID: 103 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a1:2:8000:::
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:1:: 00:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:1:8000:: single-active
Primary 2022-08-02 14:11:38.445 Resolved
 Remote SID: 301
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a3:1:: 00:00:00:00:00:00:00:00:00:00:00 3 2001:db8:a3:1:8000:: single-homed Primary
2022-08-02 11:25:56.209 Resolved
Number of protect interfaces: 0

Fast Convergence Information
ESI: 00:12:12:12:12:12:12:12 Number of PE nodes: 1
PE: 2001:db8:a1:1::
Advertised SID: 103

DF Election Information for Single-Active ESI
ESI: 00:12:12:12:12:12:12:12:12
DF Election Algorithm: MOD based
SID used for DF Election: 103
LACP OOS on NDF: Enabled
Primary PE: abcd::128:53:34:209
ESI granularity: Per ESI
Backup PE: abcd::128:53:60:230
ESI granularity: Per ESI
Last DF Election: 2022-08-02 14:11:38

```

```
regress@PE3# run show evpn vpws-instance evpn-vpws-mh
```

```
Instance: evpn-vpws-mh, Instance type: EVPN VPWS, Encapsulation type: SRv6
Route Distinguisher: 81.3.3.3:12
Number of local interfaces: 1 (1 up)
```

```

Label-Tx Interface name ESI Mode Role Status Control-Word Flow-
 Flow-Label-Rx
 ge-0/0/1.1 00:00:00:00:00:00:00:00:00:00 single-homed Primary Up No o N
No
 Local SID: 301 Advertised Label: 3 Advertised End.Dx2 SID: 2001:db8:a3:1:8000:::
 Remote SID: 103
 PE addr ESI Label End.Dx2 SID Mode Role TS
Status
 2001:db8:a1:2:: 00:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:2:8000:: single-active Backup
2022-08-02 14:11:38.241 Resolved
 2001:db8:a1:1:: 00:12:12:12:12:12:12:12:12:12 3 2001:db8:a1:1:8000:: single-active
Primary 2022-08-02 14:11:38.445 Resolved
Number of protect interfaces: 0

Fast Convergence Information
ESI: 00:12:12:12:12:12:12:12 Number of PE nodes: 2
PE: 2001:db8:a1:2::
Advertised SID: 103
PE: 2001:db8:a1:1::
Advertised SID: 103

```

PE1 with address 2001:db8:a1:1:: is elected as Primary and PE2 with address 2001:db8:a1:2:: elected as backup. Notice on backup PE, PE2 PE-CE interface i.e ae0.0 link status will be CCC down.

```
regress@PE2# run show interfaces ae0 terse
Interface Admin Link Proto Local Remote
ae0 upup
ae0.0 updown ccc
ae0.32767 upup multiservice
```

The `show route table l2xc.0 ccc` on backup points to discard. This verifies no traffic flows via backup PE2:

```
regress@PE2# run show route table l2xc.0 ccc ae0
```

```
l2xc.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
ae0.0 *[EVPN/7] 00:08:19
 Discard
```

On primary PE PE1, CCC route resolves over SRv6 tunnel with dest 2001:db8:a3:1:: which is locator configured on PE3:

```
regress@PE1# run show route table l2xc.0 ccc ae0
```

```
l2xc.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
ae0.0 *[EVPN/7] 00:09:29
 to fe80::5604:16ff:fe01:2d53 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 > to fe80::5604:16ff:fe00:561d via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

The `show route table l2xc.0 ccc` on remote PE PE3, SRv6 tunnel destination is a locator address 2001:db8:a1:1:: configured on PE1 (Primary PE):

```
regress@PE3# run show route table l2xc.0 ccc ge-0/0/1.1
```

```
l2xc.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
ge-0/0/1.1 *[EVPN/7] 00:10:50
 to fe80::5604:16ff:fe00:ae70 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
 > to fe80::5604:16ff:fe01:1ca6 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a1:1::
```

CE12 to CE3 ping:

```
regress@CE12# run ping 10.3.1.3 count 10 rapid
PING 10.3.1.3 (10.3.1.3): 56 data bytes
!!!!!!!
--- 10.3.1.3 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.427/7.764/11.430/1.433 ms
```

CE3 to CE12 ping:

```
regress@PE3# run ping logical-system CE3 10.3.1.12 count 10 rapid
PING 10.3.1.12 (10.3.1.12): 56 data bytes
!!!!!!!
--- 10.3.1.12 ping statistics ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.438/9.183/16.797/2.842 ms
```

OAM ping verification:

```
regress@PE3# run ping srv6 isis 2001:db8:a1:1:8000:: count 2 detail
```

Warning: License key missing; requires 'sr' license

```
Request for icmp_seq=1, via interface 366, packet size 64
Reply for icmp_seq=1 from 2001:db8:13::1 via interface 366 hops-limit: 62, trip-time: 12.377 ms
 Local transmit time: 2022-08-02 14:29:27 PDT 559.734 ms
 Remote receive time: 2022-08-02 14:29:27 PDT 682.067 ms
Request for icmp_seq=2, via interface 366, packet size 64
Reply for icmp_seq=2 from 2001:db8:13::1 via interface 366 hops-limit: 62, trip-time: 9.611 ms
 Local transmit time: 2022-08-02 14:29:28 PDT 562.180 ms
 Remote receive time: 2022-08-02 14:29:28 PDT 682.695 ms

--- lsping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.611/10.994/12.377/1.383 ms
```

Let's bring the ae0 interface down on primary PE PE1 so that backup PE PE2 becomes primary, as shown in Figure 11.7.

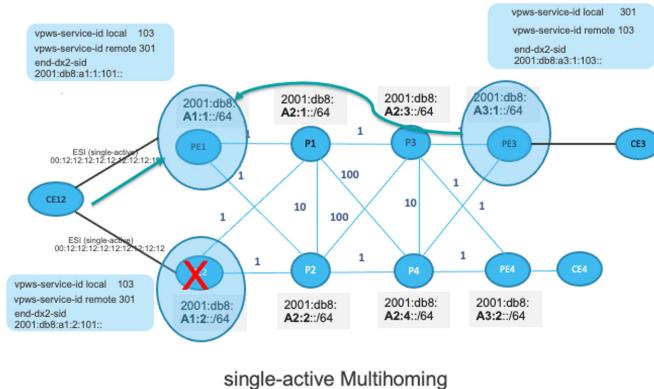


Figure 11.7 New Elected Primary PE2

Notice ae0.0 link status has changed to Up. Now traffic flows between CE12–PE2–PE3–CE3:

```
regress@PE2# run show interfaces ae0 terse
Interface Admin Link Proto Local Remote
ae0 upup
ae0.0 upup ccc
ae0.32767 upup multiservice

regress@PE2# run show route table l2xc.0 ccc ae0
l2xc.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ae0.0 *[EVPN/7] 00:01:07
 to fe80::5604:16ff:fe01:3e12 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 > to fe80::5604:16ff:fe01:125a via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

On PE3 notice tunnel destination for CCC route is the locator 2001:db8:a1:2:: configured on PE2:

```
regress@PE3# run show route table l2xc.0 ccc ge-0/0/1.1
l2xc.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

ge-0/0/1.1 *[EVPN/7] 00:02:19
 to fe80::5604:16ff:fe00:ae70 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
 > to fe80::5604:16ff:fe01:1ca6 via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
```

Check the LACP status on CE12 and PE2 when PE1 is the primary PE. The Mux state for ge-0/0/0 is Collecting distributing which is connected to primary PE1. This basically means CE12 sends traffic to PE1:

```
regress@CE12# run show lacp interfaces
Aggregated interface: ae0
 LACP state: Role Exp Def Dist Col Syn Aggr Timeout Activity
 ge-0/0/0 Actor NoNo Yes Yes Yes Yes FastActive
 ge-0/0/0 Partner NoNo Yes Yes Yes Yes FastActive
 ge-0/0/1 Actor NoNoNo No Yes Yes FastActive
 ge-0/0/1 Partner NoNo Yes Yes No Yes FastActive
 LACP protocol: Receive State Transmit State Mux State
 ge-0/0/0 Current Fast periodic Collecting distributing
 ge-0/0/1 Current Fast periodic Attached
```

```
regress@CE12# show interfaces |display set |match desc
set interfaces ge-0/0/0 description pe1-ce12
set interfaces ge-0/0/0 unit 0 description ConnectedToPE1
set interfaces ge-0/0/1 description pe2-ce12
set interfaces ge-0/0/1 unit 0 description ConnectedToPE2
set interfaces ae0 unit 0 description evpn-vpws-mh-alpha-pe1-pe2
```

```
regress@PE2# run show lacp interfaces
Aggregated interface: ae0
 LACP state: Role Exp Def Dist Col Syn Aggr Timeout Activity
 ge-0/0/0 Actor No No Yes Yes No Yes FastActive
 ge-0/0/0 Partner No No No No Yes Yes FastActive
```

```
LACP protocol: Receive State Transmit State Mux State
ge-0/0/0 Current Fast periodic Collecting distributing
```

Once we disable ae0 on primary PE1, PE2 becomes primary. Note the LACP status when PE2 becomes primary. We can see now that the Mux State field is **Collecting distributing** for interface ge-0/0/1 which is connected to the new primary, PE2. This basically means CE12 sends traffic to the new primary PE2:

```
regress@CE12# run show lacp interfaces
Aggregated interface: ae0
 LACP state: Role Exp Def Dist Col Syn Aggr Timeout Activity
 ge-0/0/0 Actor No Yes No No No Yes FastActive
 ge-0/0/0 Partner No Yes No No No Yes Fast Passive
 ge-0/0/1 Actor NoNo Yes Yes Yes Yes Yes FastActive
 ge-0/0/1 Partner NoNo Yes Yes Yes Yes Yes FastActive
 LACP protocol: Receive State Transmit State Mux State
 ge-0/0/0 Defaulted Fast periodic Detached
ge-0/0/1 Current Fast periodic Collecting distributing
```

```
regress@CE12# show interfaces |display set |match desc
set interfaces ge-0/0/0 description pe1-ce12
set interfaces ge-0/0/0 unit 0 description ConnectedToPE1
set interfaces ge-0/0/1 description pe2-ce12
set interfaces ge-0/0/1 unit 0 description ConnectedToPE2
set interfaces ae0 unit 0 description evpn-vpws-mh-alpha-pe1-pe2
```

```
regress@PE2# run show lacp interfaces
Aggregated interface: ae0
 LACP state: Role Exp Def Dist Col Syn Aggr Timeout Activity
 ge-0/0/0 Actor No No Yes Yes Yes Yes FastActive
 ge-0/0/0 Partner No No Yes Yes Yes Yes FastActive
 LACP protocol: Receive State Transmit State Mux State
 ge-0/0/0 Current Fast periodic Collecting distributing
```

# Chapter 12

## Multi-domain SRv6

### 12.1 Introduction

A service provider's network generally consists of a large number of nodes, often in the order of several thousand. The network is organized into IGP separate IGP domains such as access, aggregation, and core. There are multiple architectural options available to get end-to-end connectivity. This book discusses inter-domain route leaking options with two different flavors. ISIS Level 1 and Level 2 domains and route leaking between them is one option. Another option is by separating the flood domains by configuring multiple ISIS instances and leaking routes across the domains.

### 12.2 Multi-domain SRv6 with L1/L2

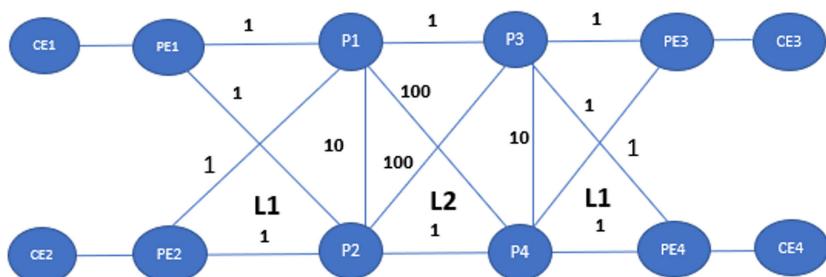


Figure 12.1 L1/L2 Multi-domain

Let's separate our lab topology into L1 and L2 domain as shown Figure 12.1. The detailed configurations are given in the Appendices. We are assuming a pre-existing L1/L2 ISIS network in this section and not covering the migration from a flat L2 domain to L1/L2 domain.

### 12.2.1 Separating the Network into L1/L2 Domains

Let's verify the separated ISIS domains by looking at the adjacencies and routes. On PE1, you can see that the ISIS adjacencies towards P1 and P2 are reporting L1 adjacencies:

```
regress@PE1# run show isis adjacency
Interface System L State Hold (secs) SNPA
ge-0/0/2.0 P1 1 Up 25
ge-0/0/3.0 P2 1 Up 24
```

On P1, the adjacencies towards P2, P3, P4 are showing as L2 and adjacencies towards PE1, PE2 are Level 1:

```
regress@P1# run show isis adjacency
Interface System L State Hold (secs) SNPA
ge-0/0/0.0 P2 2 Up 20
ge-0/0/1.0 P3 2 Up 23
ge-0/0/2.0 P4 2 Up 21
ge-0/0/3.0 PE1 1 Up 20
ge-0/0/4.0 PE2 1 Up 24
```

In ISIS, the Level 1 routes are by default leaked into Level 2. Level 2 routes are not leaked into Level 1 by default. We can verify the routes on PE1.

You can see that the loopback routes for PE1, PE2, P1, P2 are present:

```
regress@PE1# run show route table inet6.0
inet6.0: 38 destinations, 38 routes (38 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:d00::/24 *[IS-IS/15] 00:02:37, metric 3
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0
2001:db8:1:255:1::1/128
 *[Direct/0] 01:02:35
 > via lo0.0
2001:db8:2:255:2::2/128
 *[IS-IS/15] 00:02:37, metric 2
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0
2001:db8:3:255:3::3/128
 *[IS-IS/15] 00:02:41, metric 1
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0
2001:db8:4:255:4::4/128
 *[IS-IS/15] 00:02:37, metric 1
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0
....
```

The locators and End SID routes of PE1, PE2, P1, and P2 are also programmed. As there is no end-to-end connectivity, the service routes are hidden.

## 12.2.2 Leaking Summarized Locators on L1/L2

In order to get reachability to the service endpoint, we will leak the routes from Level 2 into the Level 1 domain. In SRv6, the SIDs are derived from the locators and hence the reachability to locators in each domain is sufficient to get the connectivity to all the SIDs advertised by the router. In addition, the locators from multiple domains can be summarized into a single locator while leaking. This significantly reduces the number of prefixes/locators that need to be leaked into the Level 1 domain. This ability to summarize the locators provides the high scalability to SRv6 deployments.

Let's configure aggregate routes on P1. We have /64 locators and we aggregate them into /48 on P1:

```
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
```

Then configure export policies to leak the aggregates into ISIS L1. 2001:db8:a2::/48 and 2001:db8:a3::/48 aggregates are being leaked into L1 and 2001:db8:a1::/48 is being leaked into Level 2:

```
set policy-options policy-statement leak_to_isis term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a2::/48 exact
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis term 1 to level 1
set policy-options policy-statement leak_to_isis term 1 then advertise-locator
set policy-options policy-statement leak_to_isis term 1 then accept
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis term 2 to level 2
set policy-options policy-statement leak_to_isis term 2 then accept
```

Set the export policies in ISIS:

```
set protocols isis export leak_to_isis
```

Verify that the P1 is originating the summarized locators in the database:

```
regress@P1# run show isis database P1 extensive | match Locator
SRv6 Locator: 2001:db8:a2::/64, Metric: 0, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:b2::/64, Metric: 0, MTID: 0, Flags: 0x0, Algorithm: 128
SRv6 Locator: 2001:db8:a2::/48, Metric: 10, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:a3::/48, Metric: 10, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:a2::1::/64, Metric: 0, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:b2::1::/64, Metric: 0, MTID: 0, Flags: 0x0, Algorithm: 128
SRv6 Locator: 2001:db8:a1::1::/64, Metric: 1, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:a1::2::/64, Metric: 2, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:a2::2::/64, Metric: 2, MTID: 0, Flags: 0x0, Algorithm: 0
SRv6 Locator: 2001:db8:b1::1::/64, Metric: 1, MTID: 0, Flags: 0x0, Algorithm: 128
SRv6 Locator: 2001:db8:b1::2::/64, Metric: 1, MTID: 0, Flags: 0x0, Algorithm: 128
SRv6 Locator: 2001:db8:a1::/48, Metric: 10, MTID: 0, Flags: 0x0, Algorithm: 0
```

When a route is leaked from L2 to L1, the down-bits are set. This prevents loops where a L2->L1 leaked route will get leaked back into L2 on another ABR. As we are leaking the summarized routes here, this default setting will not happen. We need to manually set the down bit for the routes being leaked into L1 via policy:

```
set policy-options policy-statement leak_to_isis term 1 then set-down-bit
```

Now we can verify the down bits are set by observing the database. 2001:db8:a2::/48 and 2001:db8:a3::/48 aggregates are being leaked into L1 and have the down bit set:

```
egress@P1# run show isis database P1 extensive level 1
IS-IS level 1 link-state database:
P1.00-00 Sequence: 0x280, Checksum: 0x96eb, Lifetime: 817 secs
 IS neighbor: PE1.00 Metric: 1
 Two-way fragment: PE1.00-00, Two-way first fragment: PE1.00-00
 P2P IPv4 Adj-SID: 300112, Weight: 0, Flags: --VL--
 P2P IPv6 Adj-SID: 300128, Weight: 0, Flags: F-VL--
 IS neighbor: PE2.00 Metric: 2
 Two-way fragment: PE2.00-00, Two-way first fragment: PE2.00-00
 P2P IPv4 Adj-SID: 300080, Weight: 0, Flags: --VL--
 P2P IPv6 Adj-SID: 300096, Weight: 0, Flags: F-VL--
 IP prefix: 1.13.1.0/24 Metric: 1 Internal Up
 IP prefix: 2.23.1.0/24 Metric: 2 Internal Up
 IP prefix: 3.34.1.0/24 Metric: 10 Internal Up
 IP prefix: 3.35.1.0/24 Metric: 10 Internal Up
 IP prefix: 3.36.1.0/24 Metric: 10 Internal Up
 V6 prefix: 2001:db8:13::/64 Metric: 1 Internal Up
 V6 prefix: 2001:db8:23::/64 Metric: 2 Internal Up
 V6 prefix: 2001:db8:34::/64 Metric: 10 Internal Up
 V6 prefix: 2001:db8:35::/64 Metric: 10 Internal Up
 V6 prefix: 2001:db8:36::/64 Metric: 10 Internal Up
 V6 prefix: 2001:db8:a2::/48 Metric: 10 External Down
 V6 prefix: 2001:db8:a2:1::/64 Metric: 0 Internal Up
 V6 prefix: 2001:db8:a3::/48 Metric: 10 External Down
```

Now we need the inet6.3 routes for the endpoints to get the service resolution.

Verify the inet6.3 routes for route to 2001:db8:a3::/48 :

```
inet6.3: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
...
2001:db8:a2::/48 *[SRV6-ISIS/14] 00:47:32, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2::
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a2::
...
2001:db8:a3::/48 *[SRV6-ISIS/14] 00:47:32, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3::
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3::
...
```

We can also verify the connectivity in the reverse direction and see that 2001:db8:a1::/48 reachability on PE3.

Inet6.3 routes on PE3:

```
regress@PE3# run show route table inet6.3
inet6.3: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a1::/48 *[SRV6-ISIS/14] 00:52:06, metric 11
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a1::
2001:db8:a2::/48 *[SRV6-ISIS/14] 00:52:06, metric 11
...
...
```

### 12.2.3 Set Anycast Via Policy

We have leaked the locator prefixes 2001:db8:a2::/48 and 2001:db8:a3::/48 via both P1 and P2. Now set the anycast flag on these aggregates via policy:

```
set policy-options policy-statement leak_to_isis term 1 then prefix-attribute-flags anycast-flag
```

The anycast flag can be verified in the database extensive output:

```
regress@PE2# run show route isis database P1 extensive
...
SRv6 Locator: 2001:db8:a2::/48, Metric: 10, MTID: 0, Flags: 0x80, Algorithm: 0
 Prefix-Attribute-Flags: 0x80(X:1,R:0,N:0,A:0)
...
...
```

### 12.2.4 SRv6 Service Prefix Resolution

Service prefix 8.88.1.0/24 has resolved on 2001:db8:a3::/48 and the service route can be seen:

```
regress@PE1> show route protocol bgp
inet.0: 25 destinations, 25 routes (25 active, 0 holddown, 0 hidden)
L3VPN_PE1_CE1.inet.0: 9 destinations, 14 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.11.1.0/24 [BGP/170] 00:39:08, localpref 100
 AS path: 100 I, validation-state: unverified
 > to 1.11.1.1 via ge-0/0/1.0
 [BGP/170] 00:27:02, localpref 100, from 2001:db8:11::1
 AS path: 100 I, validation-state: unverified
 > to 1.11.1.1 via ge-0/0/1.0
2.22.1.0/24 *[BGP/170] 00:04:19, localpref 100, from 2.2.2.2
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
 [BGP/170] 00:04:19, localpref 100, from 2001:db8:2:255:2::2
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
8.88.1.0/24 *[BGP/170] 00:01:25, localpref 100, from 2001:db8:8:255:8::8
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:25d5 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:-
 to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:-
...
...
```

The 7.77.1.0/24 service prefix has a color associated with the prefix and is resolving on the flex algo locator. As we have not leaked the flex algo locators yet, the service prefix is hidden:

```

regress@PE1# run show route protocol bgp hidden extensive
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (7 active, 0 holddown, 2 hidden)
7.77.1.0/24 (1 entry, 0 announced)
 BGP Preference: 170/-101
 Route Distinguisher: 7.7.7.7:1
 Next hop type: Unusable, Next hop index: 0
 Address: 0x747c394
 Next-hop reference count: 8, key opaque handle: 0x0, non-key opaque handle: 0x0
 Source: 2001:db8:7:255:7::7
 State: <Secondary Hidden Int Ext Changed ProtectionCand>
 Local AS: 64512 Peer AS: 64512
 Age: 3:09:38
 Validation State: unverified
 Task: BGP_64512.2001:db8:7:255:7::7
 AS path: I
 Communities: target:64512:1
 Import Accepted
 SRv6 SID: 2001:db8:b3:1:3777:0:0:0 Behavior: 19 BL: 0 NL: 0 FL: 0 AL: 0 TL: 0 TO: 0
 VPN Label: 3
 Localpref: 100
 Router ID: 128.207.209.40
 Primary Routing Table: bgp.l3vpn.0
 Thread: junos-main
 Indirect next hops: 1
 Protocol next hop: 2001:db8:b3:1:::
 Indirect next hop: 0x0 - INH Session ID: 0
77.77.77.77/32 (1 entry, 0 announced)
 BGP Preference: 170/-101
 Route Distinguisher: 7.7.7.7:1
 Next hop type: Unusable, Next hop index: 0

```

## 12.2.5 IP Ping for Service Prefix

Let's validate the service prefix reachability with both ping and traceroute:

```

regress@PE1# run ping 8.88.1.2 routing-instance L3VPN_PE1_CE1
PING 8.88.1.2 (8.88.1.2): 56 data bytes
64 bytes from 8.88.1.2: icmp_seq=0 ttl=64 time=4.090 ms
64 bytes from 8.88.1.2: icmp_seq=1 ttl=64 time=3.872 ms
64 bytes from 8.88.1.2: icmp_seq=2 ttl=64 time=4.392 ms
64 bytes from 8.88.1.2: icmp_seq=3 ttl=64 time=4.531 ms
64 bytes from 8.88.1.2: icmp_seq=4 ttl=64 time=3.413 ms
^C
--- 8.88.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.413/4.060/4.531/0.397 ms

```

```

egress@PE1# run traceroute 8.88.1.2 routing-instance L3VPN_PE1_CE1
traceroute to 8.88.1.2 (8.88.1.2), 30 hops max, 52 byte packets
 1 8.88.1.2 (8.88.1.2) 5.040 ms 4.751 ms 4.025 ms

```

Now verify the ping and traceroute on prefix 1.11.1.1 from PE3:

```
regress@PE3# run ping 1.11.1.1 routing-instance L3VPN_PE3_CE3
PING 1.11.1.1 (1.11.1.1): 56 data bytes
64 bytes from 1.11.1.1: icmp_seq=0 ttl=63 time=6.014 ms
64 bytes from 1.11.1.1: icmp_seq=1 ttl=63 time=5.147 ms
64 bytes from 1.11.1.1: icmp_seq=2 ttl=63 time=5.370 ms
64 bytes from 1.11.1.1: icmp_seq=3 ttl=63 time=4.762 ms
64 bytes from 1.11.1.1: icmp_seq=4 ttl=63 time=7.184 ms
64 bytes from 1.11.1.1: icmp_seq=5 ttl=63 time=4.771 ms
64 bytes from 1.11.1.1: icmp_seq=6 ttl=63 time=4.850 ms
^C
--- 1.11.1.1 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.762/5.443/7.184/0.821 ms
[edit]
regress@PE3# run traceroute 1.11.1.1 routing-instance L3VPN_PE3_CE3
traceroute to 1.11.1.1 (1.11.1.1), 30 hops max, 52 byte packets
1 1.11.1.2 (1.11.1.2) 3.903 ms 3.900 ms 3.493 ms
2 1.11.1.1 (1.11.1.1) 4.634 ms 5.274 ms 4.380 ms
```

## 12.2.7 Remote PE3 Going Down

In previous sections we have seen how summarization helps reduce the number of prefixes leaked and results in better scalability. The reduced scale also means a lot of granular reachability information is lost. If PE3 goes down, the summary would be still originated from P1 for 2001:db8:a3::/48 because PE4 is still up and has a contributing route 2001:db8:a3:2::/64 from PE4. The summary continues to be advertised and the service prefix will continue to point towards PE3 until all the service prefixes are withdrawn by BGP.

When there are multi-homed service prefixes, if one of the PEs goes down, the traffic should switch to the other PE at the earliest avoiding traffic loss. Due to the summarization, ingress PE1 cannot find out the loss of PE3 in IGP. This would result into slower switch to the multi-homed PE and a higher traffic loss.

Let's us bring down PE3 and observe the summary originated by P1:

```
Summary route will be seen in PE1
regress@PE1> show route 2001:db8:a3:1:11::0
inet6.0: 39 destinations, 39 routes (39 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3::/48 *[IS-IS/160] 00:18:52, metric 11
 > to fe80::5603:1ff:fe00:25d5 via ge-0/0/2.0
 to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0
inet6.3: 22 destinations, 28 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3::/48 *[SRV6-ISIS/14] 00:18:52, metric 11
 > to fe80::5603:1ff:fe00:25d5 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3::
 to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3::
```

If PE1 goes down, we can see that P1 is still originating the summary and PE1 has the summary route programmed in RIB and the service routes continue to resolve on the summary.

## 12.2.8 Both PE3/PE4 Going Down and Summary Removed

If we remove both PE3 and PE4, all routes contributing for the aggregate get deleted and hence the aggregate route also becomes deactivated. There should be at least one active contributing route that has a larger mask than the aggregate for the aggregate route to become active.

Let's simulate PE4 also going down:

```
regress@PE1> show route 2001:db8:a3:1:11::0
inet6.0: 37 destinations, 37 routes (37 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:d00::/24 *[IS-IS/15] 00:23:30, metric 3
 > to fe80::5603:1ff:fe00:5692 via ge-0/0/3.0
```

You can see that the summary route for 2001:db8:a3::/48 is no longer being advertised by P1 and there is no route in PE1 for 2001:db8:a3::/48.

## 12.2.9 Multi-domain SRv6 Flex Algo with L1/L2

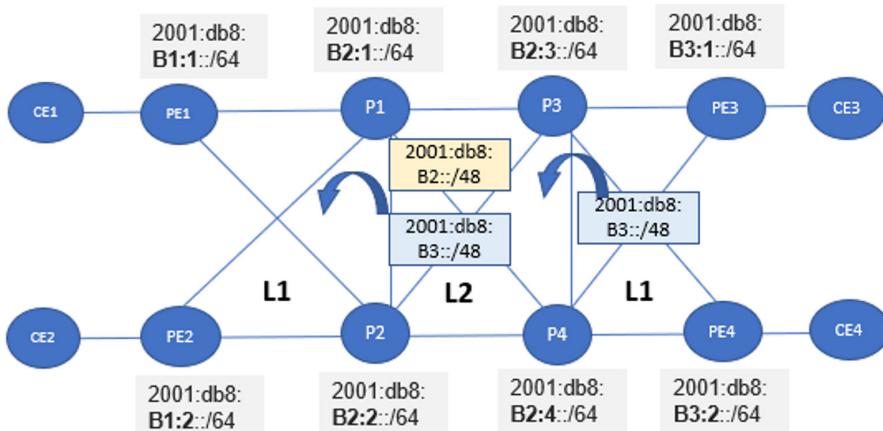


Figure 12.2 Flex Algo Locator Summarization and Leaking

Figure 12.2 shows the flex algo locators configured in Chapter 7. We summarize the locators in the rightmost L1 domain in 2001:db8:b3::/48 and leak into the core L2 domain. The core L2 domain further leaks 2001:db8:b3::/48 into the left L1 domain.

We can verify the inet6.3 table for leaked flex algo locators:

```
regress@PE1> show route table inet6.3
inet6.3: 16 destinations, 16 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
...
2001:db8:b3::/48 *[SRV6-ISIS/14] 00:00:17, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3::
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3::
```

As the flex algo protocol next hop endpoint is now reachable, colored service prefix 7.77.1.0/24 will resolve and become active:

```
regress@PE1# run show route 7.77.1.0
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
7.77.1.0/24 *[BGP/170] 00:05:21, localpref 100, from 2001:db8:7:255:7::7
 AS path: I, validation-state: unverified
 to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3::7
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3::7
```

We can also verify the reachability using service prefix ping/traceroute:

```
regress@PE1# run ping 7.77.1.2 routing-instance L3VPN_PE1_CE1
PING 7.77.1.2 (7.77.1.2): 56 data bytes
64 bytes from 7.77.1.2: icmp_seq=0 ttl=64 time=3.982 ms
64 bytes from 7.77.1.2: icmp_seq=1 ttl=64 time=4.052 ms
64 bytes from 7.77.1.2: icmp_seq=2 ttl=64 time=4.145 ms
64 bytes from 7.77.1.2: icmp_seq=3 ttl=64 time=3.871 ms
^C
--- 7.77.1.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.871/4.012/4.145/0.100 ms

regress@PE1# run traceroute 7.77.1.2 routing-instance L3VPN_PE1_CE1
traceroute to 7.77.1.2 (7.77.1.2), 30 hops max, 52 byte packets
1 7.77.1.2 (7.77.1.2) 4.890 ms 4.251 ms 4.231 ms
```

## 12.3 Multi-domain SRv6 with Multi-instance

Multi-instance ISIS is a feature where two separate instances of ISIS run on the same node. This is most commonly deployed in core/aggregation boundary nodes. Towards the core default ISIS instance runs and towards aggregation several different ISIS instances will be created based on the grouping of nodes by region or sub-region. Junos does not support running multiple instances on the same link ([RFC 8202](#)). If the link between the ABRs is required to run multiple instances then it is recommended to configure separate VLANs for each instance and run one instance of ISIS on each VLAN.

One of the characteristics of the multi-instance ISIS is that the flooding domains are completely separated and there is no default leaking across the domains as compared to the L1/L2 approach. Although flooding domains are separated, the routes from both domains go into the same default RIB. This makes it much easier to leak required prefixes across the ISIS instances. One of the main advantage of ISIS multi-instance as compared to L1/L2 is that the flooding domains are fully separated and which is not the case in L1/L2. If any L1 node is connected to an ABR which is also connected to another L1 Domain, then flooding will be done towards all such L1 domains as well. This makes flooding isolation difficult in many L1/L2 topologies.

Let's configure `metro1` ISIS instance on P1 and P2 and assign links P1->PE1, P1->PE2, P2->PE1, P2->PE2 links to the `metro1` instance. `metro1` is an independent ISIS instance and does not access the configurations from default ISIS instance. The SRv6 locators and the flex algo configurations will have to be done separately for `metro1` as shown here:

```
set protocols isis-instance metro1 source-packet-routing srv6 locator loc1
set protocols isis-instance metro1 source-packet-routing srv6 locator loc2
set routing-options flex-algorithm 128 definition metric-type delay-metric
set protocols isis-instance metro1 source-packet-routing flex-algorithm 128
```

Once the interfaces are assigned to the `metro1` instance, the ISIS adjacency comes up on the interface and can be seen an `igp-instance` specific command. ISIS adjacencies in the default instance are up towards P2, P3, and P4:

```
regress@P1# run show isis adjacency
Interface System L State Hold (secs) SNPA
ge-0/0/0.0 P2 2 Up 24
ge-0/0/1.0 P3 2 Up 18
ge-0/0/2.0 P4 2 Up 26
```

Adjacencies towards PE1 and PE2 will be seen under `igp-instance`:

```
regress@P1# run show isis adjacency igp-instance metro1
Interface System L State Hold (secs) SNPA
ge-0/0/3.0 PE1 1 Up 19
ge-0/0/4.0 PE2 1 Up 22
```

When we separate the `metro1` domain into a separate ISIS instance, it no longer sees the database of the core domain and hence will not have reachability to the nodes in core domain. We can look at the routes on PE1 in the `inet6.3` table. We can observe that the locator routes of PE2, P1, and P2 are present and there is no reachability to the nodes in default domain.

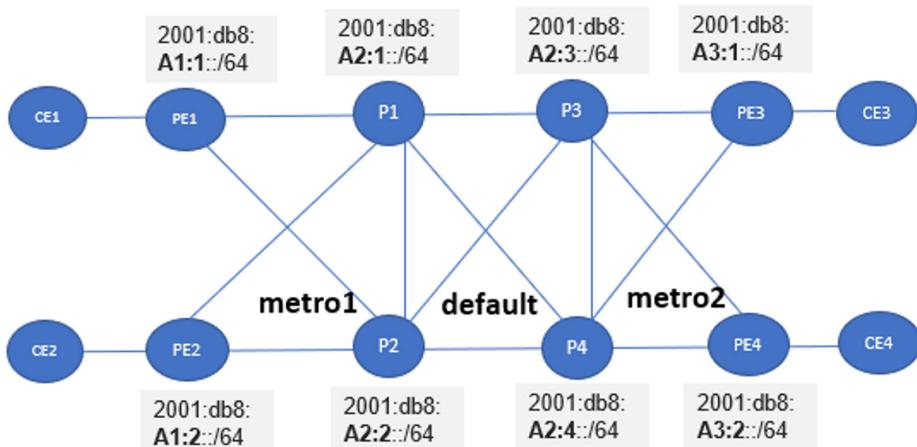


Figure 12.3 Multi-instance ISIS

Routes on PE1 after configuring multi-instance ISIS:

```
regress@PE1# run show route table inet6.3
inet6.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a1:2::/64 *[SRV6-ISIS/14] 00:58:02, metric 2
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::
2001:db8:a1:2:11::/128
 *[SRV6-ISIS/14] 00:58:02, metric 2
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2:1:1::
2001:db8:a2:1::/64 *[SRV6-ISIS/14] 00:58:26, metric 1
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:1:1::
2001:db8:a2:2::/64 *[SRV6-ISIS/14] 00:58:27, metric 1
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a2:2::
```

Similarly, we also configure metro2 ISIS instance on P3 and P4 and assign P3->PE3, P3->PE4, and P4->PE3, P4->PE4 interfaces to the metro2 instance. We also configure SRv6 locators and flex algo definition in the metro2 instance and separate the metro2 domain from the default domain. We can observe the inet6.3 table on PE3 and it shows the locators of P3, P4, PE3, PE4 are reachable and other nodes/locators in default and metro1 domain are not reachable. Routes on PE3:

```
regress@PE3# run show route table inet6.3
inet6.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a2:3::/64 *[SRV6-ISIS/14] 00:00:49, metric 1
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:3::
2001:db8:a2:4::/64 *[SRV6-ISIS/14] 00:00:19, metric 3
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a2:4:1::
2001:db8:a3:2::/64 *[SRV6-ISIS/14] 00:00:23, metric 2
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:2:1:1::
2001:db8:a3:2:11::/128
 *[SRV6-ISIS/14] 00:00:23, metric 2
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3:2:1:1::
```

### 12.3.1 Summarization and Leaking of Locators

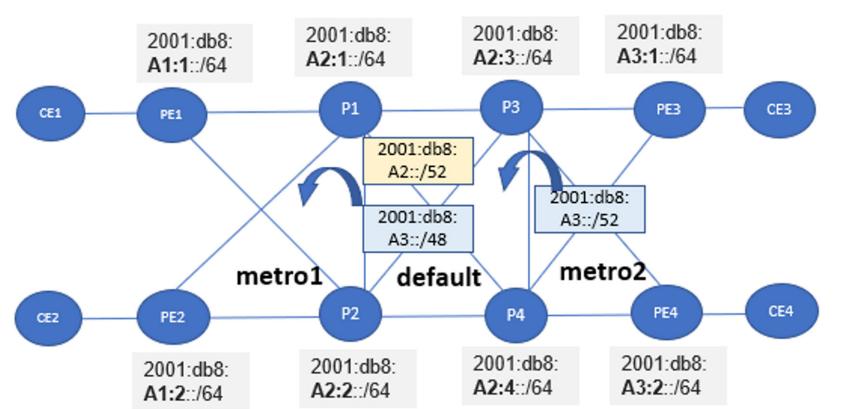


Figure 12.4 Locator Summarization and Leaking

In order to get connectivity from PE1 to PE3, we will leak the locators from metro2 domain into default domain and then further leak them into metro1 domain. We will configure aggregate routes and summarize the locators before leaking. Summarization of locators will reduce the number of leaked prefixes significantly, as described in previous sections.

Let's configure aggregate route and policies to summarize and leak 2001:db8:a3::/52 route.

Configure aggregate route on P3. We choose to aggregate the /64 locators in metro2 domain into a /52. The choice of prefix length for aggregation should be based on the number of devices that get covered by the aggregate route.

```
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/52
```

Configure policy to leak the summarized prefix and originate a locator advertisement for the prefix. The `advertise-locator` term in the `then` clause will enable this locator advertisement for the summarized route:

```
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a3::/52 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept
```

Note that we also need to leak the loopback addresses from the metro2 instance to the default instance and vice versa. The BGP session is established on these addresses and they must be reachable for BGP to keep the session in the up state:

```
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:1:255:1::1/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:2:255:2::2/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept
```

Apply the leaking policy on the ISIS protocol:

```
set protocols isis export leak_to_isis
set protocols isis-instance metro2 export leak_to_isis_metro
```

On P1, 2001:db8:a3::/48 aggregate is configured and leaked into metro1 domain. Note that the aggregate routes will become active only when there are routes that have prefixes that are under 2001:db8:a3:: with a larger prefix length. If we configure an aggregate with 2001:db8:a3::/52 length on P1, the aggregate route will not become active since the route P3/P4 originated are also of same prefix length. Let's configure a /48 length for the aggregate route:

```
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
set policy-options policy-statement leak_to_isis_metro term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis_metro term 1 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 1 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 1 then accept
```

These next configurations are done to leak loopback prefixes, which are required to bring up the BGP session between PE1, PE2 to PE3, PE4:

```
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:7:255:7::7/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:8:255:8::8/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro1 export leak_to_isis_metro
```

We can now verify on PE1 that there is a summarized locator route towards PE3:

```
egress@PE1# run show route 2001:db8:a3::
inet6.0: 32 destinations, 32 routes (32 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3::/48 *[IS-IS/165] 00:00:32, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0
inet6.3: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3::/48 *[SRV6-ISIS/14] 00:00:32, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a3::
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3::
```

Similarly, we also configure policies to summarize metro1 domain into 2001:db8:a1::/52 and leak into default domain and further summarize into 2001:db8:a1::/48 and leak into metro2 domain. We can then verify the routes on PE3 for 2001:db8:a1::/48.

Routes on PE3:

```
regress@PE3# run show route 2001:db8:a1::
inet6.0: 31 destinations, 31 routes (31 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a1::/48 *[IS-IS/160] 00:42:02, metric 11
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0
 to fe80::5603:1ff:fe00:5f42 via ge-0/0/3.0
inet6.3: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a1::/48 *[SRV6-ISIS/14] 00:42:02, metric 11
 > to fe80::5603:1ff:fe00:5489 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a1::
 to fe80::5603:1ff:fe00:5f42 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1::
```

**NOTE** We leak the prefixes on both border nodes that is P1 and P2 for metro1 and P3 and P4 for metro2. As discussed in Section 11.2.3 we can set anycast flag via policy.

### 12.3.3 SRv6 Service Prefix Resolution

Any service prefix advertised from PE3 or PE4 will have protocol next hop set to the locator of that node. When we leak the summarized locators into metro1 domain, the protocol next hop becomes reachable and service will resolve on the summary locator. We can verify service prefix state as shown here:

```
regress@PE1# run show route protocol bgp
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (7 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both
1.11.1.0/24 [BGP/170] 1w4d 10:14:24, localpref 100, from 2001:db8:11::1
 AS path: 100 I, validation-state: unverified
 > to 1.11.1.1 via ge-0/0/1.0
2.22.1.0/24 *[BGP/170] 4d 08:49:12, localpref 100, from 2001:db8:2:255:2::2
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::2
8.88.1.0/24 *[BGP/170] 00:03:50, localpref 100, from 2001:db8:8:255:8::8
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::2
11.11.11.11/32 *[BGP/170] 1w4d 10:14:24, localpref 100, from 2001:db8:11::1
 AS path: 100 I, validation-state: unverified
 > to 1.11.1.1 via ge-0/0/1.0
22.22.22.22/32 *[BGP/170] 4d 08:49:12, localpref 100, from 2001:db8:2:255:2::2
 AS path: 100 I, validation-state: unverified
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::2
88.88.88.88/32 *[BGP/170] 00:03:50, localpref 100, from 2001:db8:8:255:8::8
 AS path: 100 I, validation-state: unverified
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:a0::2
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a0::2
```

### 12.3.4 IP Ping for Service Prefix

Verify service prefix ping and traceroute to validate the service prefix reachability through the network:

```
regress@PE1# run ping 8.88.1.2 routing-instance L3VPN_PE1_CE1
PING 8.88.1.2 (8.88.1.2): 56 data bytes
64 bytes from 8.88.1.2: icmp_seq=0 ttl=64 time=5.344 ms
64 bytes from 8.88.1.2: icmp_seq=1 ttl=64 time=5.368 ms
64 bytes from 8.88.1.2: icmp_seq=2 ttl=64 time=5.455 ms
64 bytes from 8.88.1.2: icmp_seq=3 ttl=64 time=5.205 ms
^C
--- 8.88.1.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.205/5.343/5.455/0.090 ms
regress@PE1# run traceroute 8.88.1.2 routing-instance L3VPN_PE1_CE1
traceroute to 8.88.1.2 (8.88.1.2), 30 hops max, 52 byte packets
 1 8.88.1.2 (8.88.1.2) 6.295 ms 6.003 ms 5.178 ms
```

### 12.3.5 Multi-domain SRv6 Flex Algo with Multi-instance

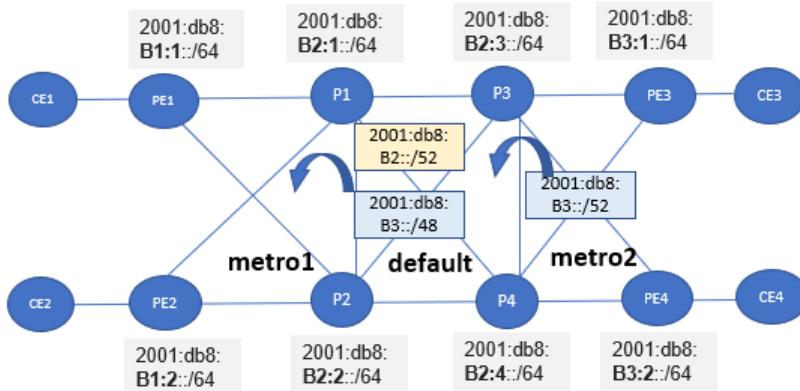


Figure 12.5 Flex Algo Locator Summarization and Leaking

In the previous section we summarized and leaked default locators. In this section we will configure aggregates for flex-algo 128 locators and leak them across domain. The configurations are very similar to what we have discussed in Section 11.3.1. The configurations for this section can be referred to in the appendices.

Once we configure the leaking policies PE1 gets reachability to PE3 through summarized locator. Service prefix 7.77.1.0 is colored and is expected to resolve on flex algo route:

```
regress@PE1# run show route table inet6.3
inet6.3: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
...
2001:db8:b3::/48 *[SRV6-ISIS/14] 00:04:17, metric 11
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3::
 to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3::
```

We can verify that 7.77.1.0/24 has resolved on flex-algo locator 2001:db8:b3::.

```
regress@PE1# run show route protocol bgp
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
L3VPN_PE1_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
1.11.1.0/24 [BGP/170] 1w5d 11:44:58, localpref 100, from 2001:db8:11::1
 AS path: 100 I, validation-state: unverified
 > to 1.11.1.1 via ge-0/0/1.0
2.22.1.0/24 *[BGP/170] 5d 10:19:46, localpref 100, from 2001:db8:2:255:2::2
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a1:2::2
7.77.1.0/24 *[BGP/170] 00:04:50, localpref 100, from 2001:db8:7:255:7::7
 AS path: I, validation-state: unverified
 > to fe80::5603:1ff:fe00:25be via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3::2
 > to fe80::5603:1ff:fe00:3248 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3::2
...
...
```

Verify ping and traceroute on service prefix to confirm the service prefix reachability:

```
regress@PE1# run ping 7.77.1.2 routing-instance L3VPN_PE1_CE1
PING 7.77.1.2 (7.77.1.2): 56 data bytes
64 bytes from 7.77.1.2: icmp_seq=0 ttl=64 time=5.567 ms
64 bytes from 7.77.1.2: icmp_seq=1 ttl=64 time=3.518 ms
64 bytes from 7.77.1.2: icmp_seq=2 ttl=64 time=3.882 ms
64 bytes from 7.77.1.2: icmp_seq=3 ttl=64 time=3.653 ms
^C
--- 7.77.1.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.518/4.155/5.567/0.826 ms
```

# Chapter 13

## cRPD Support

### 13.1 Introduction

cRPD uses Junos RPD as the control plane and a Linux kernel as the forwarding plane. Some of the use cases where cRPD is deployed are on a whitebox having merchant silicon hardware and a cRPD control plane, and on the servers where VNFs are hosted.

The system on which cRPD will be deployed must have all the necessary configuration and kernel modules installed. This includes the tunnel kernel modules and the Linux kernel configurations needed by different tunnels.

The kernel modules *must* be loaded on the host prior to launching the cRPD for tunnel forwarding to work. SRv6 is available in the mainstream Linux kernel since version 4.10.

Even though the target platform is not Junos-based, the Junos CLI and configuration management module will be used.

### 13.2 SRv6 Support

SRv6 tunnels are represented as Light Weight Tunnels (LWT) in the Linux kernel. The following functionalities are supported in cRPD:

- SRv6 locators
- End, End.X SIDS [without flavor (USD, PSP, USD) support]
- End.DT4, End.DT6 SRv6 service SIDs for VRF instances

In a CRPD environment, whenever a tunnel composite next hop needs to be installed in the kernel, a tunnel interface is created in the kernel.

Both insert and encap mode are supported in the Linux kernel. USD, PSP, and USP flavors are not supported for End SIDs in the Linux kernel and hence won't be supported in CRPD as well.

ECMP gateways should work (but weighted/frr gateways are not supported in the kernel). Maximum SRH depth would be the same as that in Junos.

Non-reduced mode for ingress SRv6 routes are not supported since the Linux kernel doesn't support that flavor yet.

In Junos by default, reduced mode is used but since Linux kernel implementation doesn't support that flavor yet, cRPD will always work on non-reduced mode irrespective of reduced/non-reduced mode in Junos. The `non-reduced mode` knob and the `insert mode` knob under `routing-options` are not available in cRPD as it has no functionality in cRPD.

Global IPv4 over SRv6 core End.DT4 is not supported with Linux kernel as of now. Hence this flavor is not supported in cRPD.

As of the writing of this document, End.DT4 SID is yet to be present in the mainline Linux kernel with full functionality. A working kernel patch is available and hence we had support with that patch including for DT4 functionality support.

### 13.2.1 SRv6 Knobs Supported in CRPD

These knobs under each hierarchy are supported in cRPD.

Locator config:

```
routing-options source-packet-routing {
 +srv6 locator <name> <locator_addr/length>
}
```

End SID config:

```
protocols isis source-packet-routing {
 +srv6 locator <name> end-sid <SID>
}
```

End-X SID config:

(P2P):

```
protocols isis {
 + interface <interface-name> level <1/2> srv6-adjacency-segment
 {unprotected} locator <name> end-x-sid <SID>
}
```

(lan):

```
protocols isis {
 + interface <interface-name> level <1/2> lan-neigbor <sys-id> srv6-adjacency-segment {unprotected}
 {locator <name> End-X-sid <SID>}
}
```

Route Preference config:

```
protocols isis {
+ srv6-preference <srv6-preference>
}
```

SRv6 is enabled through the sysctl tool in Linux kernel

- Sysctl allows modifying values for the kernel parameters
- All interfaces must set seg6\_enabled, forwarding keys

The below sysctls need to be enabled on cRPD devices:

```
sysctl net.ipv6.conf.all.seg6_enabled=1
sysctl net.ipv6.conf.all.forwarding=1
```

And the below per interface:

```
sysctl net.ipv6.conf.eth0.seg6_enabled=1 (replace eth0 with different interfaces)
```

More details are present in <https://segment-routing.org/index.php/Implementation/Configuration>

### Configuring Service SIDs

DT4 SID:

```
set routing-instances <instance_name> protocols bgp source-packet-routing srv6 locator LOC1 end-dt4-sid <SID>
```

Here end-dt4-sid specifies the exact VRF ipv4 table to look into. This is equivalent to the per-VRF VPN label in MPLS:

```
set protocols bgp source-packet-routing srv6 locator LOC1 end-dt4-sid <SID>
```

Here end-dt4-sid specifies the main IPv4 table to look into. This is equivalent to IP-v4inIPv6 decaps.

For End.DT4 SIDs you need to additionally set:

```
sysctl -wq net.vrf.strict_mode=1
```

End.DT6 SID:

```
set routing-instances <instance_name> protocols bgp source-packet-routing srv6 locator LOC1 end-dt6-sid <SID>
```

Here end-dt6-sid specifies the exact VRF ipv6 table. This is equivalent to the per-VRF VPN label in MPLS.

```
set protocols bgp source-packet-routing srv6 locator LOC1 end-dt6-sid <SID>
```

Here end-dt6-sid specifies the main ipv6 table. It's equivalent to the IPv6inIPv6 decaps.

### Configuring advertise-srv6-service:

If this knob is configured, then only the srv6 service TLV will be advertised in the BGP prefix SID attribute. The `advertise-srv6-service` knob is applicable for origination, (Re) origination. This knob is not applicable for propagation.

```
set protocols bgp group internal-peers family inet6 unicast advertise-srv6-service
```

Or:

```
set protocols bgp family inet6 unicast advertise-srv6-service
```

This is under “BGP\_AFI\_OPTIONS”

### Configuring accept-srv6-service :

If this knob is configured, then only the srv6 service TLV present in the BGP prefix SID attribute will be accepted.

```
set protocols bgp group internal-peers family inet6 unicast accept-srv6-service
```

OR

```
set protocols bgp family inet6 unicast accept-srv6-service
```

This is under “BGP\_AFI\_OPTIONS”

If we discard received srv6 service TLV present in BGP prefix SID attribute (when the above knob is not configured), then that NLRI will not be downloaded to the kernel/PFE.

The method to set BGP next hop for egress role is as mentioned in earlier chapters, but Transport Class based infrastructure is not supported yet with cRPD for SRv6.

## 13.3 LAB

The topology in Figure 13.1 will be used to demonstrate the functionality in cRPD. It's a subset of the master topology used in the earlier sections.



Fig 13.1 Topology

Okay, PE1 and P routers are CRPD devices and the rest are VMX devices. Let's show the configuration under the router on each of these devices.

CE1:

```
set system ports console log-out-on-disconnect
set interfaces ge-0/0/0 unit 0 description ConnectedToPE1
set interfaces ge-0/0/0 unit 0 family inet address 10.10.10.2/24
set interfaces ge-0/0/0 unit 0 family inet address 1.11.1.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 10:10:10::2/120
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:11::1/48
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 1.2.0.1/24
set interfaces ge-0/0/1 unit 0 family inet6 address ::ffff:1.2.0.1/120
set interfaces lo0 unit 0 family inet address 100.100.100.100/32
set interfaces lo0 unit 0 family inet address 11.11.11.11/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.1100.00
set interfaces lo0 unit 0 family inet6 address 100:100:100::100/128
set interfaces lo0 unit 0 family inet6 address 2001:db8:11:255:11::1/128
set policy-options policy-statement BGP_export term 0 from protocol static
set policy-options policy-statement BGP_export term 0 then reject
set policy-options policy-statement BGP_export term 1 from family inet6
set policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:11::1
set policy-options policy-statement BGP_export term 1 then accept
set policy-options policy-statement BGP_export term 2 from protocol direct
set policy-options policy-statement BGP_export term 2 then accept
set routing-options rib inet.0 static route 0.0.0.0/0 next-hop 1.11.1.2
set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:11::1
set routing-options router-id 11.11.11.11
set routing-options autonomous-system 100
set routing-options ipv6-router-id 2001:db8:11:255:11::11
set routing-options router-id 11.11.11.11
set routing-options autonomous-system 100
set routing-options ipv6-router-id 2001:db8:11:255:11::11
set protocols bgp group eBGPv6 type external
set protocols bgp group eBGPv6 family inet unicast local-ipv4-address 1.11.1.1
set protocols bgp group eBGPv6 family inet6 unicast
set protocols bgp group eBGPv6 export BGP_export
set protocols bgp group eBGPv6 peer-as 64512
set protocols bgp group eBGPv6 neighbor 2001:db8:11::2
```

PE1 (cRPD): In cRPD devices the following procedures are required.

After deploying the VM, create a container with cRPD:

```
docker create --name PE1_CRPD -h PE1_CRPD -it --privileged crpd:latest
docker start PE1_CRPD
```

#Make the containers as named namespaces:

```
function docker_netns() {
 pid=$(docker inspect --format '{{.State.Pid}}' $1)
 ln -sfT /proc/$pid/ns/net /var/run/netns/$1
}
mkdir /var/run/netns
docker_netns PE1_CRPD
```

#Move the eths into their containers:

```
ip link set ens3f1 netns PE1_CRPD
ip link set ens3f2 netns PE1_CRPD
ip link set ens3f3 netns PE1_CRPD
```

```
#Bring the eths up and assign ip-addresses (example).
#fabric-inf:
ip netns exec PE1_CRPD ip link set ens3f1 up
ip netns exec PE1_CRPD ip addr add 12.12.12.1/24 dev ens3f1
ip netns exec PE1_CRPD ip addr add 12:12:12::1/120 dev ens3f1

#vfr-intf (example):
ip netns exec PE1_CRPD ip link set ens3f2 up
ip netns exec PE1_CRPD ip addr add 10.10.10.1/24 dev ens3f2
ip netns exec PE1_CRPD ip addr add 10:10:10::1/120 dev ens3f2

#Assign router-id/loopback (example):
ip netns exec PE1_CRPD ip addr add 1.1.1.1/32 dev lo
ip netns exec PE1_CRPD ip addr add 1:1:1::1/128 dev lo
```

The `docker exec -it PE1_CRPD /bin/bash` is the way to get into the cRPD router prompt.

Additional addresses can be added or removed using

- `ip address add` - Add new protocol address.
- `ip address del` - Delete protocol address

The below addresses are added as suggested for the required interfaces:

```
root@PE1_CRPD# run show interfaces routing
Interface State Addresses
tunl0 Up MPLS enabled
 ISO enabled
sit0 Up MPLS enabled
 ISO enabled
 INET6 ::172.17.0.2
 INET6 ::127.0.0.1
__crpd-vrf3 Up ISO enabled
 INET enabled
 INET6 enabled
lsi Up MPLS enabled
 ISO enabled
 INET6 fe80::8c18:82ff:fe83:b275
lo.0 Up MPLS enabled
 ISO enabled
 INET 1.1.1.1
 INET 128.51.157.253
 INET6 2001:db8:1:255:1::1
 INET6 fe80::1
 ISO 47.0005.80ff.f800.0000.0108.0001.1282.2001.8110
 ISO 47.0005.0019.2168.1000
eth0 Up MPLS enabled
 ISO enabled
 INET 172.17.0.2
 INET6 fe80::42:acff:fe11:2
ens3f2 Up ISO enabled
 INET 10.10.10.1
 INET 1.11.1.2
 INET6 2001:db8:11::2
```

```

INET6 10:10:10::1
INET6 fe80::5404:16ff:fe01:5954
ens3f1 Up MPLS enabled
ISO enabled
INET 12.12.12.1
INET6 2001:db8:13::1
INET6 12:12:12::1
INET6 fe80::5404:16ff:fe01:5956

```

Then the Junos CLI configs are enabled as in other vmx routers:

```

set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.1282.2001.8110.00
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.1000.00
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 2001:db8:a1:1:1:1::0
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then accept
set policy-options policy-statement L3VPN_PE1_CE1_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE1_CE1_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement export-gre then community add gre
set policy-options policy-statement export-gre then accept
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options community L3VPN_1_COM members target:64512:1
set policy-options community gre members encapsulation:0L:11
set policy-options community ipip members encapsulation:0L:7
set policy-options community udp members encapsulation:0L:13
set routing-instances L3VPN_PE1_CE1 instance-type vrf
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet unicast local-ipv4-address
1.11.1.2
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet6 unicast
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 neighbor 2001:db8:11::1
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 advertise-prefix-sid
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 accept-prefix-sid
set routing-instances L3VPN_PE1_CE1 protocols bgp source=packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a1:1:3111:0:0:0
set routing-instances L3VPN_PE1_CE1 protocols bgp source=packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a1:1:4111:0:0:0
set routing-instances L3VPN_PE1_CE1 interface ens3f2
set routing-instances L3VPN_PE1_CE1 route-distinguisher 1.1.1.1:1
set routing-instances L3VPN_PE1_CE1 vrf-import L3VPN_PE1_CE1_import
set routing-instances L3VPN_PE1_CE1 vrf-export L3VPN_PE1_CE1_export
set routing-instances L3VPN_PE1_CE1 vrf-table-label
set routing-options source=packet-routing srv6 locator loc1 2001:db8:a1:1::/64
set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options autonomous-system 64512
set routing-options ipv6-router-id 2001:db8:1:255:1::1
set routing-options forwarding-table export LBPP
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop

```

```

set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
set protocols bgp group ibgpv6 family inet6 unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
set protocols isis interface all
set protocols isis interface ens3f1 level 2 srv6-adjacency-segment unprotected locator loc1 end-x-sid
2001:db8:a1:1:213::0
set protocols isis interface ens3f1 level 2 metric 1
set protocols isis interface ens3f1 point-to-point
set protocols isis interface lo.0
set protocols isis interface lo0.0
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:11::0
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:12::0
set protocols isis level 1 disable

```

Here also the initial configuration on the cRPD container is similar to PE1 (with the changes in names and IP addresses):

```

root@P_CRPD# run show interfaces routing
Interface State Addresses
ens3f2 Up ISO enabled
 MPLS enabled
 INET 23.23.23.1
 INET6 23:23:23::1
 INET6 fe80::5404:16ff:fe01:5959
 INET6 2001:db8:57::1
ens3f1 Up ISO enabled
 MPLS enabled
 INET 12.12.12.2
 INET6 12:12:12::2
 INET6 fe80::5404:16ff:fe01:5947
 INET6 2001:db8:13::2
irb Up ISO enabled
 MPLS enabled
 INET6 fe80::b84d:8eff:fea5:8dbf
tunl0 Up ISO enabled
 MPLS enabled
sit0 Up ISO enabled
 INET6 ::172.17.0.2
 INET6 ::127.0.0.1
 MPLS enabled
lsi Up ISO enabled
 INET6 fe80::7c43:85ff:fe21:46c5
 MPLS enabled
lo.0 Up MPLS enabled
 ISO enabled
 INET6 fe80::1
 INET 2.2.2.2
 INET6 2:2:2::2
 ISO 47.0005.80ff.f800.0000.0108.0001.1282.2001.8140
 ISO 47.0005.80ff.f800.0000.0108.0001.1282.2001.8111
 INET6 2001:db8:3:255:3::3
ip6tnl0 Up MPLS enabled
 ISO enabled
 INET6 fe80::4c78:65ff:fed2:6705

```

```

gretap0 Down ISO enabled
 MPLS enabled
gre0 Up ISO enabled
 MPLS enabled
eth0 Up MPLS enabled
 ISO enabled
 INET 172.17.0.2
 INET6 fe80::42:acff:fe11:2
erspan0 Down ISO enabled
 MPLS enabled

```

```

root@P_CRPD# show | display set
set version "20220114.015550_devpublish.r1234101 [_devpublish]"
set system root-authentication encrypted-password "<actual password here>"
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.1282.2001.8140.00
set interfaces lo0 unit 0 family iso address 47.0005.80ff.f800.0000.0108.0001.1282.2001.8111.00
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a2:1::/64
set routing-options ipv6-router-id 2001:db8:3:255:3::3
set protocols isis interface ens3f1 level 2 metric 10
set protocols isis interface ens3f1 point-to-point
set protocols isis interface ens3f2 level 2 metric 1
set protocols isis interface ens3f2 point-to-point
set protocols isis interface lo0.0
set protocols isis level 1 disable

```

PE2:

```

set interfaces ge-0/0/0 unit 0 description ConnectedToP
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:57::2/64
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description ConnectedToCE2
set interfaces ge-0/0/1 unit 0 family inet address 7.77.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:77::2/48
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.7000.00
set interfaces lo0 unit 0 family inet6 address 2001:db8:7:255:7::7/128
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then next-hop 2001:db8:a3:1:11::0
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then accept
set policy-options policy-statement L3VPN_PE2_CE2_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE2_CE2_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options policy-statement reject-all then reject
set policy-options community L3VPN_1_COM members target:64512:1
set policy-options community gre members encapsulation:0L:11
set policy-options community ipip members encapsulation:0L:7
set policy-options community udp members encapsulation:0L:13
set routing-instances L3VPN_PE2_CE2 instance-type vrf
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 7.77.1.2
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 family inet6 unicast
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 neighbor 2001:db8:77::1
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 advertise-prefix-sid

```

```

set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 accept-prefix-sid
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3777:0:0:0
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:1:4777:0:0:0
set routing-instances L3VPN_PE2_CE2 interface ge-0/0/1.0
set routing-instances L3VPN_PE2_CE2 route-distinguisher 7.7.7.7:1
set routing-instances L3VPN_PE2_CE2 vrf-import L3VPN_PE2_CE2_import
set routing-instances L3VPN_PE2_CE2 vrf-export L3VPN_PE2_CE2_export
set routing-instances L3VPN_PE2_CE2 vrf-table-label
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:1::/64
set routing-options rib inet6.0 static route ::123.1.1.1/128 discard
set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options router-id 81.2.2.2
set routing-options autonomous-system 64512
set routing-options ipv6-router-id 2001:db8:7:255:7::7
set routing-options static route 123.1.1.1/32 discard
set routing-options forwarding-table export LBPP
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:7:255:7::7
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
set protocols bgp multipath
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:1:275::0
set protocols isis interface ge-0/0/0.0 level 2 metric 1
set protocols isis interface ge-0/0/0.0 point-to-point
set protocols isis interface all level 2 disable
set protocols isis interface lo0.0
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:1:11::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:1:12::0
set protocols isis level 1 disable

```

## CE2:

```

set system ports console log-out-on-disconnect
set interfaces ge-0/0/0 unit 0 description ConnectedToPE3
set interfaces ge-0/0/0 unit 0 family inet address 7.77.1.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:77::1/48
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 family inet address 2.2.0.1/24
set interfaces ge-0/0/1 unit 0 family inet6 address ::ffff:2.2.0.1/120
set interfaces lo0 unit 0 family inet address 77.77.77.77/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.7700.00
set interfaces lo0 unit 0 family inet6 address 2001:db8:77:255:77::77/128
set policy-options policy-statement BGP_export term 0 from protocol static
set policy-options policy-statement BGP_export term 0 then reject

```

```

set policy-options policy-statement BGP_export term 1 from family inet6
set policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:77::1
set policy-options policy-statement BGP_export term 1 then accept
set policy-options policy-statement BGP_export term 2 from protocol direct
set policy-options policy-statement BGP_export term 2 then accept
set routing-options rib inet.0 static route 0.0.0/0 next-hop 7.77.1.2
set routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:77::1
set routing-options router-id 77.77.77.77
set routing-options autonomous-system 100
set routing-options ipv6-router-id 2001:db8:77:255:77::77
set protocols bgp group eBGPv6 type external
set protocols bgp group eBGPv6 family inet unicast local-ipv4-address 7.77.1.1
set protocols bgp group eBGPv6 family inet6 unicast
set protocols bgp group eBGPv6 export BGP_export
set protocols bgp group eBGPv6 peer-as 64512
set protocols bgp group eBGPv6 neighbor 2001:db8:77::2

```

### 13.3.1 Traffic Between CE1 and CE2

Once the configurations are applied you can see that the topology is in action. There is an eBGP session between adjacent CE and PE devices and an iBGP session between PE1 and PE2. ISIS runs between PE1 and PE2.

CE1 - PE1 (crpd) - P(crpd) - PE2 - CE2

CE1: You can see the eBGP session on CE1.

```
root@CE1_REL# run show bgp summary
```

```

Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0
 0 0 0 0 0 0
inet6.0
 1 1 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/
Accepted/Damped...
2001:db8:11::2 64512 12350 12391 0 3 3d 20:55:15 Establ
 inet.0: 0/0/0/0
 inet6.0: 1/1/1/0

```

PE1: It has an ISIS session with a P router:

```
root@PE1_CRPD# run show isis adjacency
Interface System L State Hold (secs) SNPA
ens3f1 P_CRPD 2 Up 23
```

It has an eBGP session with CE1 and iBGP session with PE2:

```
[edit]
root@PE1_CRPD# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
```

```

Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 0 0 0 0 0 0
inet6.0 0 0 0 0 0 0
bgp.l3vpn.0 6 6 0 0 0 0
bgp.l3vpn-inet6.0 5 5 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/
Accepted/Damped...
2001:db8:7:255:7::7 64512 12353 12358 0 0 3d 20:56:05 Establ
 inet.0: 0/0/0/0
 inet6.0: 0/0/0/0
 bgp.l3vpn.0: 6/6/6/0
 bgp.l3vpn-inet6.0: 5/5/5/0
 L3VPN_PE1_CE1.inet.0: 5/6/6/0
 L3VPN_PE1_CE1.inet6.0: 5/5/5/0
2001:db8:11::1 100 12393 12350 0 0 3d 20:56:19 Establ
 L3VPN_PE1_CE1.inet.0: 6/8/8/0
 L3VPN_PE1_CE1.inet6.0: 5/7/7/0

```

The P router has ISIS adjacencies with PE1 and PE2:

```

root@P_CRPD# run show isis adjacency
Interface System L State Hold (secs) SNPA
ens3f1 PE1_CRPD 2 Up 21
ens3f2 PE2_RE 2 Up 23

```

PE2:

It has an ISIS adjacency with P router:

```

root@PE2_RE# run show isis adjacency
Interface System L State Hold (secs) SNPA
ge-0/0/0.0 P_CRPD 2 Up 26

```

It has iBGP session with PE1 and eBGP session with CE2:

```

root@PE2_RE# run show bgp summary
Warning: License key missing; requires 'bgp' license
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0 0 0 0 0 0 0
inet6.0 0 0 0 0 0 0
bgp.l3vpn.0 8 8 0 0 0 0
bgp.l3vpn-inet6.0 7 7 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/

```

```
Accepted/Damped...
2001:db8:1:255:1::1 64512 12363 12356 0 0 3d 20:57:28 Establ
inet.0: 0/0/0/0
inet6.0: 0/0/0/0
bgp.l3vpn.0: 8/8/8/0
bgp.l3vpn-inet6.0: 7/7/7/0
L3VPN_PE2_CE2.inet.0: 7/8/8/0
L3VPN_PE2_CE2.inet6.0: 7/7/7/0
2001:db8:77::1 100 12398 12354 0 0 3d 20:58:29 Establ
L3VPN_PE2_CE2.inet.0: 5/6/6/0
L3VPN_PE2_CE2.inet6.0: 4/5/5/0
```

CE2: It has eBGP session with PE2:

```
[edit]
root@CE2_RE# run show bgp summary
Warning: License key missing; requires 'bgp' license
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 1 Down peers: 0
Table Tot Paths Act Paths Suppressed History Damp State Pending
inet.0
inet6.0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/Received/
Accepted/Damped...
2001:db8:77::2 64512 12351 12392 0 1 3d 20:56:32 Establ
inet.0: 0/0/0/0
inet6.0: 2/2/2/0
```

Ping flow: CE1 to CE2 loopback address ping is performed as shown:

```
[edit]
root@CE1_RE# run ping 2001:db8:77:255:77::77
PING6(56=40+8+8 bytes) 2001:db8:11::1 --> 2001:db8:77:255:77::77
16 bytes from 2001:db8:77:255:77::77, icmp_seq=0 hlim=62 time=5.369 ms
16 bytes from 2001:db8:77:255:77::77, icmp_seq=1 hlim=62 time=4.830 ms
^C
--- 2001:db8:77:255:77::77 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 4.830/5.099/5.369/0.270 ms

root@CE1_RE# run show route 2001:db8:77:255:77::77
inet6.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:77::/48 *[BGP/170] 3d 20:34:48, localpref 100
 AS path: 64512 I, validation-state: unverified
 > to 2001:db8:11::2 via ge-0/0/0.0
```

You can see that the route is pointing to PE1 at CE1.

PE1: Here you can see that the route is pointing to SRv6 encapsulation using SRv6 DT6 Service SID 2001:db8:a3:1:4777:0:0:0. A new IPv6 encapsulation with SRH having the service SID 2001:db8:a3:1:4777:0:0:0 is created:

```

root@PE1_CRPD# run show route 2001:db8:77:255:77::77
L3VPN_PE1_CE1.inet6.0: 16 destinations, 18 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:77:255:77::77/128
 *[BGP/170] 3d 20:40:29, localpref 100, from 2001:db8:7:255:7::7
 AS path: 100 I, validation-state: unverified
 > to fe80::5404:16ff:fe01:5947 via ens3f1, SRV6-Tunnel, Dest: 2001:db8:a3:1:11::

[edit]
root@PE1_CRPD# run show route 2001:db8:77:255:77::77 extensive
L3VPN_PE1_CE1.inet6.0: 16 destinations, 18 routes (16 active, 0 holddown, 0 hidden)
2001:db8:77:255:77::77/128 (1 entry, 1 announced)
TSI:
KRT-MFS Advt. to KRT-Netlink
KRT-Netlink in-kernel 2001:db8:77:255:77::77/128 -> {composite(-)}
 *BGP Preference: 170/-101
 Route Distinguisher: 7.7.7.7:1
 Next hop type: Indirect, Next hop index: 0
 Address: 0x55f85d824cdc
 Next-hop reference count: 15
 Source: 2001:db8:7:255:7::7
 Next hop type: Chain, Next hop index: 0
 Next hop: via Chain Tunnel Composite, SRV6
 Next hop: ELNH Address 0x55f85d8231dc, selected
 SRV6-Tunnel: Reduced-SRH Encap-mode
 Src: 2001:db8:1:255:1::1 Dest: 2001:db8:a3:1:11::
 Segment-list[0] 2001:db8:a3:1:11::
 Next hop type: Router, Next hop index: 0
 Address: 0x55f85d8231dc
 Next-hop reference count: 18
 Next hop: fe80::5404:16ff:fe01:5947 via ens3f1
 Protocol next hop: 2001:db8:a3:1:11::
 Composite next hop: 0x55f85e31bb60 - INH Session ID: 0x0
 Indirect next hop: 0x55f85d8e3008 - INH Session ID: 0x0
 State: <Secondary Active Int Ext ProtectionCand>
 Local AS: 64512 Peer AS: 64512
 Age: 3d 20:44:13 Metric2: 2
 Validation State: unverified
 ORR Generation-ID: 0
 Task: BGP_64512,2001:db8:7:255:7::7
 Announcement bits (2): 1-KRT MFS 2-KRT
 AS path: 100 I
 Communities: target:64512:1
 Import Accepted
 SRv6 SID: 2001:db8:a3:1:4777::
 VPN Label: 3
 Localpref: 100
 Router ID: 81.2.2.2
 Primary Routing Table: bgp.l3vpn-inet6.0
 Thread: junos-main
 Composite next hops: 1
 Protocol next hop: 2001:db8:a3:1:11:: Metric: 2
 Composite next hop: 0x55f85e31bb60 - INH Session ID: 0x0
 Indirect next hop: 0x55f85d8e3008 - INH Session ID: 0x0
 Indirect path forwarding next hops: 1
 Next hop type: Chain

```

```

Next hop: fe80::5404:16ff:fe01:5947 via ens3f1
2001:db8:a3:1:11::/128 Originating RIB: inet6.3
 Metric: 2 Node path count: 1
 Forwarding nexthops: 1
 Next hop type: Chain
 Next hop: fe80::5404:16ff:fe01:5947 via ens3f1

```

And now you can see that the vrf prefix (CE2 loopback) received from remote iBGP peer PE2 has SRv6 SID in the advertisement:

```

L3VPN_PE1_CE1.inet6.0: 16 destinations, 18 routes (16 active, 0 holddown, 0 hidden)
 Prefix Nexthop MED Lclpref AS path
* ::ffff:2.2.0.0/120 2001:db8:a3:1:11:: 100 100 I
* 2001:db8:77::/48 2001:db8:a3:1:11:: 100 I
 2001:db8:77:255:77::/128
* 2001:db8:a3:1:11:: 100 100 I
 abcd::128:53:49:130/128
* 2001:db8:a3:1:11:: 100 100 I
 abcd::128:53:74:209/128
* 2001:db8:a3:1:11:: 100 100 I

* 2001:db8:77:255:77::/128 (1 entry, 1 announced)
 Import Accepted
 Route Distinguisher: 7.7.7.7:1
 VPN Label: 3
 Nexthop: 2001:db8:a3:1:11::
 Localpref: 100
 AS path: 100 I
 Communities: target:64512:1
 SRv6 SID: 2001:db8:a3:1:4777::


```

P: At transit nodes the traffic will be forwarded based on locator (belongs to the service SID):

```

root@P_CRPD# run show route 2001:db8:a3:1:4777:0:0:0
inet6.0: 28 destinations, 30 routes (28 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1::/64 *[IS-IS/18] 3d 20:50:48, metric 1
 > to fe80::5604:16ff:fe01:595e via ens3f2


```

PE2: The DT6 SID results in the decap of the additional header and lookup of the inner packet having destination address 2001:db8:77:255:77::77 in the vrf table. Then the packet gets forwarded towards CE2:

```

[edit]
root@PE2_RE# run show route 2001:db8:a3:1:4777:0:0:0
inet6.0: 22 destinations, 22 routes (22 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:a3:1:4777::/128
 *[BGP/170] 3d 20:53:08
 AS path: I, validation-state: unverified
 to table L3VPN_PE2_CE2.inet6.0

[edit]
root@PE2_RE# run show route 2001:db8:77:255:77::77
L3VPN_PE2_CE2.inet6.0: 15 destinations, 16 routes (15 active, 0 holddown, 0 hidden)


```

```
+ = Active Route, - = Last Active, * = Both
2001:db8:77:255:77::77/128
 *[BGP/170] 3d 20:53:22, localpref 100
 AS path: 100 I, validation-state: unverified
 > to 2001:db8:77::1 via ge-0/0/1.0
```

CE2: Here we can see that the destination is on the loopback:

```
root@CE2_RE# run show route 2001:db8:77:255:77::77
inet6.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:77:255:77::77/128
 *[Direct/0] 1w2d 22:37:08
 > via lo0.0
```

# Appendices: Lab Configs

The Appendices numbering matches the book chapter numbering.

## Appendix 1: Preparing Network for SRv6 Migration

### 1.1 Base-config:

PE1:

```
set interfaces ge-0/0/1 unit 0 description ConnectedToCE1
set interfaces ge-0/0/1 unit 0 family inet address 1.11.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToP1
set interfaces ge-0/0/3 unit 0 family inet address 1.13.1.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToP2
set interfaces ge-0/0/4 unit 0 family inet address 1.14.1.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.1000.00

set logical-systems CE1 interfaces ge-0/0/0 unit 0 description ConnectedToPE1
set logical-systems CE1 interfaces ge-0/0/0 unit 0 family inet address 1.11.1.1/24
set logical-systems CE1 interfaces ge-0/0/0 unit 0 family iso
set logical-systems CE1 interfaces ge-0/0/0 unit 0 family mpls
set logical-systems CE1 interfaces lo0 unit 1 family inet address 11.11.11.11/32
set logical-systems CE1 interfaces lo0 unit 1 family iso address 47.0005.0019.2168.1100.00
set logical-systems CE1 routing-options router-id 11.11.11.11

set routing-options router-id 1.1.1.1
```

PE2:

```
set interfaces ge-0/0/2 unit 0 description ConnectedToCE2
set interfaces ge-0/0/2 unit 0 family inet address 2.22.1.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToP1
set interfaces ge-0/0/3 unit 0 family inet address 2.23.1.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToP2
set interfaces ge-0/0/4 unit 0 family inet address 2.24.1.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 2.2.2.2/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.2000.00

set logical-systems CE2 interfaces ge-0/0/1 unit 0 description ConnectedToPE2
set logical-systems CE2 interfaces ge-0/0/1 unit 0 family inet address 2.22.1.1/24
set logical-systems CE2 interfaces ge-0/0/1 unit 0 family iso
set logical-systems CE2 interfaces ge-0/0/1 unit 0 family inet6 address 2:22:1::1/64
set logical-systems CE2 interfaces ge-0/0/1 unit 0 family mpls
set logical-systems CE2 interfaces lo0 unit 1 family inet address 22.22.22.22/32
set logical-systems CE2 interfaces lo0 unit 1 family iso address 47.0005.0019.2168.2200.00
set logical-systems CE2 routing-options router-id 22.22.22.22

set routing-options router-id 2.2.2.2
```

PE3:

```
set interfaces ge-0/0/1 unit 0 description ConnectedToCE3
set interfaces ge-0/0/1 unit 0 family inet address 7.77.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToP3
set interfaces ge-0/0/3 unit 0 family inet address 5.57.1.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToP4
set interfaces ge-0/0/4 unit 0 family inet address 6.67.1.2/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 7.7.7.7/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.7000.00

set logical-systems CE3 interfaces ge-0/0/0 unit 0 description ConnectedToPE3
set logical-systems CE3 interfaces ge-0/0/0 unit 0 family inet address 7.77.1.1/24
set logical-systems CE3 interfaces ge-0/0/0 unit 0 family iso
set logical-systems CE3 interfaces ge-0/0/0 unit 0 family mpls
set logical-systems CE3 interfaces lo0 unit 1 family inet address 77.77.77.77/32
set logical-systems CE3 interfaces lo0 unit 1 family iso address 47.0005.0019.2168.7700.00
set logical-systems CE3 routing-options router-id 77.77.77.77

set routing-options router-id 7.7.7.7
```

PE4:

```

set interfaces ge-0/0/1 unit 0 description ConnectedToCE3
set interfaces ge-0/0/1 unit 0 family inet address 8.88.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToP3
set interfaces ge-0/0/3 unit 0 family inet address 5.58.1.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToP4
set interfaces ge-0/0/4 unit 0 family inet address 6.68.1.2/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 8.8.8.8/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.8000.00

set logical-systems CE4 interfaces ge-0/0/2 unit 0 description ConnectedToPE4
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family inet address 8.88.1.1/24
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family iso
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family mpls
set logical-systems CE4 interfaces lo0 unit 1 family inet address 88.88.88.88/32
set logical-systems CE4 interfaces lo0 unit 1 family iso address 47.0005.0019.2168.8800.00
set logical-systems CE4 routing-options router-id 88.88.88.88

set routing-options router-id 8.8.8.8

```

P1:

```

set interfaces ge-0/0/0 unit 0 description ConnectedToP2
set interfaces ge-0/0/0 unit 0 family inet address 3.34.1.1/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description ConnectedToP3
set interfaces ge-0/0/1 unit 0 family inet address 3.35.1.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 description ConnectedToP4
set interfaces ge-0/0/2 unit 0 family inet address 3.36.1.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToPE1
set interfaces ge-0/0/3 unit 0 family inet address 1.13.1.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToPE2
set interfaces ge-0/0/4 unit 0 family inet address 2.23.1.2/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 3.3.3.3/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.3000.00
set routing-options router-id 3.3.3.3

```

P2:

```

set interfaces ge-0/0/0 unit 0 description ConnectedToP1
set interfaces ge-0/0/0 unit 0 family inet address 3.34.1.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls

```

```
set interfaces ge-0/0/1 unit 0 description ConnectedToP3
set interfaces ge-0/0/1 unit 0 family inet address 4.45.1.1/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 description ConnectedToP4
set interfaces ge-0/0/2 unit 0 family inet address 4.46.1.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToPE1
set interfaces ge-0/0/3 unit 0 family inet address 1.14.1.2/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToPE2
set interfaces ge-0/0/4 unit 0 family inet address 2.24.1.2/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 4.4.4.4/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.4000.00
set routing-options router-id 4.4.4.4
```

P3:

```
set interfaces ge-0/0/0 unit 0 description ConnectedToP1
set interfaces ge-0/0/0 unit 0 family inet address 3.35.1.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description ConnectedToP2
set interfaces ge-0/0/1 unit 0 family inet address 4.45.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 description ConnectedToP4
set interfaces ge-0/0/2 unit 0 family inet address 5.56.1.1/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToPE3
set interfaces ge-0/0/3 unit 0 family inet address 5.57.1.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToPE4
set interfaces ge-0/0/4 unit 0 family inet address 5.58.1.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 5.5.5.5/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.5000.00

set routing-options router-id 5.5.5.5
```

P4:

```
set interfaces ge-0/0/0 unit 0 description ConnectedToP1
set interfaces ge-0/0/0 unit 0 family inet address 3.36.1.2/24
set interfaces ge-0/0/0 unit 0 family iso
set interfaces ge-0/0/0 unit 0 family mpls
set interfaces ge-0/0/1 unit 0 description ConnectedToP2
set interfaces ge-0/0/1 unit 0 family inet address 4.46.1.2/24
set interfaces ge-0/0/1 unit 0 family iso
set interfaces ge-0/0/1 unit 0 family mpls
set interfaces ge-0/0/2 unit 0 description ConnectedToP3
```

```

set interfaces ge-0/0/2 unit 0 family inet address 5.56.1.2/24
set interfaces ge-0/0/2 unit 0 family iso
set interfaces ge-0/0/2 unit 0 family mpls
set interfaces ge-0/0/3 unit 0 description ConnectedToPE3
set interfaces ge-0/0/3 unit 0 family inet address 6.67.1.1/24
set interfaces ge-0/0/3 unit 0 family iso
set interfaces ge-0/0/3 unit 0 family mpls
set interfaces ge-0/0/4 unit 0 description ConnectedToPE4
set interfaces ge-0/0/4 unit 0 family inet address 6.68.1.1/24
set interfaces ge-0/0/4 unit 0 family iso
set interfaces ge-0/0/4 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 6.6.6.6/32
set interfaces lo0 unit 0 family iso address 47.0005.0019.2168.6000.00

set routing-options router-id 6.6.6.6

```

## 1.2 Existing Legacy Network config:

```

PE1:
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.1
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet-vpn unicast extended-nexthop
set protocols bgp group ibgp family inet6-vpn unicast
set protocols bgp group ibgp neighbor 2.2.2.2
set protocols bgp group ibgp neighbor 7.7.7.7
set protocols bgp group ibgp neighbor 8.8.8.8
set protocols bgp multipath
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable

set logical-systems CE1 protocols bgp group eBGPv4 type external
set logical-systems CE1 protocols bgp group eBGPv4 export BGP_export
set logical-systems CE1 protocols bgp group eBGPv4 peer-as 64512
set logical-systems CE1 protocols bgp group eBGPv4 neighbor 1.11.1.2
set logical-systems CE1 policy-options policy-statement BGP_export term 0 from protocol direct
set logical-systems CE1 policy-options policy-statement BGP_export term 0 from interface lo0.1
set logical-systems CE1 policy-options policy-statement BGP_export term 0 then accept
set logical-systems CE1 routing-options rib inet.0 static route 0.0.0.0/0 next-hop 1.11.1.2
set logical-systems CE1 routing-options router-id 11.11.11.11
set logical-systems CE1 routing-options autonomous-system 100

set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 1.1.1.1
set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then accept

```

```

set policy-options policy-statement L3VPN_PE1_CE1_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE1_CE1_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options community L3VPN_1_COM members target:64512:1

set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options autonomous-system 64512
set routing-options forwarding-table export LBPP

```

PE2:

```

set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 2.2.2.2
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet-vpn unicast extended-nexthop
set protocols bgp group ibgp family inet6-vpn unicast
set protocols bgp group ibgp neighbor 1.1.1.1
set protocols bgp group ibgp neighbor 7.7.7.7
set protocols bgp group ibgp neighbor 8.8.8.8
set protocols bgp multipath
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable

set policy-options policy-statement L3VPN_PE1_CE1_export term 0 then next-hop 2.2.2.2
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then accept
set policy-options policy-statement L3VPN_PE2_CE2_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE2_CE2_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options community L3VPN_1_COM members target:64512:1

set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options autonomous-system 64512
set routing-options forwarding-table export LBPP

```

```

set logical-systems CE2 protocols bgp group eBGPv4 type external
set logical-systems CE2 protocols bgp group eBGPv4 export BGP_export
set logical-systems CE2 protocols bgp group eBGPv4 peer-as 64512
set logical-systems CE2 protocols bgp group eBGPv4 neighbor 2.22.1.2
set logical-systems CE2 policy-options policy-statement BGP_export term 0 from protocol direct
set logical-systems CE2 policy-options policy-statement BGP_export term 0 from interface lo0.1
set logical-systems CE2 policy-options policy-statement BGP_export term 0 then accept
set logical-systems CE2 routing-options rib inet.0 static route 0.0.0.0/0 next-hop 2.22.1.2
set logical-systems CE2 routing-options router-id 22.22.22.22
set logical-systems CE2 routing-options autonomous-system 100

```

PE3:

```

set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 7.7.7.7
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet-vpn unicast extended-nexthop
set protocols bgp group ibgp family inet6-vpn unicast
set protocols bgp group ibgp neighbor 1.1.1.1
set protocols bgp group ibgp neighbor 2.2.2.2
set protocols bgp group ibgp neighbor 8.8.8.8
set protocols bgp multipath
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable

set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 7.7.7.7
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then accept
set policy-options policy-statement L3VPN_PE3_CE3_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE3_CE3_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options community L3VPN_1_COM members target:64512:1

set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options autonomous-system 64512
set routing-options forwarding-table export LBPP

set logical-systems CE3 protocols bgp group eBGPv4 type external
set logical-systems CE3 protocols bgp group eBGPv4 export BGP_export
set logical-systems CE3 protocols bgp group eBGPv4 peer-as 64512
set logical-systems CE3 protocols bgp group eBGPv4 neighbor 7.77.1.2
set logical-systems CE3 policy-options policy-statement BGP_export term 0 from protocol direct

```

```
set logical-systems CE3 policy-options policy-statement BGP_export term 0 from interface lo0.1
set logical-systems CE3 policy-options policy-statement BGP_export term 0 then accept
set logical-systems CE3 routing-options rib inet.0 static route 0.0.0.0/0 next-hop 7.77.1.2
set logical-systems CE3 routing-options router-id 77.77.77.77
set logical-systems CE3 routing-options autonomous-system 100
```

PE4:

```
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 8.8.8.8
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet-vpn unicast extended-nexthop
set protocols bgp group ibgp family inet6-vpn unicast
set protocols bgp group ibgp neighbor 1.1.1.1
set protocols bgp group ibgp neighbor 2.2.2.2
set protocols bgp group ibgp neighbor 7.7.7.7
set protocols bgp multipath
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable

set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then next-hop 8.8.8.8
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then accept
set policy-options policy-statement L3VPN_PE4_CE4_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_PE4_CE4_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options policy-statement mpath-resolve then multipath-resolve
set policy-options community L3VPN_1_COM members target:64512:1

set routing-options resolution preserve-nexthop-hierarchy
set routing-options resolution rib inet6.0 import mpath-resolve
set routing-options resolution rib inet.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn-inet6.0 import mpath-resolve
set routing-options resolution rib bgp.l3vpn.0 import mpath-resolve
set routing-options router-id 8.8.8.8
set routing-options autonomous-system 64512
set routing-options forwarding-table export LBPP

set logical-systems CE4 interfaces ge-0/0/2 unit 0 description ConnectedToPE4
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family inet address 8.88.1.1/24
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family iso
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family mpls
set logical-systems CE4 interfaces lo0 unit 1 family inet address 88.88.88.88/32
set logical-systems CE4 interfaces lo0 unit 1 family iso address 47.0005.0019.2168.8800.00
set logical-systems CE4 protocols bgp group eBGPv4 type external
set logical-systems CE4 protocols bgp group eBGPv4 export BGP_export
set logical-systems CE4 protocols bgp group eBGPv4 peer-as 64512
```

```
set logical-systems CE4 protocols bgp group eBGPv4 neighbor 8.88.1.2
set logical-systems CE4 policy-options policy-statement BGP_export term 0 from protocol direct
set logical-systems CE4 policy-options policy-statement BGP_export term 0 from interface lo0.1
set logical-systems CE4 policy-options policy-statement BGP_export term 0 then accept
set logical-systems CE4 routing-options rib inet.0 static route 0.0.0/0 next-hop 8.88.1.2
set logical-systems CE4 routing-options router-id 88.88.88.88
set logical-systems CE4 routing-options autonomous-system 100
```

P1:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 10
set protocols isis interface ge-0/0/0.0 point-to-point
set protocols isis interface ge-0/0/1.0 level 2 metric 1
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 metric 100
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface fxp0.0 disable
```

P2:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 10
set protocols isis interface ge-0/0/0.0 point-to-point
set protocols isis interface ge-0/0/1.0 level 2 metric 100
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 metric 1
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/3.0
```

```
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface fxp0.0 disable
```

P3:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 1
set protocols isis interface ge-0/0/0.0 point-to-point
set protocols isis interface ge-0/0/1.0 level 2 metric 100
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 metric 10
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable
```

P4:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 100
set protocols isis interface ge-0/0/0.0 point-to-point
set protocols isis interface ge-0/0/1.0 level 2 metric 1
set protocols isis interface ge-0/0/1.0 point-to-point
set protocols isis interface ge-0/0/2.0 level 2 metric 10
set protocols isis interface ge-0/0/2.0 point-to-point
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/3.0 point-to-point
set protocols isis interface ge-0/0/4.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 point-to-point
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols isis level 1 disable
set protocols ldp interface ge-0/0/0.0
set protocols ldp interface ge-0/0/1.0
set protocols ldp interface ge-0/0/2.0
set protocols ldp interface ge-0/0/3.0
set protocols ldp interface ge-0/0/4.0
set protocols ldp interface fxp0.0 disable
set protocols mpls interface ge-0/0/0.0
set protocols mpls interface ge-0/0/1.0
set protocols mpls interface ge-0/0/2.0
```

```
set protocols mpls interface ge-0/0/3.0
set protocols mpls interface ge-0/0/4.0
set protocols mpls interface fxp0.0 disable
```

### 1.3 Enabling IPv6 Addresses on CEs:

PE1:

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:11::2/48
set logical-systems CE1 interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:11::1/48
set logical-systems CE1 interfaces lo0 unit 1 family inet6 address 2001:db8:11:255:11::11/128
set logical-systems CE1 routing-options ipv6-router-id 2001:db8:11:255:11::11
set logical-systems CE1 routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:11::1
set protocols mpls ipv6-tunneling
set logical-systems CE1 policy-options policy-statement BGP_export term 0 from protocol static
set logical-systems CE1 policy-options policy-statement BGP_export term 0 then reject
set logical-systems CE1 policy-options policy-statement BGP_export term 1 from family inet6
set logical-systems CE1 policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:11::1
set logical-systems CE1 policy-options policy-statement BGP_export term 1 then accept
set logical-systems CE1 policy-options policy-statement BGP_export term 2 from protocol direct
set logical-systems CE1 policy-options policy-statement BGP_export term 2 then accept
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv4 family inet unicast
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv4 family inet6 unicast
set logical-systems CE1 protocols bgp group eBGPv4 family inet unicast
set logical-systems CE1 protocols bgp group eBGPv4 family inet6 unicast
```

PE2:

```
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:22::2/48
set logical-systems CE2 interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:22::1/48
set logical-systems CE2 interfaces lo0 unit 1 family inet6 address 2001:db8:22:255:22::22/128
set logical-systems CE2 routing-options ipv6-router-id 2001:db8:22:255:22::22
set logical-systems CE2 routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:22::1
set protocols mpls ipv6-tunneling
set logical-systems CE2 policy-options policy-statement BGP_export term 0 from protocol static
set logical-systems CE2 policy-options policy-statement BGP_export term 0 then reject
set logical-systems CE2 policy-options policy-statement BGP_export term 1 from family inet6
set logical-systems CE2 policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:22::1
set logical-systems CE2 policy-options policy-statement BGP_export term 1 then accept
set logical-systems CE2 policy-options policy-statement BGP_export term 2 from protocol direct
set logical-systems CE2 policy-options policy-statement BGP_export term 2 then accept
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv4 family inet unicast
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv4 family inet6 unicast
set logical-systems CE2 protocols bgp group eBGPv4 family inet unicast
set logical-systems CE2 protocols bgp group eBGPv4 family inet6 unicast
```

PE3:

```
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:77::2/48
set logical-systems CE3 interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:77::1/48
set logical-systems CE3 interfaces lo0 unit 1 family inet6 address 2001:db8:77:255:77::77/128
set logical-systems CE3 routing-options ipv6-router-id 2001:db8:77:255:77::77
set logical-systems CE3 routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:77::1
set protocols mpls ipv6-tunneling
set logical-systems CE3 policy-options policy-statement BGP_export term 0 from protocol static
set logical-systems CE3 policy-options policy-statement BGP_export term 0 then reject
```

```

set logical-systems CE3 policy-options policy-statement BGP_export term 1 from family inet6
set logical-systems CE3 policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:77::1
set logical-systems CE3 policy-options policy-statement BGP_export term 1 then accept
set logical-systems CE3 policy-options policy-statement BGP_export term 2 from protocol direct
set logical-systems CE3 policy-options policy-statement BGP_export term 2 then accept
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv4 family inet unicast
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv4 family inet6 unicast
set logical-systems CE3 protocols bgp group eBGPv4 family inet unicast
set logical-systems CE3 protocols bgp group eBGPv4 family inet6 unicast

```

PE4:

```

set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:88::2/48
set logical-systems CE4 interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:88::1/48
set logical-systems CE4 interfaces lo0 unit 1 family inet6 address 2001:db8:88:255:88::88/128
set logical-systems CE4 routing-options ipv6-router-id 2001:db8:88:255:88::88
set logical-systems CE4 routing-options rib inet6.0 static route 0::0/0 next-hop 2001:db8:88::1
set protocols mpls ipv6-tunneling
set logical-systems CE4 policy-options policy-statement BGP_export term 0 from protocol static
set logical-systems CE4 policy-options policy-statement BGP_export term 0 then reject
set logical-systems CE4 policy-options policy-statement BGP_export term 1 from family inet6
set logical-systems CE4 policy-options policy-statement BGP_export term 1 then next-hop 2001:db8:88::1
set logical-systems CE4 policy-options policy-statement BGP_export term 1 then accept
set logical-systems CE4 policy-options policy-statement BGP_export term 2 from protocol direct
set logical-systems CE4 policy-options policy-statement BGP_export term 2 then accept
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv4 family inet unicast
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv4 family inet6 unicast
set logical-systems CE4 protocols bgp group eBGPv4 family inet unicast
set logical-systems CE4 protocols bgp group eBGPv4 family inet6 unicast

```

P1, P2, P3, P4:

```
set protocols mpls ipv6-tunneling
```

## 1.4 Enabling IPv6 in the legacy network:

PE1:

```

set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:13::1/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:14::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:1:255:1::1/128
set routing-options ipv6-router-id 2001:db8:1:255:1::1

```

PE2:

```

set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:23::1/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:24::1/24
set interfaces lo0 unit 0 family inet6 address 2001:db8:2:255:2::2/128
set routing-options ipv6-router-id 2001:db8:2:255:2::2

```

PE3:

```

set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:57::2/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:67::2/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:7:255:7::7/128
set routing-options ipv6-router-id 2001:db8:7:255:7::7

```

PE4:

```
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:68::2/64
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:58::2/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:8:255:8::8/128
set routing-options ipv6-router-id 2001:db8:8:255:8::8
```

P1:

```
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:13::2/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:23::2/64
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:34::1/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:35::1/64
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:36::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:3:255:3::3/128
set routing-options ipv6-router-id 2001:db8:3:255:3::3
```

P2:

```
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:14::2/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:24::2/64
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:34::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:45::1/64
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:46::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:4:255:4::4/128
set routing-options ipv6-router-id 2001:db8:4:255:4::4
```

P3:

```
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:35::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:45::2/64
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:56::1/64
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:57::1/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:58::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:5:255:5::5/128
set routing-options ipv6-router-id 2001:db8:5:255:5::5
```

P4:

```
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:db8:36::2/64
set interfaces ge-0/0/1 unit 0 family inet6 address 2001:db8:46::2/64
set interfaces ge-0/0/2 unit 0 family inet6 address 2001:db8:56::2/64
set interfaces ge-0/0/3 unit 0 family inet6 address 2001:db8:67::1/64
set interfaces ge-0/0/4 unit 0 family inet6 address 2001:db8:68::1/64
set interfaces lo0 unit 0 family inet6 address 2001:db8:6:255:6::6/128
set routing-options ipv6-router-id 2001:db8:6:255:6::6
```

## Appendix 2: Configuring SRv6 Transport

### Section 2.1.2 configuring locators

PE1:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:1::/64
PE2:
```

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:2::/64
Pl:
```

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a2:1::/64
```

P2:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a2:2::/64
```

P3:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a2:3::/64
```

P4:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a2:4::/64
```

PE3:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:1::/64
PE4:
```

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:2::/64
```

### Section 2.2 Configuring End SID and End.X SID

PE1:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:1:12:0::0 flavor psp
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:213:0:0:0 flavor usd psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:214:0:0:0 flavor usd psp
```

PE2:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:2:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a1:2:12:0::0 flavor psp
```

```
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:2:223:0:0:0 flavor usd psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:2:224:0:0:0 flavor usd psp
```

P1:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:1:11::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:1:12::0 flavor psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:231::0:0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:232::0:0 flavor usd psp
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:234::0:0 flavor usd psp
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:235::0:0 flavor usd psp
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:236::0:0 flavor usd psp
```

P2:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:2:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:2:12:0::0 flavor psp

set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:241:0::0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:242:0::0 flavor usd psp
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:243:0::0 flavor usd psp
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:245:0::0 flavor usd psp
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:246:0::0 flavor usd psp
```

P3:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:3:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:3:12:0::0 flavor psp
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:253:0::0 flavor usd psp
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:254:0::0 flavor usd psp
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:256:0::0 flavor usd psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:257:0::0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:258:0::0 flavor usd psp
```

P4.

```
set protocols isis source-packet-routing srv6 locator loc1 end-x-sid 2001:db8:a2:4:11::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-x-sid 2001:db8:a2:4:12::0 flavor psp
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:263::0:0 flavor usd psp
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:264::0:0 flavor usd psp
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:265::0:0 flavor usd psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:267::0:0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:268::0:0 flavor usd psp
```

PE3:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:1:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:1:12:0::0 flavor psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:1:275:0:0:0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:1:276:0:0:0 flavor usd psp
commit
```

PE4:

```
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:2:11:0::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a3:2:12:0::0 flavor psp
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:2:285:0:0:0 flavor usd psp
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:2:286:0:0:0 flavor usd psp
```

## Appendix 3: Configuring SRv6 Services

### 3.1 Enabling SRv6 END.DT4 and END.DT6 under BGP:

PE1:

```
set routing-instances L3VPN_PE1_CE1 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a1:1:3:1:11:0
set routing-instances L3VPN_PE1_CE1 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a1:1:4:1:11:0
```

PE2:

```
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a1:2:3:2:22:0
set routing-instances L3VPN_PE2_CE2 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a1:2:4:2:22:0
```

PE3:

```
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:1:3:7:77:0
set routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:1:4:7:77:0
```

PE4:

```
set routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a3:2:3:8:88:0
set routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a3:2:4:8:88:0
```

### 3.2 Enabling iBGP peers to accept and advertise SRv6 Services:

PE1:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet6 unicast extended-nexthop-color
```

PE2:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet6 unicast extended-nexthop-color
```

PE3:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet6 unicast extended-nexthop-color
```

PE4:

```
set protocols bgp group ibgp family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgp family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgp family inet unicast extended-nexthop
set protocols bgp group ibgp family inet6 unicast extended-nexthop-color
```

From JUNOS 22.3 onwards no further config changes are required to manipulate the Next-Hop of the services in Egress PEs.

## Appendix 4: Enabling TI-LFA Protection

### Section 4.3 TILFA at headend

#Adjusting the topology

P4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 100
```

```
set protocols isis interface ge-0/0/4.0 level 2 metric 100
```

P2:

```
set protocols isis interface ge-0/0/4.0 level 2 metric 100
```

PE3:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 100
```

PE4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 100
```

PE2:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 100
```

#Enabling TILFA

PE1:

```
set protocols isis backup-spf-options use-post-convergence-lfa
```

```
set protocols isis interface ge-0/0/2.0 level 2 post-convergence-lfa
```

```
set protocols isis backup-spf-options use-source-packet-routing
```

#Resetting

P4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
set protocols isis interface ge-0/0/4.0 level 2 metric 1
```

P2:

```
set protocols isis interface ge-0/0/4.0 level 2 metric 1
```

PE3:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

PE4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

PE2:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

#### 4.4 TILFA at transit

#Enable TILFA at transit P1:

```
set protocols isis backup-spf-options use-post-convergence-lfa
```

```
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
```

```
set protocols isis backup-spf-options use-source-packet-routing
```

#Enable insert mode

P1:

```
set routing-options source-packet-routing srv6 transit-srh-insert
```

#### 4.5 Anycast SID

```
#Enable Anycast
```

P4:

```
set routing-options source-packet-routing srv6] locator loc2 2001:db8:aa:0::/64
;
set routing-options source-packet-routing srv6 locator loc2 anycast
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:aa:0:1::0 flavor psp
deactivate protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:4:12:0::0
#resetting
P4:
deactivate routing-options source-packet-routing srv6] locator loc2
deactivate protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:aa:0:1::0
activate protocols isis source-packet-routing srv6 locator loc1 end-sid 2001:db8:a2:4:12:0::0
```

## Appendix 5: SRv6 to MPLS Service Interworking

### Section 5.1 Enabling IPv6 EBGP sessions to CEs

PE1:

```
#configure Routing instance and v6 eBGP connection to CE1
```

```
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 neighbor 2001:db8:11::1
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 1.11.1.2
set routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv6 family inet6 unicast
set logical-systems CE1 protocols bgp group eBGPv6 type external
set logical-systems CE1 protocols bgp group eBGPv6 export BGP_export
set logical-systems CE1 protocols bgp group eBGPv6 peer-as 64512
set logical-systems CE1 protocols bgp group eBGPv6 neighbor 2001:db8:11::2
set logical-systems CE1 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 1.11.1.1
set logical-systems CE1 protocols bgp group eBGPv6 family inet6 unicast
```

PE2:

```
#configure Routing instance and v6 eBGP connection to CE2
```

```
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 neighbor 2001:db8:22::1
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 2.22.1.2
set routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv6 family inet6 unicast
```

```

set logical-systems CE2 protocols bgp group eBGPv6 type external
set logical-systems CE2 protocols bgp group eBGPv6 export BGP_export
set logical-systems CE2 protocols bgp group eBGPv6 peer-as 64512
set logical-systems CE2 protocols bgp group eBGPv6 neighbor 2001:db8:22::2
set logical-systems CE2 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 2.22.1.1
set logical-systems CE2 protocols bgp group eBGPv6 family inet6 unicast

```

PE3:

#configure Routing instance and v6 eBGP connection to CE3

```

set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv6 neighbor 2001:db8:77::1
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 7.77.1.2
set routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv6 family inet6 unicast

set logical-systems CE3 protocols bgp group eBGPv6 type external
set logical-systems CE3 protocols bgp group eBGPv6 export BGP_export
set logical-systems CE3 protocols bgp group eBGPv6 peer-as 64512
set logical-systems CE3 protocols bgp group eBGPv6 neighbor 2001:db8:77::2
set logical-systems CE3 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 7.77.1.1
set logical-systems CE3 protocols bgp group eBGPv6 family inet6 unicast

```

PE4:

#configure Routing instance and v6 eBGP connection to CE4

```

set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv6 type external
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv6 peer-as 100
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv6 neighbor 2001:db8:88::1
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 8.88.1.2
set routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv6 family inet6 unicast
set logical-systems CE4 protocols bgp group eBGPv6 type external
set logical-systems CE4 protocols bgp group eBGPv6 export BGP_export
set logical-systems CE4 protocols bgp group eBGPv6 peer-as 64512
set logical-systems CE4 protocols bgp group eBGPv6 neighbor 2001:db8:88::2
set logical-systems CE4 protocols bgp group eBGPv6 family inet unicast local-ipv4-address 8.88.1.1
set logical-systems CE4 protocols bgp group eBGPv6 family inet6 unicast

```

## Section 5.1 Enabling IPv6 IBGP sessions among PEs

PE1:

#iBGP peering from PE2 to PE1, PE3 and PE4

```

set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6-vpn unicast
set protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
set protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8

```

PE2:

#iBGP peering from PE3 to PE1, PE2 and PE4

```
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6-vpn unicast
set protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
set protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
```

PE3:

#iBGP peering from PE1 to PE2, PE3 and PE4

```
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:7:255:7::7
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6-vpn unicast
set protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
commit
```

PE4:

#iBGP peering from PE4 to PE1, PE2 and PE3

```
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 local-address 2001:db8:8:255:8::8
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet-vpn unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6-vpn unicast
set protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
set protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
```

## Section 5.1 Advertise SRv6 services on IPv6 sessions

PE1:

```
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
```

PE2:

```
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
```

PE3:

```
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
```

PE4:

```
set protocols bgp group ibgpv6 family inet-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
set protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
set protocols bgp group ibgpv6 family inet unicast extended-nexthop
set protocols bgp group ibgpv6 family inet6 unicast extended-nexthop-color
```

## Section 5.2.1 to Section 5.2.6 Configuring SRv6 to MPLS service interworking

PE1:

```
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
deactivate protocols bgp group ibgp
set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 next-hop 2001:db8:a2:3::
set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 resolve
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 next-hop 2001:db8:a2:4::
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 resolve
```

PE2:

```
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
deactivate protocols bgp group ibgp
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 next-hop 2001:db8:a2:3::
set routing-options rib inet6.3 static route 2001:db8:5:255:5::5/128 resolve
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 next-hop 2001:db8:a2:4::
set routing-options rib inet6.3 static route 2001:db8:6:255:6::6/128 resolve
```

P3:

```
set routing-options autonomous-system 64512
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers family inet6 unicast
set protocols bgp group internal-peers family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers cluster 2.2.2.2
set protocols bgp group internal-peers peer-as 64512
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 local-address 2001:db8:5:255:5::5
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 cluster 2.2.2.2
set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 local-address 2001:db8:5:255:5::5
```

```

set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 cluster 2.2.2.2
set protocols bgp group internal-peers export exportstatic1
delete protocols bgp group internal-peers family inet-vpn
set protocols bgp group internal-peers family inet-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet-vpn unicast accept-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast accept-srv6-service
set protocols bgp group internal-peers-mpls type internal
set protocols bgp group internal-peers-mpls family inet-vpn unicast
set protocols bgp group internal-peers-mpls family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers-mpls family inet6-vpn unicast
set protocols bgp group internal-peers-mpls cluster 2.2.2.2
set protocols bgp group internal-peers-mpls peer-as 64512
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 local-address
2001:db8:5:255:5::5
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 local-address
2001:db8:5:255:5::5
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls export exportstatic2
set routing-instances L3VPN protocols bgp source-packet-routing srv6 locator loc1 end-dt6-sid
2001:db8:a2:3:31:0::0
set routing-instances L3VPN protocols bgp source-packet-routing srv6 locator loc1 end-dt4-sid
2001:db8:a2:3:41:0::0

#set protocols bgp export exportstatic1

delete policy-options policy-statement exportstatic1
set policy-options policy-statement exportstatic1 term 1 from protocol bgp
set policy-options policy-statement exportstatic1 term 1 from family inet-vpn
set policy-options policy-statement exportstatic1 term 1 then next-hop self
set policy-options policy-statement exportstatic1 term 1 then accept
set policy-options policy-statement exportstatic1 term exportstatic1 from family inet6-vpn
set policy-options policy-statement exportstatic1 term exportstatic1 from protocol bgp
set policy-options policy-statement exportstatic1 term exportstatic1 then next-hop 2001:db8:a2:3::1
set policy-options policy-statement exportstatic1 term exportstatic1 then accept

delete policy-options policy-statement exportstatic2 term exportstatic1 from family inet6
set policy-options policy-statement exportstatic2 term exportstatic1 from protocol bgp
set policy-options policy-statement exportstatic2 term exportstatic1 then next-hop 5.5.5.5
set policy-options policy-statement exportstatic2 term exportstatic1 then accept

#policy-options

set policy-options policy-statement L3VPN_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_export term 0 then accept
set policy-options policy-statement L3VPN_export term 0 then next-hop 2001:db8:a2:3::1
set policy-options policy-statement L3VPN_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options community L3VPN_1_COM members target:64512:1

set routing-instances L3VPN instance-type vrf
set routing-instances L3VPN route-distinguisher 5.5.5.5:5
set routing-instances L3VPN vrf-import L3VPN_import
set routing-instances L3VPN vrf-export L3VPN_export
set routing-instances L3VPN vrf-table-label;

```

P4:

```

set routing-options autonomous-system 64512
set protocols bgp group internal-peers type internal
set protocols bgp group internal-peers family inet6 unicast
set protocols bgp group internal-peers family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers cluster 2.2.2.2
set protocols bgp group internal-peers peer-as 64512
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 local-address 2001:db8:6:255:6::6
set protocols bgp group internal-peers neighbor 2001:db8:1:255:1::1 cluster 2.2.2.2
set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 local-address 2001:db8:6:255:6::6
set protocols bgp group internal-peers neighbor 2001:db8:2:255:2::2 cluster 2.2.2.2
set protocols bgp group internal-peers export exportstatic1
delete protocols bgp group internal-peers family inet-vpn
set protocols bgp group internal-peers family inet-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet-vpn unicast accept-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast advertise-srv6-service
set protocols bgp group internal-peers family inet6-vpn unicast accept-srv6-service
set protocols bgp group internal-peers-mpls type internal
set protocols bgp group internal-peers-mpls family inet-vpn unicast
set protocols bgp group internal-peers-mpls family inet-vpn unicast extended-nexthop
set protocols bgp group internal-peers-mpls family inet6-vpn unicast
set protocols bgp group internal-peers-mpls cluster 2.2.2.2
set protocols bgp group internal-peers-mpls peer-as 64512
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 local-address
2001:db8:5:255:6::6
set protocols bgp group internal-peers-mpls neighbor 2001:db8:7:255:7::7 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 local-address
2001:db8:5:255:6::6
set protocols bgp group internal-peers-mpls neighbor 2001:db8:8:255:8::8 cluster 2.2.2.2
set protocols bgp group internal-peers-mpls export exportstatic2

delete policy-options policy-statement exportstatic1
set policy-options policy-statement exportstatic1 term 1 from protocol bgp
set policy-options policy-statement exportstatic1 term 1 from family inet-vpn
set policy-options policy-statement exportstatic1 term 1 then next-hop self
set policy-options policy-statement exportstatic1 term 1 then accept
set policy-options policy-statement exportstatic1 term exportstatic1 from family inet6-vpn
set policy-options policy-statement exportstatic1 term exportstatic1 from protocol bgp
set policy-options policy-statement exportstatic1 term exportstatic1 then next-hop 2001:db8:a2:4:::
set policy-options policy-statement exportstatic1 term exportstatic1 then accept

set policy-options policy-statement exportstatic2 term exportstatic1 from protocol bgp
set policy-options policy-statement exportstatic2 term exportstatic1 then next-hop 6.6.6.6
set policy-options policy-statement exportstatic2 term exportstatic1 then accept

#policy-options

set policy-options policy-statement L3VPN_export term 0 then community add L3VPN_1_COM
set policy-options policy-statement L3VPN_export term 0 then accept
set policy-options policy-statement L3VPN_export term 0 then next-hop 2001:db8:a2:4:::
set policy-options policy-statement L3VPN_import term 0 from community L3VPN_1_COM
set policy-options policy-statement L3VPN_import term 0 then accept
set policy-options policy-statement LBPP term 1 then load-balance per-packet
set policy-options community L3VPN_1_COM members target:64512:1

```

```
set routing-instances L3VPN instance-type vrf
set routing-instances L3VPN route-distinguisher 6.6.6.6:6
set routing-instances L3VPN vrf-import L3VPN_import
set routing-instances L3VPN vrf-export L3VPN_export
set routing-instances L3VPN vrf-table-label;
```

PE3:

```
deactivate routing-options source-packet-routing srv6
deactivate protocols bgp source-packet-routing srv6
deactivate routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6
deactivate protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
deactivate protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
deactivate protocols isis source-packet-routing srv6

deactivate protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
deactivate protocols bgp group ibgp

set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 7.7.7.7
deactivate routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6
```

PE4:

```
deactivate routing-options source-packet-routing srv6
deactivate protocols bgp source-packet-routing srv6
deactivate routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6
deactivate protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
deactivate protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
deactivate protocols isis source-packet-routing srv6
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
deactivate protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2
set protocols bgp group ibgpv6 type internal
set protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
set protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
deactivate protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
deactivate protocols bgp group ibgp
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then next-hop 8.8.8.8
deactivate routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6
```

### Section 5.2.7 Restoring back SRv6 end-to-end

PE1:

```
activate protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
activate protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
delete protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
delete protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
activate protocols bgp group ibgp
delete routing-options rib inet6.3
```

PE2:

```
activate protocols bgp group ibgpv6 neighbor 2001:db8:7:255:7::7
activate protocols bgp group ibgpv6 neighbor 2001:db8:8:255:8::8
activate protocols bgp group ibgp

delete protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
delete protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
delete routing-options rib inet6.3
```

P3:

```
delete routing-options autonomous-system 64512
delete protocols bgp
delete routing-instances L3VPN
delete policy-options policy-statement exportstatic1
delete policy-options policy-statement exportstatic2
#policy-options
delete policy-options policy-statement L3VPN_export
delete policy-options policy-statement L3VPN_import
delete policy-options community L3VPN_1_COM members target:64512:1
```

P4:

```
delete routing-options autonomous-system 64512
delete protocols bgp
delete routing-instances L3VPN
delete policy-options policy-statement exportstatic1
delete policy-options policy-statement exportstatic2
#policy-options
delete policy-options policy-statement L3VPN_export
delete policy-options policy-statement L3VPN_import
delete policy-options community L3VPN_1_COM members target:64512:1
```

PE3:

```
activate routing-options source-packet-routing srv6
activate protocols bgp source-packet-routing srv6
activate routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6
activate protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
activate protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
activate protocols isis source-packet-routing srv6
activate protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
activate protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2

delete protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
delete protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
activate protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
activate protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
activate protocols bgp group ibgp
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then next-hop 2001:db8:a3:1::
activate routing-instances L3VPN_PE3_CE3 protocols bgp source-packet-routing srv6
```

PE4:

```
activate routing-options source-packet-routing srv6
activate protocols bgp source-packet-routing srv6
activate routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6
```

```

activate protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
activate protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
activate protocols isis source-packet-routing srv6
activate protocols bgp group ibgpv6 neighbor 2001:db8:1:255:1::1
activate protocols bgp group ibgpv6 neighbor 2001:db8:2:255:2::2

delete protocols bgp group ibgpv6 neighbor 2001:db8:5:255:5::5
delete protocols bgp group ibgpv6 neighbor 2001:db8:6:255:6::6
activate protocols bgp group ibgpv6 family inet6-vpn unicast accept-srv6-service
activate protocols bgp group ibgpv6 family inet6-vpn unicast advertise-srv6-service
activate protocols bgp group ibgp
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then next-hop 2001:db8:a3:2::1
activate routing-instances L3VPN_PE4_CE4 protocols bgp source-packet-routing srv6
commit

```

## Appendix 6: BGP-LS support

### Section 6.1.3 to 6.1.5 Configuring BGP-LS

PE1:

```

set protocols isis traffic-engineering advertisement always
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering ipv6
set protocols bgp group ibgp family traffic-engineering unicast

```

PE2:

```
set protocols isis traffic-engineering advertisement always
```

P1:

```
set protocols isis traffic-engineering advertisement always
```

P2:

```
set protocols isis traffic-engineering advertisement always
```

P3:

```
set protocols isis traffic-engineering advertisement always
```

P4:

```
set protocols isis traffic-engineering advertisement always
```

```
commit
```

PE3:

```

set protocols isis traffic-engineering advertisement always
set protocols isis traffic-engineering l3-unicast-topology
set protocols isis traffic-engineering ipv6

```

```
set protocols bgp group ibgp family traffic-engineering unicast
```

```

set protocols bgp group ibgp export nlri2bgp
set policy-options policy-statement ted_import then accept
set protocols mpls traffic-engineering database import policy ted_import
set protocols mpls traffic-engineering database import l3-unicast-topology bgp-link-state

set policy-options policy-statement nlri2bgp term 1 from family traffic-engineering
set policy-options policy-statement nlri2bgp term 1 then next-hop self
set policy-options policy-statement nlri2bgp term 1 then accept

```

PE4:

```
set protocols isis traffic-engineering advertisement always
```

## Appendix 7: Configuring SRv6 Flex Algo

PE1:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b1:1::/64 algorithm 128
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:1:2:1:3:0
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:1:2:1:4:0
set protocols isis source-packet-routing srv6 locator loc2
set protocols isis source-packet-routing flex-algorithm 128
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b1:1:1:1::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b1:1:1:2::0 flavor psp
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class

```

The remote flex algo usd flavored sids and 128 locator routes are present in the 128 color transport class table.

```

junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 10:17:25, metric 1
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:1::
2001:db8:b2:1:1::/128
 *[SRV6-ISIS/14] 10:17:25, metric 1
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:1:1::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 10:17:25, metric 1
 > to fe80::5604:dff:fe00:6d7 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:-
2001:db8:b2:2:1::/128
 *[SRV6-ISIS/14] 10:17:25, metric 1
 > to fe80::5604:dff:fe00:6d7 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:1:-
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 10:17:25, metric 4
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:-
2001:db8:b2:3:1::/128
 *[SRV6-ISIS/14] 10:17:25, metric 4
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:1:-
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 10:17:25, metric 2

```

```

 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:::
2001:db8:b2:4:1:1::/128
 *[SRV6-ISIS/14] 10:17:25, metric 2
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:1:::
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 10:17:25, metric 3
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:::
2001:db8:b3:1:1::/128
 *[SRV6-ISIS/14] 10:17:25, metric 3
 > to fe80::5604:dff:fe00:9de4 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:1:::

```

P1:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b2:1::/64
set protocols isis source-packet-routing flex-algorithm 128
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:2:3:4:0
set protocols isis interface ge-0/0/0.0 delay-metric 10
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:2:3:5:0
set protocols isis interface ge-0/0/1.0 delay-metric 100
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:2:3:6:0
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:2:3:1:0
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:2:3:2:0
set protocols isis interface ge-0/0/4.0 delay-metric 1
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class

```

```

junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:32, metric 1
 > to fe80::5604:dff:fe00:966a via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:::
2001:db8:b1:1:1::/128
 *[SRV6-ISIS/14] 00:00:32, metric 1
 > to fe80::5604:dff:fe00:966a via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:1:::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 00:00:32, metric 2
 > to fe80::5604:dff:fe00:966a via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:::
2001:db8:b2:2:1::/128
 *[SRV6-ISIS/14] 00:00:32, metric 2
 > to fe80::5604:dff:fe00:966a via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:1:::
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 00:00:32, metric 3
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:::
2001:db8:b2:3:1::/128
 *[SRV6-ISIS/14] 00:00:32, metric 3
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:1:::
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 00:00:32, metric 1
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:::
2001:db8:b2:4:1::/128
 *[SRV6-ISIS/14] 00:00:32, metric 1
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:1:::
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:00:32, metric 2
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:::

```

```

2001:db8:b3:1:1:1::/128
 *[SRV6-ISIS/14] 00:00:32, metric 2
 > to fe80::5604:dff:fe00:9296 via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:1:1::

P2:

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b2:2::/64 algorithm 128
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:2:4:3:0
set protocols isis interface ge-0/0/0.0 delay-metric 10
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:2:4:5:0
set protocols isis interface ge-0/0/1.0 delay-metric 10
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:2:4:6:0
set protocols isis interface ge-0/0/2.0 delay-metric 100
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:2:4:1:0
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:2:4:2:0
set protocols isis interface ge-0/0/4.0 delay-metric 1
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:2:1:1::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:2:1:2::0 flavor psp

set protocols isis source-packet-routing flex-algorithm 128
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class

junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:38, metric 1
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1::1
2001:db8:b1:1:1::/128
 *[SRV6-ISIS/14] 00:00:38, metric 1
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:1:1::1
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 00:00:38, metric 2
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1::1
2001:db8:b2:1:1::/128
 *[SRV6-ISIS/14] 00:00:38, metric 2
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:1:1::1
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 00:00:38, metric 5
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:3::1
 2001:db8:b2:3:1:1::/128
 *[SRV6-ISIS/14] 00:00:38, metric 5
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:1:1::1
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 00:00:38, metric 3
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4::1
 2001:db8:b2:4:1:1::/128
 *[SRV6-ISIS/14] 00:00:38, metric 3
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:1:1::1
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:00:38, metric 4
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1::1
 2001:db8:b3:1:1:1::/128
 *[SRV6-ISIS/14] 00:00:38, metric 4
 > to fe80::5604:dff:fe00:3751 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:1:1::1

```

P3:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b2:3::/64 algorithm 128
set protocols isis source-packet-routing flex-algorithm 128
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:2:5:3:0
set protocols isis interface ge-0/0/0.0 delay-metric 100
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:2:5:4:0
set protocols isis interface ge-0/0/1.0 delay-metric 10
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:2:5:6:0
set protocols isis interface ge-0/0/2.0 delay-metric 10
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:2:5:7:0
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:2:5:8:0
set protocols isis interface ge-0/0/4.0 delay-metric 1
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:3:1:1::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:3:1:2::0 flavor psp

set routing-options flex-algorithm 128 definition metric-type delay
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class
junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:42, metric 4
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:::
2001:db8:b1:1:1::/128
 *[SRV6-ISIS/14] 00:00:42, metric 4
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:1:::
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 00:00:42, metric 3
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:::
2001:db8:b2:1:1::/128
 *[SRV6-ISIS/14] 00:00:42, metric 3
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:1:::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 00:00:42, metric 5
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:::
2001:db8:b2:2:1::/128
 *[SRV6-ISIS/14] 00:00:42, metric 5
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:1:::
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 00:00:42, metric 2
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:::
2001:db8:b2:4:1::/128
 *[SRV6-ISIS/14] 00:00:42, metric 2
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:1:::
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:00:42, metric 1
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:::
2001:db8:b3:1:1::/128
 *[SRV6-ISIS/14] 00:00:42, metric 1
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:1:::

```

P4:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b2:4::/64 algorithm 128
set protocols isis source-packet-routing flex-algorithm 128
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:2:6:3:0
set protocols isis interface ge-0/0/0.0 delay-metric 1
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:2:6:4:0
set protocols isis interface ge-0/0/1.0 delay-metric 100
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:2:6:5:0
set protocols isis interface ge-0/0/2.0 delay-metric 10
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:2:6:7:0
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:2:6:8:0
set protocols isis interface ge-0/0/4.0 delay-metric 1
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:4:1:1::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b2:4:1:2::0 flavor psp
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class
junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:42, metric 4
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:::
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:48, metric 2
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:::
2001:db8:b1:1::1:/128
 *[SRV6-ISIS/14] 00:00:48, metric 2
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:1:::
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 00:00:48, metric 1
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:::
2001:db8:b2:1::1:/128
 *[SRV6-ISIS/14] 00:00:48, metric 1
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:1:::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 00:00:48, metric 3
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:::
2001:db8:b2:2:1::/128
 *[SRV6-ISIS/14] 00:00:48, metric 3
 > to fe80::5604:dff:fe00:62cc via ge-0/0/0.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:1:::
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 00:00:48, metric 2
 > to fe80::5604:dff:fe00:7705 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:::
2001:db8:b2:3:1::1:/128
 *[SRV6-ISIS/14] 00:00:48, metric 2
 > to fe80::5604:dff:fe00:7705 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:1:::
2001:db8:b3:1::/64 *[SRV6-ISIS/14] 00:00:48, metric 1
 > to fe80::5604:dff:fe00:7705 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:::
2001:db8:b3:1:1::1:/128
 *[SRV6-ISIS/14] 00:00:48, metric 1
 > to fe80::5604:dff:fe00:7705 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b3:1:1:::

```

PE3:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b3:1::/64 algorithm 128
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:1:2:7:5:0
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:1:2:7:6:0
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis source-packet-routing flex-algorithm 128
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b3:1:1:1::0 flavor usd
set protocols isis source-packet-routing srv6 locator loc2 end-sid 2001:db8:b3:1:1:2::0 flavor psp
set routing-options flex-algorithm 128 definition metric-type delay
set routing-options route-distinguisher-id 10
set routing-options transport-class auto-create
set routing-options flex-algorithm 128 use-transport-class
junos-rti-tc-128.inet6.3: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:42, metric 4
 > to fe80::5604:dff:fe00:3ded via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1::
2001:db8:b1:1::/64 *[SRV6-ISIS/14] 00:00:57, metric 3
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1::
2001:db8:b1:1:1::/128
 *[SRV6-ISIS/14] 00:00:57, metric 3
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b1:1:1::
2001:db8:b2:1::/64 *[SRV6-ISIS/14] 00:00:57, metric 2
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1::
2001:db8:b2:1:1::/128
 *[SRV6-ISIS/14] 00:00:57, metric 2
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:1:1::
2001:db8:b2:2::/64 *[SRV6-ISIS/14] 00:00:57, metric 4
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2::
2001:db8:b2:2:1::/128
 *[SRV6-ISIS/14] 00:00:57, metric 4
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:2:1:1::
2001:db8:b2:3::/64 *[SRV6-ISIS/14] 00:00:57, metric 1
 > to fe80::5604:dff:fe00:966d via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:-
2001:db8:b2:3:1:1::/128
 *[SRV6-ISIS/14] 00:00:57, metric 1
 > to fe80::5604:dff:fe00:966d via ge-0/0/2.0, SRV6-Tunnel, Dest: 2001:db8:b2:3:1:1::
2001:db8:b2:4::/64 *[SRV6-ISIS/14] 00:00:57, metric 1
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:-
2001:db8:b2:4:1:1::/128
 *[SRV6-ISIS/14] 00:00:57, metric 1
 > to fe80::5604:dff:fe00:5c22 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:b2:4:1:1::

```

## Appendix 8: IPv6-only Networks

### 8.1 Removing LDP and MPLS from CORE:

PE1, PE2, PE3, PE4 P1, P2, P3, P4:

```
delete protocols ldp
delete protocols mpls
```

### 8.2 Removing IPv4 addresses from CORE:

PE1:

```
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
delete routing-instances L3VPN_PE1_CE1 protocols bgp group eBGPv4
delete logical-systems CE1 protocols bgp group eBGPv4
delete protocols bgp group ibgp
```

PE2:

```
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
delete routing-instances L3VPN_PE2_CE2 protocols bgp group eBGPv4
delete logical-systems CE2 protocols bgp group eBGPv4
delete protocols bgp group ibgp
```

PE3:

```
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
delete routing-instances L3VPN_PE3_CE3 protocols bgp group eBGPv4
delete logical-systems CE3 protocols bgp group eBGPv4
delete protocols bgp group ibgp
```

PE4:

```
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
delete routing-instances L3VPN_PE4_CE4 protocols bgp group eBGPv4
delete logical-systems CE4 protocols bgp group eBGPv4
delete protocols bgp group ibgp
```

P1:

```
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces ge-0/0/0 unit 0 family inet
delete interfaces ge-0/0/1 unit 0 family inet
delete interfaces ge-0/0/2 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
```

P2:

```
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces ge-0/0/0 unit 0 family inet
delete interfaces ge-0/0/1 unit 0 family inet
delete interfaces ge-0/0/2 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
```

P3:

```
delete interfaces ge-0/0/0 unit 0 family inet
delete interfaces ge-0/0/1 unit 0 family inet
delete interfaces ge-0/0/2 unit 0 family inet
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
```

P4:

```
delete interfaces ge-0/0/0 unit 0 family inet
delete interfaces ge-0/0/1 unit 0 family inet
delete interfaces ge-0/0/2 unit 0 family inet
delete interfaces ge-0/0/3 unit 0 family inet
delete interfaces ge-0/0/4 unit 0 family inet
delete interfaces lo0 unit 0 family inet
delete routing-options router-id
```

## Appendix 9: Micro-loop Avoidance

### Preparing network for M-loop avoidance

PE1:

```
set protocols isis spf-options delay 1000

set firewall family inet6 filter filter1 term 1 from destination-address 2001:db8:a3:1:11:0::0/128
set firewall family inet6 filter filter1 term 1 then count cnt
set firewall family inet6 filter filter1 term 1 then accept
set firewall family inet6 filter filter1 term 2 then accept
set interfaces ge-0/0/2.0 unit 0 family inet6 filter output filter1
```

PE2:

```
set protocols isis interface ge-0/0/2.0 level 2 metric 2
```

P1:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 2
```

P4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 5
commit
```

PE3:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 5
```

### Enabling M-loop avoidance

PE1:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

PE2:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

P1:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

P2:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

P3:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

P4:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

PE3:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

PE4:

```
set protocols isis spf-options microloop-avoidance post-convergence-path delay 10000
```

Restoring original configs

PE2:

```
set protocols isis interface ge-0/0/2.0 level 2 metric 1
```

P2:

```
set protocols isis interface ge-0/0/0.0 level 2 metric 1
```

P4:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

PE3:

```
set protocols isis interface ge-0/0/3.0 level 2 metric 1
```

## Appendix 10: Configuring SRv6-TE

In this book we will take PE1 as ingress router and rest of the PEs as egress nodes for SRv6 TE for our simplicity. User can configure any PEs as ingress or egress depending on their network architecture. Thus in this section SRv6 TE configuration will be present in PE1 only.

### 10.1 Configuring SRv6 TE on PE1:

PE1:

```
#SRv6 tunnel towards PE2:
set protocols source-packet-routing segment-list to_PE2_via_P1 srv6
set protocols source-packet-routing segment-list to_PE2_via_P1 hop1 srv6-sid 2001:db8:a2:1:11::0
set protocols source-packet-routing segment-list to_PE2_via_P1 hop2 srv6-sid 2001:db8:a1:2:11::0::0
set protocols source-packet-routing segment-list to_PE2_via_P2 srv6
set protocols source-packet-routing segment-list to_PE2_via_P2 hop1 srv6-sid 2001:db8:a2:2:11::0::0
set protocols source-packet-routing segment-list to_PE2_via_P2 hop2 srv6-sid 2001:db8:a1:2:11::0::0
set protocols source-packet-routing srv6
set protocols source-packet-routing source-routing-path to_PE2 srv6
set protocols source-packet-routing source-routing-path to_PE2 to 2001:db8:2:255:2::2
set protocols source-packet-routing source-routing-path to_PE2 color 2
set protocols source-packet-routing source-routing-path to_PE2 primary to_PE2_via_P1
set protocols source-packet-routing source-routing-path to_PE2 primary to_PE2_via_P2

#SRv6 tunnel towards PE3:
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 srv6
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop1 srv6-sid 2001:db8:a2:1:11::0
```

```

set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE3_via_P1_P3 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 srv6
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P3 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 srv6
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop2 srv6-sid 2001:db8:a2:4:11::0
set protocols source-packet-routing segment-list to_PE3_via_P2_P4 hop3 srv6-sid 2001:db8:a3:1:11::0
set protocols source-packet-routing srv6
set protocols source-packet-routing source-routing-path to_PE3 srv6
set protocols source-packet-routing source-routing-path to_PE3 to 2001:db8:7:255:7::7
set protocols source-packet-routing source-routing-path to_PE3 color 3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P1_P3
set protocols source-packet-routing source-routing-path to_PE3 primary to_PE3_via_P2_P3
set protocols source-packet-routing source-routing-path to_PE3 secondary to_PE3_via_P2_P4

#SRv6 tunnel towards PE4:
set protocols source-packet-routing segment-list to_PE4_via_P1_P3 srv6
set protocols source-packet-routing segment-list to_PE4_via_P1_P3 hop1 srv6-sid 2001:db8:a2:1:11::0
set protocols source-packet-routing segment-list to_PE4_via_P1_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE4_via_P1_P3 hop3 srv6-sid 2001:db8:a3:2:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P4 srv6
set protocols source-packet-routing segment-list to_PE4_via_P2_P4 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P4 hop2 srv6-sid 2001:db8:a2:4:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P4 hop3 srv6-sid 2001:db8:a3:2:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P3 srv6
set protocols source-packet-routing segment-list to_PE4_via_P2_P3 hop1 srv6-sid 2001:db8:a2:2:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P3 hop2 srv6-sid 2001:db8:a2:3:11::0
set protocols source-packet-routing segment-list to_PE4_via_P2_P3 hop3 srv6-sid 2001:db8:a3:2:11::0
set protocols source-packet-routing srv6
set protocols source-packet-routing source-routing-path to_PE4 srv6
set protocols source-packet-routing source-routing-path to_PE4 to 2001:db8:8:255:8::8
set protocols source-packet-routing source-routing-path to_PE4 color 4
set protocols source-packet-routing source-routing-path to_PE4 primary to_PE4_via_P1_P3
set protocols source-packet-routing source-routing-path to_PE4 primary to_PE4_via_P2_P4
set protocols source-packet-routing source-routing-path to_PE4 secondary to_PE4_via_P2_P3
set routing-options forwarding-table srv6-chain-merge

```

With the above config our SRv6 TE Transport Tunnel routes are downloaded in inet-color6.0 with old resolution schemes:

```

[edit]
root@PE1# run show route protocol spring-te
inet6color.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:2:255:2::2-2<c6>/160
 *[SPRING-TE/8] 00:00:11, metric 1, metric2 2
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:2:255:2::2-
2<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:2:255:2::-
2<c6>
2001:db8:7:255:7::7-3<c6>/160
 *[SPRING-TE/8] 00:00:11, metric 1, metric2 3
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>

```

```

 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:7:255:7::7-3<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:7:255:7::7-3<c6>
2001:db8:8:255:8::8-4<c6>/160
 *[SPRING-TE/8] 00:00:11, metric 1, metric2 3
 to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
 > to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
```

## 10.2 Additional configurations for new resolution scheme starting from JUNOS 22.3:

```

set routing-options transport-class auto-create
set routing-options route-distinguisher-id 1.1.1.1
set protocols source-packet-routing use-transport-class
```

Now our SRv6 TE Transport tunnel routes will be present in color.tc RIBs:

```

junos-rti-tc-2.inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

2001:db8:2:255:2::2/128
 *[SPRING-TE/8] 00:00:03, metric 1, metric2 2
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest:
2001:db8:2:255:2::2-2<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:2:255:2::2-2<c6>
```

```

junos-rti-tc-3.inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

2001:db8:7:255:7::7/128
 *[SPRING-TE/8] 00:00:03, metric 1, metric2 3
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest:
2001:db8:7:255:7::7-3<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:7:255:7::7-3<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:7:255:7::7-3<c6>
```

```

junos-rti-tc-4.inet6.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```

2001:db8:8:255:8::8/128
 *[SPRING-TE/8] 00:00:03, metric 1, metric2 3
 to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
 > to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest:
2001:db8:8:255:8::8-4<c6>
```

## 10.3 BGP service mapping over SRv6 TE Colored Tunnels with new resolution scheme:

### 10.3.1 Configuring COLOR Community in Egress PE2, PE3 and PE4:

PE2:

```
set policy-options community L3VPN_1_COLOR members color:1:2
set policy-options policy-statement L3VPN_PE2_CE2_export term 0 then community add L3VPN_1_COLOR
```

PE3:

```
set policy-options community L3VPN_1_COLOR members color:1:3
set policy-options policy-statement L3VPN_PE3_CE3_export term 0 then community add L3VPN_1_COLOR
```

PE4:

```
set policy-options community L3VPN_1_COLOR members color:1:4
set policy-options policy-statement L3VPN_PE4_CE4_export term 0 then community add L3VPN_1_COLOR
```

### 10.3.2 Verifying one IPv4 and IPv6 service route from PE1 to each PE3:

#IPv4 Service via PE3 (show route towards IPv4 loopback address of CE3):  
L3VPN\_PE1\_CE1.inet.0: 9 destinations, 10 routes (9 active, 0 holddown, 0 hidden)  
+ = Active Route, - = Last Active, \* = Both

```
77.77.77.77/32 *[BGP/170] 00:00:11, localpref 100, from 2001:db8:7:255:7::7
 AS path: 100 I, validation-state: unverified
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

#IPv6 Service via PE3 (show route towards IPv6 loopback address of CE3):  
L3VPN\_PE1\_CE1.inet6.0: 11 destinations, 12 routes (11 active, 0 holddown, 0 hidden)  
+ = Active Route, - = Last Active, \* = Both

```
2001:db8:77:255:77::77/128
 *[BGP/170] 00:00:58, localpref 100, from 2001:db8:7:255:7::7
 AS path: 100 I, validation-state: unverified
 > to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:7:255:7::7-
3<c6>
 to fe80::5603:7ff:fe00:6b27 via ge-0/0/3.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
 to fe80::5603:7ff:fe00:37ca via ge-0/0/4.0, SRV6-Tunnel, Dest: 2001:db8:a3:1::
```

This verification can be done for other services as well.

## Appendix 11: Configuring SRv6 EVPN-VPWS

### Section 11.1 EVPN VPWS Single homed

PE1:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:1::/64
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 next-hop 2001:db8:a3:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 resolve
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 next-hop 2001:db8:a1:2:11:0::0
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 resolve
set routing-options resolution preserve-nexthop-hierarchy
set policy-options policy-statement vrf-export-evpnvpws term a then community add rt-evpnvpws
set policy-options policy-statement vrf-export-evpnvpws term a then accept
set policy-options policy-statement vrf-export-evpnvpws term b then reject
set policy-options policy-statement vrf-import-evpnvpws term a from protocol bgp
set policy-options policy-statement vrf-import-evpnvpws term a from community rt-evpnvpws
set policy-options policy-statement vrf-import-evpnvpws term a then accept
set policy-options policy-statement vrf-import-evpnvpws term b then reject
set policy-options community rt-evpnvpws members target:200:1
set protocols bgp group ibgpv6 family evpn signaling advertise-srv6-service
set protocols bgp group ibgpv6 family evpn signaling accept-srv6-service
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 encapsulation flexible-ethernet-services
set interfaces ge-0/0/1 unit 4 description evpn-vpws-ce-intf
set interfaces ge-0/0/1 unit 4 encapsulation vlan-ccc
set interfaces ge-0/0/1 unit 4 vlan-id 300
set routing-instances evpn-vpws instance-type evpn-vpws
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id local 102
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id remote 201
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing
srv6 locator loc1
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-packet-routing
srv6 locator end-dx2-sid 2001:db8:a1:1:101:0:0:0
set routing-instances evpn-vpws protocols evpn encapsulation srv6
set routing-instances evpn-vpws protocols evpn traceoptions file PE1_evpn.log
set routing-instances evpn-vpws protocols evpn traceoptions file size 100m
set routing-instances evpn-vpws protocols evpn traceoptions flag all
set routing-instances evpn-vpws interface ge-0/0/1.4
set routing-instances evpn-vpws route-distinguisher 1.1.1.1:20
set routing-instances evpn-vpws vrf-import vrf-import-evpnvpws
set routing-instances evpn-vpws vrf-export vrf-export-evpnvpws
```

PE3:

```
set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:1::/64
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 next-hop 2001:db8:a1:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 resolve
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 next-hop 2001:db8:a1:2:11:0::0
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 resolve
set routing-options resolution preserve-nexthop-hierarchy
set policy-options policy-statement vrf-export-evpnvpws term a then community add rt-evpnvpws
set policy-options policy-statement vrf-export-evpnvpws term a then accept
set policy-options policy-statement vrf-export-evpnvpws term b then reject
set policy-options policy-statement vrf-import-evpnvpws term a from protocol bgp
set policy-options policy-statement vrf-import-evpnvpws term a from community rt-evpnvpws
```

```

set policy-options policy-statement vrf-import-evpnvpws term a then accept
set policy-options policy-statement vrf-import-evpnvpws term b then reject
set policy-options community rt-evpnvpws members target:200:1
set protocols bgp group ibgpv6 family evpn signaling advertise-srv6-service
set protocols bgp group ibgpv6 family evpn signaling accept-srv6-service
set interfaces ge-0/0/1 description pe3->ce3-alpha
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 encapsulation flexible-ethernet-services
set interfaces ge-0/0/1 unit 4 description evpn-vpws-ce-intf
set interfaces ge-0/0/1 unit 4 encapsulation vlan-ccc
set interfaces ge-0/0/1 unit 4 vlan-id 300
set routing-instances evpn-vpws instance-type evpn-vpws
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id local 201
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id remote 102
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-
packet-routing srv6 locator loc1
set routing-instances evpn-vpws protocols evpn interface ge-0/0/1.4 vpws-service-id source-
packet-routing srv6 locator end-dx2-sid 2001:db8:a3:1:103:0:0:0
set routing-instances evpn-vpws protocols evpn encapsulation srv6
set routing-instances evpn-vpws protocols evpn traceoptions file pe3-evpn-vpws.log
set routing-instances evpn-vpws protocols evpn traceoptions file size 100m
set routing-instances evpn-vpws protocols evpn traceoptions flag all
set routing-instances evpn-vpws interface ge-0/0/1.4
set routing-instances evpn-vpws route-distinguisher 7.7.7.7:20
set routing-instances evpn-vpws vrf-import vrf-import-evpnvpws
set routing-instances evpn-vpws vrf-export vrf-export-evpnvpws

```

#### CE1 (as logical system in PE1):

```

set interfaces ge-0/0/0 vlan-tagging
set logical-systems CE1 interfaces ge-0/0/0 unit 1 description ce1-pe1-evpnvpws
set logical-systems CE1 interfaces ge-0/0/0 unit 1 vlan-id 300
set logical-systems CE1 interfaces ge-0/0/0 unit 1 family inet address 20.4.1.1/24

```

#### CE3 (as logical system in PE3):

```

set interfaces ge-0/0/0 vlan-tagging
set logical-systems CE3 interfaces ge-0/0/0 unit 1 description ce3-pe3-evpnvpws
set logical-systems CE3 interfaces ge-0/0/0 unit 1 vlan-id 300
set logical-systems CE3 interfaces ge-0/0/0 unit 1 family inet address 20.4.1.3/24

```

## Section 11.2 EVPN VPWS all-active/active-active

#### On PE1:

```

set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:1::/64
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 next-hop
2001:db8:a3:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 resolve
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 next-hop
2001:db8:a1:2:11:0::0
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 resolve
set routing-options resolution preserve-nexthop-hierarchy
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then community add rt-evpn-
vpws-mh
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-export-evpn-vpws-mh term b then reject

```

```

set policy-options policy-statement vrf-import-evpn-vpws-mh term a from protocol bgp
set policy-options policy-statement vrf-import-evpn-vpws-mh term a from community rt-evpn-vpws-mh
set policy-options policy-statement vrf-import-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-import-evpn-vpws-mh term b then reject
set policy-options community rt-evpn-vpws-mh members target:200:2
set interfaces ge-0/0/2 gigether-options 802.3ad ae0
set interfaces ae0 description pe1->ce12mh-alpha
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 esi 00:12:12:12:12:12:12:12
set interfaces ae0 esi single-active
set interfaces ae0 esi df-election-granularity per-esi lacp-oos-on-ndf
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:00:00:12
set interfaces ae0 unit 0 description evpn-vpws-mh-ce-intf
set interfaces ae0 unit 0 encapsulation vlan-ccc
set interfaces ae0 unit 0 vlan-id 30
set routing-instances evpn-vpws-mh instance-type evpn-vpws
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id local 103
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id remote 301
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id source-packet-routing
srv6 locator loc1
set routing-instances evpn-vpws-mh protocols evpn encapsulation srv6
set routing-instances evpn-vpws-mh protocols evpn traceoptions file PE1_evpn-vpws-mh.log
set routing-instances evpn-vpws-mh protocols evpn traceoptions file size 100m
set routing-instances evpn-vpws-mh protocols evpn traceoptions flag all
set routing-instances evpn-vpws-mh interface ae0.0
set routing-instances evpn-vpws-mh route-distinguisher 1.1.1.1:12
set routing-instances evpn-vpws-mh vrf-import vrf-import-evpn-vpws-mh
set routing-instances evpn-vpws-mh vrf-export vrf-export-evpn-vpws-mh

```

On PE2:

```

set routing-options source-packet-routing srv6 locator loc1 2001:db8:a1:2::/64
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 next-hop 2001:db8:a3:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:7:255:7::7/128 resolve
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 next-hop 2001:db8:a1:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 resolve
set routing-options resolution preserve-nexthop-hierarchy
set interfaces ge-0/0/0 gigether-options 802.3ad ae0
set interfaces ge-0/0/0 gigether-options 802.3ad ae0
set interfaces ae0 description pe2->ce12-mh
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 encapsulation flexible-ethernet-services
set interfaces ae0 esi 00:12:12:12:12:12:12:12
set interfaces ae0 esi single-active
set interfaces ae0 esi df-election-granularity per-esi lacp-oos-on-ndf
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:00:00:12
set interfaces ae0 unit 0 description evpn-vpws-mh-ce-intf
set interfaces ae0 unit 0 encapsulation vlan-ccc
set interfaces ae0 unit 0 vlan-id 30
set routing-instances evpn-vpws-mh instance-type evpn-vpws
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id local 103
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id remote 301
set routing-instances evpn-vpws-mh protocols evpn interface ae0.0 vpws-service-id source-packet-routing

```

```

srv6 locator loc1
set routing-instances evpn-vpws-mh protocols evpn encapsulation srv6
set routing-instances evpn-vpws-mh protocols evpn traceoptions file PE2_evpn-vpws-mh.log
set routing-instances evpn-vpws-mh protocols evpn traceoptions file size 100m
set routing-instances evpn-vpws-mh protocols evpn traceoptions flag all
set routing-instances evpn-vpws-mh interface ae0.0
set routing-instances evpn-vpws-mh route-distinguisher 2.2.2.2:12
set routing-instances evpn-vpws-mh vrf-import vrf-import-evpn-vpws-mh
set routing-instances evpn-vpws-mh vrf-export vrf-export-evpn-vpws-mh
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then community add rt-evpn-vpws-mh
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-export-evpn-vpws-mh term b then reject
set policy-options policy-statement vrf-import-evpn-vpws-mh term a from protocol bgp
set policy-options policy-statement vrf-import-evpn-vpws-mh term a from community rt-evpn-vpws-mh
set policy-options policy-statement vrf-import-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-import-evpn-vpws-mh term b then reject
set policy-options community rt-evpn-vpws-mh members target:200:2

```

### PE3:

```

set routing-options source-packet-routing srv6 locator loc1 2001:db8:a3:1::/64
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 next-hop 2001:db8:a1:1:11:0::0
set routing-options rib inet6.3 static route 2001:db8:1:255:1::1/128 resolve
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 next-hop 2001:db8:a1:2:11:0::0
set routing-options rib inet6.3 static route 2001:db8:2:255:2::2/128 resolve
set routing-options resolution preserve-nexthop-hierarchy
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then community add rt-evpn-vpws-mh
set policy-options policy-statement vrf-export-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-export-evpn-vpws-mh term b then reject
set policy-options policy-statement vrf-import-evpn-vpws-mh term a from protocol bgp
set policy-options policy-statement vrf-import-evpn-vpws-mh term a from community rt-evpn-vpws-mh
set policy-options policy-statement vrf-import-evpn-vpws-mh term a then accept
set policy-options policy-statement vrf-import-evpn-vpws-mh term b then reject
set policy-options community rt-evpn-vpws-mh members target:200:2
set interfaces ge-0/0/1 description pe3->ce3-alpha
set interfaces ge-0/0/1 vlan-tagging
set interfaces ge-0/0/1 encapsulation flexible-ethernet-services
set interfaces ge-0/0/1 unit 1 description evpn-vpws-ce-intf
set interfaces ge-0/0/1 unit 1 encapsulation vlan-ccc
set interfaces ge-0/0/1 unit 1 vlan-id 30

set routing-instances evpn-vpws-mh instance-type evpn-vpws
set routing-instances evpn-vpws-mh protocols evpn interface ge-0/0/1.1 vpws-service-id local 301
set routing-instances evpn-vpws-mh protocols evpn interface ge-0/0/1.1 vpws-service-id remote 103
set routing-instances evpn-vpws-mh protocols evpn interface ge-0/0/1.1 vpws-service-id source-packet-
routing srv6 locator loc1
set routing-instances evpn-vpws-mh protocols evpn encapsulation srv6
set routing-instances evpn-vpws-mh protocols evpn traceoptions file PE3_evpn-vpws-mh.log
set routing-instances evpn-vpws-mh protocols evpn traceoptions file size 100m
set routing-instances evpn-vpws-mh protocols evpn traceoptions flag all
set routing-instances evpn-vpws-mh interface ge-0/0/1.1
set routing-instances evpn-vpws-mh route-distinguisher 7.7.7.7:12
set routing-instances evpn-vpws-mh vrf-import vrf-import-evpn-vpws-mh
set routing-instances evpn-vpws-mh vrf-export vrf-export-evpn-vpws-mh

```

CE12:

```
set interfaces ge-0/0/0 description pe1-ce12
set interfaces ge-0/0/0 gigether-options 802.3ad ae0
set interfaces ge-0/0/1 description pe2-ce12
set interfaces ge-0/0/1 gigether-options 802.3ad ae0
set interfaces ae0 vlan-tagging
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 unit 0 description evpn-vpws-mh-alpha-pe1-pe2
set interfaces ae0 unit 0 vlan-id 30
set interfaces ae0 unit 0 family inet address 10.3.1.12/24
```

CE3:

```
set interfaces ge-0/0/0 vlan-tagging
set logical-systems CE3 interfaces ge-0/0/0 unit 2 description ce3-pe3
set logical-systems CE3 interfaces ge-0/0/0 unit 2 vlan-id 30
set logical-systems CE3 interfaces ge-0/0/0 unit 2 family inet address 10.3.1.3/24
```

### Section 11.2 EVPN VPWS all-active/active-active

Configuration is identical to all-active configuration captured in section 11.2. Only change will be with ESI mode set to single-active instead of all-active.

This config changes over all-active is all it takes to make it single active.

PE1:

```
set interfaces ae0 unit 0 esi single-active
```

PE2:

```
set interfaces ae0 unit 0 esi single-active
```

## Appendix 12: Multi-domain SRv6

### Section 12.2 Convert the topology to L1/L2

PE1:

```
set protocols isis level 2 disable
delete protocols isis level 1 disable
delete protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:213:0:0:0
set protocols isis interface ge-0/0/2.0 level 1 metric 1
delete protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:1:214:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
```

PE2:

```
set protocols isis level 2 disable
delete protocols isis level 1 disable
delete protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:2:224:0:0:0
set protocols isis interface ge-0/0/2.0 level 1 metric 1
delete protocols isis interface ge-0/0/3.0 level 2 srv6-adjacency-segment
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a1:2:223:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
```

Pl:

```
delete protocols isis level 1 disable
set routing-options flex-algorithm 128 definition metric-type delay-metric
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:234:0:0:0
set protocols isis interface ge-0/0/0.0 level 2 metric 10
delete protocols isis interface ge-0/0/0.0 level 1
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:235:0:0:0
set protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
deactivate protocols isis interface ge-0/0/1.0 level 2 post-convergence-lfa
set protocols isis interface ge-0/0/1.0 level 2 metric 1
delete protocols isis interface ge-0/0/1.0 level 1
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:236:0:0:0
set protocols isis interface ge-0/0/2.0 level 2 metric 100
delete protocols isis interface ge-0/0/2.0 level 1
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:231:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
delete protocols isis interface ge-0/0/4.0 level 2
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:1:232:0:0:0
set protocols isis interface ge-0/0/4.0 level 1 metric 2
```

P2:

```
delete protocols isis level 1 disable
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:243:0:0:0
set protocols isis interface ge-0/0/0.0 level 2 metric 10
delete protocols isis interface ge-0/0/0.0 level 1
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:245:0:0:0
set protocols isis interface ge-0/0/1.0 level 2 metric 100
delete protocols isis interface ge-0/0/1.0 level 1
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:246:0:0:0
set protocols isis interface ge-0/0/2.0 level 2 metric 1
delete protocols isis interface ge-0/0/2.0 level 1
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
```

```

end-x-sid 2001:db8:a2:2:241:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
delete protocols isis interface ge-0/0/4.0 level 2
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:2:242:0:0:0
set protocols isis interface ge-0/0/4.0 level 1 metric 1

```

P3:

```

delete protocols isis level 1 disable
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:253:0:0:0
set protocols isis interface ge-0/0/0.0 level 2 metric 1
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:254:0:0:0
set protocols isis interface ge-0/0/1.0 level 2 metric 100
set protocols isis interface ge-0/0/1.0 level 1 disable
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:256:0:0:0
set protocols isis interface ge-0/0/2.0 level 2 metric 10
set protocols isis interface ge-0/0/2.0 level 1 disable
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:257:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
set protocols isis interface ge-0/0/3.0 level 2 disable
delete protocols isis interface ge-0/0/4.0 level 2
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:3:258:0:0:0
set protocols isis interface ge-0/0/4.0 level 1 metric 1
set protocols isis interface ge-0/0/4.0 level 2 disable

```

P4:

```

delete protocols isis level 1 disable
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:263:0:0:0
set protocols isis interface ge-0/0/0.0 level 2 metric 100
set protocols isis interface ge-0/0/0.0 level 1 disable
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:264:0:0:0
set protocols isis interface ge-0/0/1.0 level 2 metric 1
set protocols isis interface ge-0/0/1.0 level 1 disable
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:265:0:0:0
set protocols isis interface ge-0/0/2.0 level 2 metric 10
set protocols isis interface ge-0/0/2.0 level 1 disable
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:267:0:0:0 flavor psp
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:267:0:0:0 flavor usp
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:267:0:0:0 flavor usd
set protocols isis interface ge-0/0/3.0 level 1 metric 1
set protocols isis interface ge-0/0/3.0 level 2 disable
delete protocols isis interface ge-0/0/4.0 level 2

```

```
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a2:4:268:0:0:0
set protocols isis interface ge-0/0/4.0 level 1 metric 1
set protocols isis interface ge-0/0/4.0 level 2 disable
```

PE3:

```
set protocols isis level 2 disable
delete protocols isis level 1 disable
set routing-options flex-algorithm 128 definition metric-type delay-metric
delete protocols isis interface ge-0/0/2.0 level 2
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:1:275:0:0:0
set protocols isis interface ge-0/0/2.0 level 1 metric 1
set protocols isis interface ge-0/0/2.0 level 2 disable
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:1:276:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 5
set protocols isis interface ge-0/0/3.0 level 2 disable
```

PE4:

```
set protocols isis level 2 disable
delete protocols isis level 1 disable
delete protocols isis interface ge-0/0/2.0 level 2
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:2:285:0:0:0
set protocols isis interface ge-0/0/2.0 level 1 metric 1
set protocols isis interface ge-0/0/2.0 level 2 disable
delete protocols isis interface ge-0/0/3.0 level 2
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc1
end-x-sid 2001:db8:a3:2:286:0:0:0
set protocols isis interface ge-0/0/3.0 level 1 metric 1
set protocols isis interface ge-0/0/3.0 level 2 disable
```

PE1:

```
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:1:213:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:1:214:0:0:0 flavor usd psp usp
```

PE2:

```
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:2:223:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b1:2:224:0:0:0 flavor usd psp usp
```

P1:

```
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:234:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/0.0 delay-metric 10
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
```

```

end-x-sid 2001:db8:b2:1:235:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/1.0 delay-metric 100
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:236:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:231:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:1:232:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/4.0 delay-metric 1

```

P2:

```

set routing-options source-packet-routing srv6 locator loc2 2001:db8:b2:2::/64 algorithm 128
set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:243:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/0.0 delay-metric 10
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:245:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/1.0 delay-metric 10
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:246:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 100
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:241:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:2:242:0:0:0 flavor usd psp usp

```

P3:

```

set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:253:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/0.0 delay-metric 100
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:254:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/1.0 delay-metric 10
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:256:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 10
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:257:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:3:258:0:0:0 flavor usd psp usp

```

P4:

```

set protocols isis interface ge-0/0/0.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:263:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/0.0 delay-metric 1
set protocols isis interface ge-0/0/1.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:264:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/1.0 delay-metric 100
set protocols isis interface ge-0/0/2.0 level 2 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:265:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 10

```

```
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:267:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/3.0 delay-metric 1
set protocols isis interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b2:4:268:0:0:0 flavor usd psp usp
```

PE3:

```
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:1:275:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:1:276:0:0:0 flavor usd psp usp
```

PE4:

```
set protocols isis interface ge-0/0/2.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:2:285:0:0:0 flavor usd psp usp
set protocols isis interface ge-0/0/2.0 delay-metric 1
set protocols isis interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected locator loc2
end-x-sid 2001:db8:b3:2:286:0:0:0 flavor usd psp usp
```

## Section 12.2.2 Summarization and leaking across L1/L2

P1:

```
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
delete policy-options policy-statement leak_to_isis
set policy-options policy-statement leak_to_isis term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a2::/48 exact
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis term 1 to level 1
set policy-options policy-statement leak_to_isis term 1 then advertise-locator
set policy-options policy-statement leak_to_isis term 1 then set-down-bit
set policy-options policy-statement leak_to_isis term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 1 then accept

set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis term 2 to level 2
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis
commit
```

P2:

```

set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
delete policy-options policy-statement leak_to_isis
set policy-options policy-statement leak_to_isis term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a2::/48 exact
set policy-options policy-statement leak_to_isis term 1 to level 1
set policy-options policy-statement leak_to_isis term 1 then advertise-locator
set policy-options policy-statement leak_to_isis term 1 then set-down-bit
set policy-options policy-statement leak_to_isis term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 1 then accept

set policy-options policy-statement leak_to_isis term 1a from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis term 1a to level 1
set policy-options policy-statement leak_to_isis term 1a then advertise-locator
set policy-options policy-statement leak_to_isis term 1a then set-down-bit
set policy-options policy-statement leak_to_isis term 1a then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 1a then accept

set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis term 2 to level 2
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis

```

P3:

```

set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
delete policy-options policy-statement leak_to_isis
set policy-options policy-statement leak_to_isis term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a2::/48 exact
set policy-options policy-statement leak_to_isis term 1 to level 1
set policy-options policy-statement leak_to_isis term 1 then advertise-locator
set policy-options policy-statement leak_to_isis term 1 then set-down-bit
set policy-options policy-statement leak_to_isis term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 1 then accept

set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis term 2 to level 2

```

```

set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis

```

P4:

```

set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
delete policy-options policy-statement leak_to_isis
set policy-options policy-statement leak_to_isis term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis term 1 from route-filter 2001:db8:a2::/48 exact
set policy-options policy-statement leak_to_isis term 1 then advertise-locator
set policy-options policy-statement leak_to_isis term 1 then set-down-bit
set policy-options policy-statement leak_to_isis term 1 to level 1
set policy-options policy-statement leak_to_isis term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 1 then accept

set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 to level 2
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis

```

### Section 12.2.9 Summarization and leaking flex-algo locators across L1/L2

P1:

```

set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48

```

```

set policy-options policy-statement leak_to_isis term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b2::/48 exact
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis term 4 then advertise-locator
set policy-options policy-statement leak_to_isis term 4 then set-down-bit
set policy-options policy-statement leak_to_isis term 4 to level 1
set policy-options policy-statement leak_to_isis term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 4 then accept

set policy-options policy-statement leak_to_isis term 4a from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis term 4a to level 1
set policy-options policy-statement leak_to_isis term 4a then advertise-locator
set policy-options policy-statement leak_to_isis term 4a then set-down-bit
set policy-options policy-statement leak_to_isis term 4a then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 4a then accept

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 to level 2
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

set protocols isis export leak_to_isis

```

P2:

```

set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48

set policy-options policy-statement leak_to_isis term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b2::/48 exact
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis term 4 then advertise-locator
set policy-options policy-statement leak_to_isis term 4 then set-down-bit
set policy-options policy-statement leak_to_isis term 4 to level 1
set policy-options policy-statement leak_to_isis term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 4 then accept

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 to level 2
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

```

```
set protocols isis export leak_to_isis
```

P3:

```

set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48

set policy-options policy-statement leak_to_isis term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b2::/48 exact
set policy-options policy-statement leak_to_isis term 4 then advertise-locator

```

```

set policy-options policy-statement leak_to_isis term 4 then set-down-bit
set policy-options policy-statement leak_to_isis term 4 to level 1
set policy-options policy-statement leak_to_isis term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 4 then accept

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 to level 2
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

set protocols isis export leak_to_isis

```

P4:

```

set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48

set policy-options policy-statement leak_to_isis term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis term 4 from route-filter 2001:db8:b2::/48 exact
set policy-options policy-statement leak_to_isis term 4 then advertise-locator
set policy-options policy-statement leak_to_isis term 4 then set-down-bit
set policy-options policy-statement leak_to_isis term 4 to level 1
set policy-options policy-statement leak_to_isis term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 4 then accept

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 to level 2
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

set protocols isis export leak_to_isis

```

### Section 12.3 Multi-instance ISIS

Pl:

```

set protocols isis-instance metro1 source-packet-routing srv6 locator loc1
set protocols isis-instance metro1 source-packet-routing srv6 locator loc2
set routing-options flex-algorithm 128 definition metric-type delay-metric
set protocols isis-instance metro1 source-packet-routing flex-algorithm 128
delete protocols isis interface ge-0/0/3.0
set protocols isis-instance metro1 interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:1:241:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/3.0 level 1 metric 1
set protocols isis-instance metro1 interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:1:241:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/3.0 level 1 metric 1
set protocols isis-instance metro1 interface ge-0/0/3.0 point-to-point
set protocols isis-instance metro1 interface ge-0/0/3.0 level 2 disable
delete protocols isis interface ge-0/0/4.0
set protocols isis-instance metro1 interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:1:232:0:0:0 flavor usd psp usp

```

```

set protocols isis-instance metro1 interface ge-0/0/4.0 delay-metric 1
set protocols isis-instance metro1 interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:1:232:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/4.0 level 1 metric 2
set protocols isis-instance metro1 interface ge-0/0/4.0 point-to-point
set protocols isis-instance metro1 interface ge-0/0/4.0 level 2 disable
commit

```

Router P2:

```

set protocols isis-instance metro1 source-packet-routing srv6 locator loc1
set protocols isis-instance metro1 source-packet-routing srv6 locator loc2
set protocols isis-instance metro1 source-packet-routing flex-algorithm 128
delete protocols isis interface ge-0/0/3.0
set protocols isis-instance metro1 interface ge-0/0/3.0 point-to-point
set protocols isis-instance metro1 interface ge-0/0/3.0 level 2 disable
set protocols isis-instance metro1 interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:2:241:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/3.0 delay-metric 1
set protocols isis-instance metro1 interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:2:241:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/3.01 delay-metric 1
delete protocols isis interface ge-0/0/4.0
set protocols isis-instance metro1 interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:2:242:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/4.0 delay-metric 1
set protocols isis-instance metro1 interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:2:242:0:0:0 flavor usd psp usp
set protocols isis-instance metro1 interface ge-0/0/4.0 level 1 metric 1
set protocols isis-instance metro1 interface ge-0/0/4.0 point-to-point
set protocols isis-instance metro1 interface ge-0/0/4.0 level 2 disable

```

P3:

```

set protocols isis-instance metro2 source-packet-routing srv6 locator loc1
set protocols isis-instance metro2 source-packet-routing srv6 locator loc2
set protocols isis-instance metro2 source-packet-routing flex-algorithm 128
set routing-options flex-algorithm 128 definition metric-type delay-metric
delete protocols isis interface ge-0/0/3.0
set protocols isis-instance metro2 interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:3:257:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/3.0 level 1 metric 1
set protocols isis-instance metro2 interface ge-0/0/3.0 level 2 disable
set protocols isis-instance metro2 interface ge-0/0/3.0 point-to-point
set protocols isis-instance metro2 interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:3:257:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/3.0 delay-metric 1
delete protocols isis interface ge-0/0/4.0
set protocols isis-instance metro2 interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:3:258:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/4.0 delay-metric 1
set protocols isis-instance metro2 interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:3:258:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/4.0 level 1 metric 1
set protocols isis-instance metro2 interface ge-0/0/4.0 level 2 disable
set protocols isis-instance metro2 interface ge-0/0/4.0 point-to-point

```

P4:

```

set protocols isis-instance metro2 source-packet-routing srv6 locator loc1
set protocols isis-instance metro2 source-packet-routing srv6 locator loc2
set protocols isis-instance metro2 source-packet-routing flex-algorithm 128
delete protocols isis interface ge-0/0/3.0
set protocols isis-instance metro2 interface ge-0/0/3.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:4:267:0:0:0 flavor psp usp usd
set protocols isis-instance metro2 interface ge-0/0/3.0 level 1 metric 1
set protocols isis-instance metro2 interface ge-0/0/3.0 level 2 disable
set protocols isis-instance metro2 interface ge-0/0/3.0 point-to-point
set protocols isis interfacemetro2 interface ge-0/0/3.0 level 2 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:4:267:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/3.0 delay-metric 1

delete protocols isis interface ge-0/0/4.0
set protocols isis-instance metro2 interface ge-0/0/4.0 level 2 srv6-adjacency-segment unprotected
locator loc2 end-x-sid 2001:db8:b2:4:268:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/4.0 delay-metric 1
set protocols isis-instance metro2 interface ge-0/0/4.0 level 1 srv6-adjacency-segment unprotected
locator loc1 end-x-sid 2001:db8:a2:4:268:0:0:0 flavor usd psp usp
set protocols isis-instance metro2 interface ge-0/0/4.0 level 1 metric 1
set protocols isis-instance metro2 interface ge-0/0/4.0 level 2 disable
set protocols isis-instance metro2 interface ge-0/0/4.0 point-to-point

```

### Section 12.3.1 Summarization and leaking across multi-instance

P1:

```

#Configure aggregate routes
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
#Policy to leak into metro1
set policy-options policy-statement leak_to_isis_metro term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis_metro term 1 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 1 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 1 then accept

#Policy to leak into ISIS default
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a1::/52 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept
#Leak loopback addresses
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:7:255:7::7/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:8:255:8::8/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact

```

```

set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro1 export leak_to_isis_metro

```

P2:

```

#configure aggregate routes
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
#Policy to leak into metro1
set policy-options policy-statement leak_to_isis_metro term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a3::/48 exact
set policy-options policy-statement leak_to_isis_metro term 1 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 1 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 1 then accept

#Policy to leak into ISIS default
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a1::/52 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept
#Policy to leak loopback addresses
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:7:255:7::7/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:8:255:8::8/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:1:255:1::1/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:2:255:2::2/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro1 export leak_to_isis_metro

```

P3:

```

#Configure aggregate routes
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/52

#Policy to leak into metro1
set policy-options policy-statement leak_to_isis_metro term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 1 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 1 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 1 then accept

```

```
#Policy to leak into ISIS default
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a3::/52 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

#Policy to leak loopback addresses
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:1:255:1::1/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:2:255:2::2/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro2 export leak_to_isis_metro
```

P4:

```
#Configure aggregate routes
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/52

#Policy to leak into metro2
set policy-options policy-statement leak_to_isis_metro term 1 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a1::/48 exact
set policy-options policy-statement leak_to_isis_metro term 1 from route-filter 2001:db8:a2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 1 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 1 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 1 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 1 then accept

#Policy to leak into ISIS default
set policy-options policy-statement leak_to_isis term 2 from protocol aggregate
set policy-options policy-statement leak_to_isis term 2 from route-filter 2001:db8:a3::/52 exact
set policy-options policy-statement leak_to_isis term 2 then advertise-locator
set policy-options policy-statement leak_to_isis term 2 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 2 then accept

#Policy to leak loopback addresses
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:7:255:7::7/128
exact
set policy-options policy-statement leak_to_isis term 3 from route-filter 2001:db8:8:255:8::8/128
exact
set policy-options policy-statement leak_to_isis term 3 then accept
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:1:255:1::1/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 from route-filter
2001:db8:2:255:2::2/128 exact
set policy-options policy-statement leak_to_isis_metro term 3 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro2 export leak_to_isis_metro
```

### Section 12.3.8 Summarization and leaking flex-algo locators across multi-instance

P1:

#Configure aggregate routes

```
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48
```

#Policy to leak into metro1

```
set policy-options policy-statement leak_to_isis_metro term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis_metro term 4 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 4 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 4 then accept
```

#Policy to leak into default

```
set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b1::/52 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept
```

```
set protocols isis export leak_to_isis
set protocols isis-instance metro1 export leak_to_isis_metro
```

P2:

#Configure aggregate routes

```
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b1::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/48
```

#Policy to leak into metro1

```
set policy-options policy-statement leak_to_isis_metro term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b3::/48 exact
set policy-options policy-statement leak_to_isis_metro term 4 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 4 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 4 then accept
#Policy to leak into ISIS default
```

```

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b1::/52 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro1 export leak_to_isis_metro

```

P3:

#Configure aggregate routes

```

set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/52

```

#Policy to leak into metro2

```

set policy-options policy-statement leak_to_isis_metro term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 4 then advertise-locator
set policy-options policy-statement leak_to_isis_metro term 4 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 4 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis_metro term 4 then accept

```

#Policy to leak into default

```

set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b3::/52 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

```

```

set protocols isis export leak_to_isis
set protocols isis-instance metro2 export leak_to_isis_metro

```

P4:

#Configure aggregate routes

```

set routing-options rib inet6.0 aggregate route 2001:db8:b1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:b2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:b3::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a1::/48
set routing-options rib inet6.0 aggregate route 2001:db8:a2::/52
set routing-options rib inet6.0 aggregate route 2001:db8:a3::/52

```

#Policy to leak into metro2

```

set policy-options policy-statement leak_to_isis_metro term 4 from protocol aggregate
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b1::/48 exact
set policy-options policy-statement leak_to_isis_metro term 4 from route-filter 2001:db8:b2::/52 exact
set policy-options policy-statement leak_to_isis_metro term 4 then advertise-locator

```

```
set policy-options policy-statement leak_to_isis_metro term 4 then set-down-bit
set policy-options policy-statement leak_to_isis_metro term 4 then prefix-attribute-flags anycast-
flag
set policy-options policy-statement leak_to_isis_metro term 4 then accept

#Policy to leak into default ISIS
set policy-options policy-statement leak_to_isis term 5 from protocol aggregate
set policy-options policy-statement leak_to_isis term 5 from route-filter 2001:db8:b3::/52 exact
set policy-options policy-statement leak_to_isis term 5 then advertise-locator
set policy-options policy-statement leak_to_isis term 5 then prefix-attribute-flags anycast-flag
set policy-options policy-statement leak_to_isis term 5 then accept

set protocols isis export leak_to_isis
set protocols isis-instance metro2 export leak_to_isis_metro
```

## Appendix 13: cRPD Support

The configuration is not required as its already covered in LAB 13.3 section.

## Key Segment Routing Resources

The authors of this book highly recommend the following SR resources. Also examine the [Juniper TechLibrary](#) and its up to date information and specifications.

The following books are available on Segment Routing at <https://www.juniper.net/dayone>:

- *Day One: Configuring Segment Routing With Junos*
- *Day One: Migrating To Segment Routing*
- *Day One: Inside Segment Routing*