Configure Segment Routing for BGP

Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) that allows you to create loop-free inter-domain routing between autonomous systems. An autonomous system is a set of routers under a single technical administration. Routers in an autonomous system can use multiple Interior Gateway Protocols (IGPs) to exchange routing information inside the autonomous system and an EGP to route packets outside the autonomous system.

This module provides the configuration information used to enable Segment Routing for BGP.

Note For additional information on implementing BGP on your router, see the *BGP Configuration Guide for Cisco 8000 Series Routers*.

Segment Routing for BGP

In a traditional BGP-based data center (DC) fabric, packets are forwarded hop-by-hop to each node in the autonomous system. Traffic is directed only along the external BGP (eBGP) multipath ECMP. No traffic engineering is possible.

In an MPLS-based DC fabric, the eBGP sessions between the nodes exchange BGP labeled unicast (BGP-LU) network layer reachability information (NLRI). An MPLS-based DC fabric allows any leaf (top-of-rack or border router) in the fabric to communicate with any other leaf using a single label, which results in higher packet forwarding performance and lower encapsulation overhead than traditional BGP-based DC fabric. However, since each label value might be different for each hop, an MPLS-based DC fabric is more difficult to troubleshoot and more complex to configure.

BGP has been extended to carry segment routing prefix-SID index. BGP-LU helps each node learn BGP prefix SIDs of other leaf nodes and can use ECMP between source and destination. Segment routing for BGP simplifies the configuration, operation, and troubleshooting of the fabric. With segment routing for BGP, you can enable traffic steering capabilities in the data center using a BGP prefix SID.

Configure BGP Prefix Segment Identifiers

Segments associated with a BGP prefix are known as BGP prefix SIDs. The BGP prefix SID is global within a segment routing or BGP domain. It identifies an instruction to forward the packet over the ECMP-aware best-path computed by BGP to the related prefix. The BGP prefix SID is manually configured from the segment routing global block (SRGB) range of labels.

Each BGP speaker must be configured with an SRGB using the **segment-routing global-block** command. See the About the Segment Routing Global Block section for information about the SRGB.

Note You must enable SR and explicitly configure the SRGB before configuring SR BGP. The SRGB must be explicitly configured, even if you are using the default range (16000 – 23999). BGP uses the SRGB and the index in the BGP prefix-SID attribute of a learned BGP-LU advertisement to allocate a local label for a given destination.

If SR and the SRGB are enabled after configuring BGP, then BGP is not aware of the SRGB, and therefore it allocates BGP-LU local labels from the dynamic label range instead of from the SRGB. In this case, restart the BGP process in order to allocate BGP-LU local labels from the SRGB.

Note Because the values assigned from the range have domain-wide significance, we recommend that all routers within the domain be configured with the same range of values.

To assign a BGP prefix SID, first create a routing policy using the **set label-index** *index* attribute, then associate the index to the node.

Note A routing policy with the **set label-index** attribute can be attached to a network configuration or redistribute configuration. Other routing policy language (RPL) configurations are possible. For more information on routing policies, refer to the "Implementing Routing Policy" chapter in the *Routing Configuration Guide for Cisco 8000 Series Routers*.

Example

The following example shows how to configure the SRGB, create a BGP route policy using a $SID parameter and **set label-index** attribute, and then associate the prefix-SID index to the node.

```
RP/0/RP0/CPU0:router(config)# segment-routing global-block 16000 23999

RP/0/RP0/CPU0:router(config)# route-policy SID($SID)
RP/0/RP0/CPU0:router(config-rpl)# set label-index $SID
RP/0/RP0/CPU0:router(config-rpl)# end policy

RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 1.1.1.1
```

```
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# network 1.1.1.3/32 route-policy SID(3)
RP/0/RP0/CPU0:router(config-bgp-af)# allocate-label all
RP/0/RP0/CPU0:router(config-bgp-af)# commit
RP/0/RP0/CPU0:router(config-bgp-af)# end


RP/0/RP0/CPU0:router# show bgp 1.1.1.3/32
BGP routing table entry for 1.1.1.3/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 74          74
    Local Label: 16003
Last Modified: Sep 29 19:52:18.155 for 00:07:22
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.2
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.2
  3
    99.3.21.3 from 99.3.21.3 (1.1.1.3)
      Received Label 3
      Origin IGP, metric 0, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 74
      Origin-AS validity: not-found
      Label Index: 3
```

Segment Routing Egress Peer Engineering

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
BGP PeerSet SID	Release 7.3.2	BGP peer SIDs are used to express source-routed interdomain paths and are of two types: Peer Node SIDs and Peer Adjacency SIDs. This release supports a new type of BGP peering SID, called BGP Peer Set SID. It is a group or set of BGP peer SIDs, that can provide load balancing over BGP neighbors (nodes) or links (adjacencies). The BGP peer Set SID can be associated with any combination of Peer Node SIDs or Peer Adjacency SIDs.

Segment routing egress peer engineering (EPE) uses a controller to instruct an ingress provider edge, or a content source (node) within the segment routing domain, to use a specific egress provider edge (node) and a specific external interface to reach a destination. BGP peer SIDs are used to express source-routed inter-domain paths.

Below are the BGP-EPE peering SID types:

- PeerNode SID—To an eBGP peer. Pops the label and forwards the traffic on any interface to the peer.
- PeerAdjacency SID—To an eBGP peer via interface. Pops the label and forwards the traffic on the related interface.
- PeerSet SID—To a set of eBGP peers. Pops the label and forwards the traffic on any interface to the set of peers. All the peers in a set might not be in the same AS.

  Multiple PeerSet SIDs can be associated with any combination of PeerNode SIDs or PeerAdjacency SIDs.

The controller learns the BGP peer SIDs and the external topology of the egress border router through BGP-LS EPE routes. The controller can program an ingress node to steer traffic to a destination through the egress node and peer node using BGP labeled unicast (BGP-LU).

EPE functionality is only required at the EPE egress border router and the EPE controller.

Usage Guidelines and Limitations

- When enabling BGP EPE, you must enable MPLS encapsulation on the egress interface connecting to the eBGP peer. This can be done by enabling either BGP labeled unicast (BGP-LU) address family or MPLS static for the eBGP peer.

  For information about BGP-LU, refer to the Implementing BGP chapter in the *BGP Configuration Guide for Cisco 8000 Series Routers*.

  For information about MPLS static, refer to the Implementing MPLS Static Labeling chapter in the *MPLS Configuration Guide for Cisco 8000 Series Routers*.
- Note the following points related to the IP-lookup backup support for EPEs:
  - This feature works only when you enable the **epe backup enable**, under the Global Address Family ID (AFI).
  - With this feature, an IP-Lookup backup is installed for each Egress Peer Engineering. This means, when all the paths of that EPE go down, the Forwarding Information Base (FIB) table searches in the IP table for the destination IP address in the data packet and forwards them accordingly.
  - The peer-set EPEs have a backup installed only when the mentioned CLI knob is enabled.

Configure Segment Routing Egress Peer Engineering

This task explains how to configure segment routing EPE on the EPE egress node.

SUMMARY STEPS

1. **router bgp** *as-number*
2. **neighbor** *ip-address*
3. **remote-as** *as-number*
4. **egress-engineering**
5. **exit**
6. **mpls static**
7. **interface** *type interface-path-id*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**router bgp** *as-number* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 1`	Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 2	**neighbor** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.10.10.2`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 3	**remote-as** *as-number* **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 3`	Creates a neighbor and assigns a remote autonomous system number to it.
Step 4	**egress-engineering** **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)# egress-engineering`	Configures the egress node with EPE for the eBGP peer.

Step 5	**exit**
	Example:
	`RP/0/RP0/CPU0:router(config-bgp-nbr)# exit`
	`RP/0/RP0/CPU0:router(config-bgp)# exit`
	`RP/0/RP0/CPU0:router(config)#`	

Step 6	**mpls static** **Example:** `RP/0/RP0/CPU0:router(config)# mpls static`	Configure MPLS static on the egress interface connecting to the eBGP peer.
Step 7	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-mpls-static)# interface`	Specifies the egress interface connecting to the eBGP peer.

	Command or Action	Purpose
	GigabitEthernet0/0/1/2	

```
router bgp 1
 neighbor 10.10.10.2
  remote-as 3
  egress-engineering
 !
!
mpls static
 interface GigabitEthernet0/0/1/2
!
```

Configuring Manual BGP-EPE Peering SIDs

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Manual BGP-EPE Peer SIDs	Release 7.3.2	BGP Peering SIDs that are allocated dynamically are not persistent and can be reallocated after a reload or a process restart. This feature allows you to manually configure BGP Egress Peer Engineering (EPE) Peering SIDs. This functionality provides predictability, consistency, and reliability if there are system reloads or process restarts.

Configuring manual BGP-EPE Peer SIDs allows for persistent EPE label values. Manual BGP-EPE SIDs are advertised through BGP-LS and are allocated from the Segment Routing Local Block (SRLB). See Configure Segment Routing Global Block and Segment Routing Local Block for information about the SRLB.

Each PeerNode SID, PeerAdjacency SID, and PeerSet SID is configured with an index value. This index serves as an offset from the configured SRLB start value and the resulting MPLS label (SRLB start label + index) is assigned to these SIDs. This label is used by CEF to perform load balancing across the individual BGP PeerSet SIDs, BGP PeerNode SID, or ultimately across each first-hop adjacency associated with that BGP PeerNode SID or BGP PeerSet SID.

Configuring Manual PeerNode SID

Each eBGP peer will be associated with a PeerNode SID index that is configuration driven.

```
RP/0/0/CPU0:PE1(config)# router bgp 10
RP/0/0/CPU0:PE1(config-bgp)# neighbor 10.10.10.2
RP/0/0/CPU0:PE1(config-bgp-nbr)# remote-as 20
RP/0/0/CPU0:PE1(config-bgp-nbr)# egress-engineering
RP/0/0/CPU0:PE1(config-bgp-nbr)# peer-node-sid index 600
```

Configuring Manual PeerAdjacency SID

Any first-hop for which an adjacency SID is configured needs to be in the resolution chain of at least one eBGP peer that is configured for egress-peer engineering. Otherwise such a kind of "orphan" first-hop with regards to BGP has no effect on this feature. This is because BGP only understands next-hops learnt by the BGP protocol itself and in addition only the resolving IGP next-hops for those BGP next-hops.

```
RP/0/0/CPU0:PE1(config)# router bgp 10
RP/0/0/CPU0:PE1(config-bgp)# adjacencies
RP/0/0/CPU0:PE1(config-bgp-adj)# 1.1.1.2
RP/0/0/CPU0:PE1(config-bgp-adj)# adjacency-sid index 500
```

Configuring Manual PeerSet SID

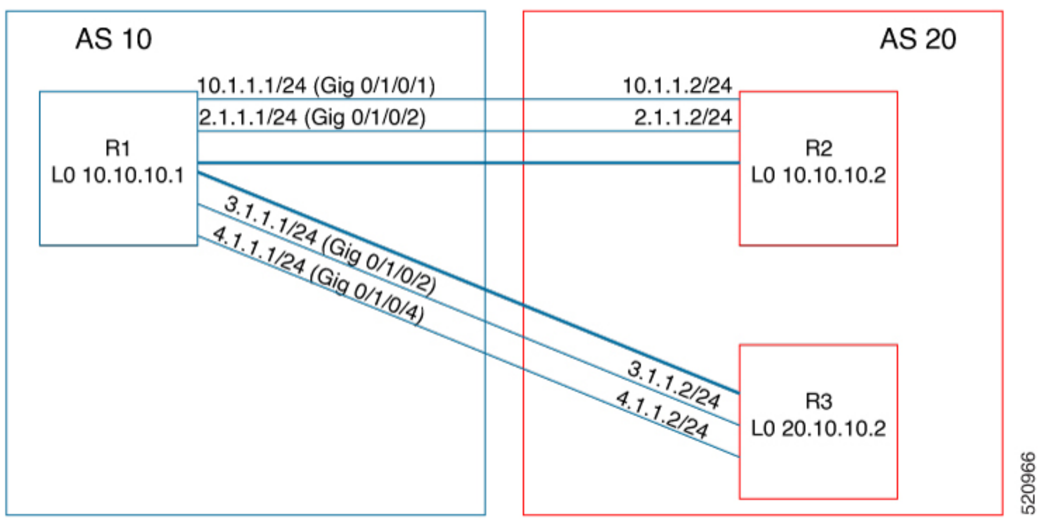
The PeerSet SID is configured under global Address Family. This configuration results in the creation of a Peer-Set SID EPE object.

```
RP/0/0/CPU0:PE1(config)# router bgp 10
RP/0/0/CPU0:PE1(config-bgp)# address-family ipv4 unicast
RP/0/0/CPU0:PE1(config-bgp-afi)# peer-set-id 1
RP/0/0/CPU0:PE1(config-bgp-peer-set)# peer-set-sid 300
```

Example

Topology

The example in this section uses the following topology.



In this example, BGP-EPE peer SIDs are allocated from the default SRLB label range (15000 – 15999). The BGP-EPE peer SIDs are configured as follows:

- PeerNode SIDs to 10.10.10.2 with index 600 (label 15600), and for 20.10.10.2 with index 700 (label 15700)
- PeerAdj SID to link 1.1.1.2 with index 500 (label 15500)
- PeerSet SID 1 to load balance over BGP neighbors 10.10.10.1 and 20.10.10.2 with SID index 300 (label 15300)

- PeerSet SID 2 to load balance over BGP neighbor 20.10.10.2 and link 1.1.1.2 with SID index 400 (label 15400)

Configuration on R1

```
router bgp 10
 address-family ipv4 unicast
  peer-set-id 1
   peer-set-sid index 300
  !
  peer-set-id 2
   peer-set-sid index 400
  !
 !
 adjacencies
  1.1.1.2
   adjacency-sid index 500
   peer-set 2
  !
 !
 neighbor 10.10.10.2
  remote-as 20
  egress-engineering
  peer-node-sid index 600
  peer-set 1
 !
 neighbor 20.10.10.2
  egress-engineering
  peer-node-sid index 700
  peer-set 1
  peer-set 2
 !
```

To further show the load balancing of this example:

- 15600 is load balanced over {1.1.1.1 and 2.1.1.1}
- 15700 is load balanced over {3.1.1.1 and 4.1.1.1}
- 15500 is load balanced over {1.1.1.1}
- 15300 is load balanced over {1.1.1.1, 2.1.1.1, 3.1.1.1 and 4.1.1.1}
- 15400 is load balanced over {1.1.1.1, 3.1.1.1 and 4.1.1.1}

Advertising EPE-Enabled BGP Neighbors via BGP-LU

Table 3: Feature History Table

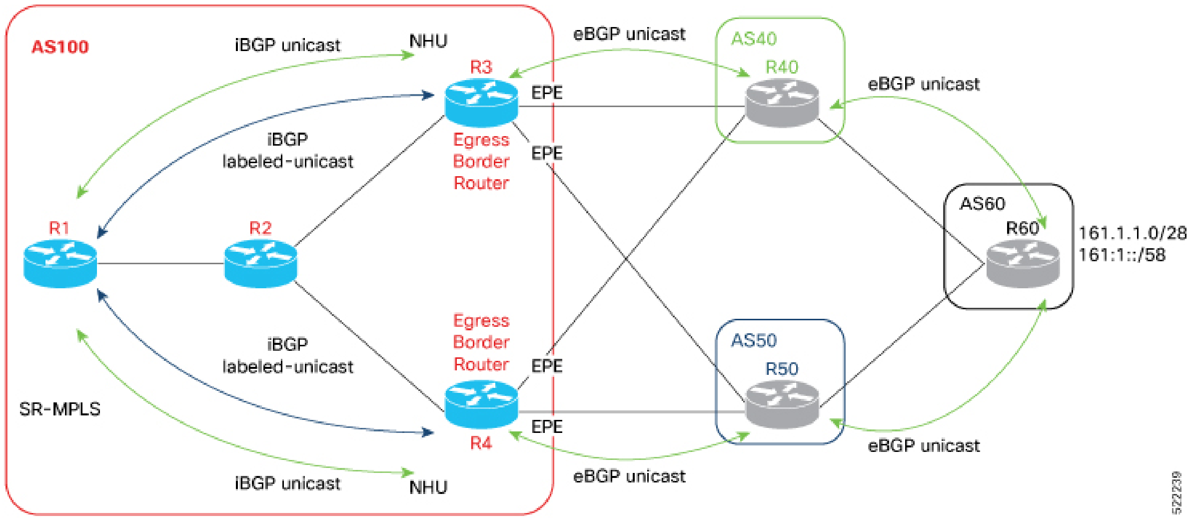
Feature Name	Release	Description
Advertising EPE-Enabled BGP Neighbors via BGP-LU	Release 7.3.3	BGP peering segments/SIDs are part of the Segment Routing Centralized BGP Egress Peer Engineering solution (BGP-EPE). A BGP-EPE-enabled border router allocates and programs BGP peering SIDs (EPE labels) to steer traffic over a specific external interface/BGP neighbor to reach a particular destination. This feature provides an alternate BGP-EPE solution leveraging BGP peering segments. It allows a BGP-EPE-enabled border router to use BGP Labeled Unicast (BGP-LU) to advertise the IP address of a neighbor with an LU label equal to the EPE label assigned to that neighbor.

BGP peering segments/SIDs are part of the Segment Routing Centralized BGP Egress Peer Engineering solution (BGP-EPE), as described in IETF RFC 9087. A BGP-EPE-enabled border router allocates and programs BGP peering SIDs (EPE labels) to steer traffic over a specific external interface/BGP neighbor to reach a particular destination.

This feature provides an alternate BGP-EPE solution leveraging BGP peering segments. It allows a BGP-EPE-enabled border router to use BGP Labeled Unicast (BGP-LU) to advertise the IP address of a neighbor with an LU label equal to the EPE label assigned to that neighbor.

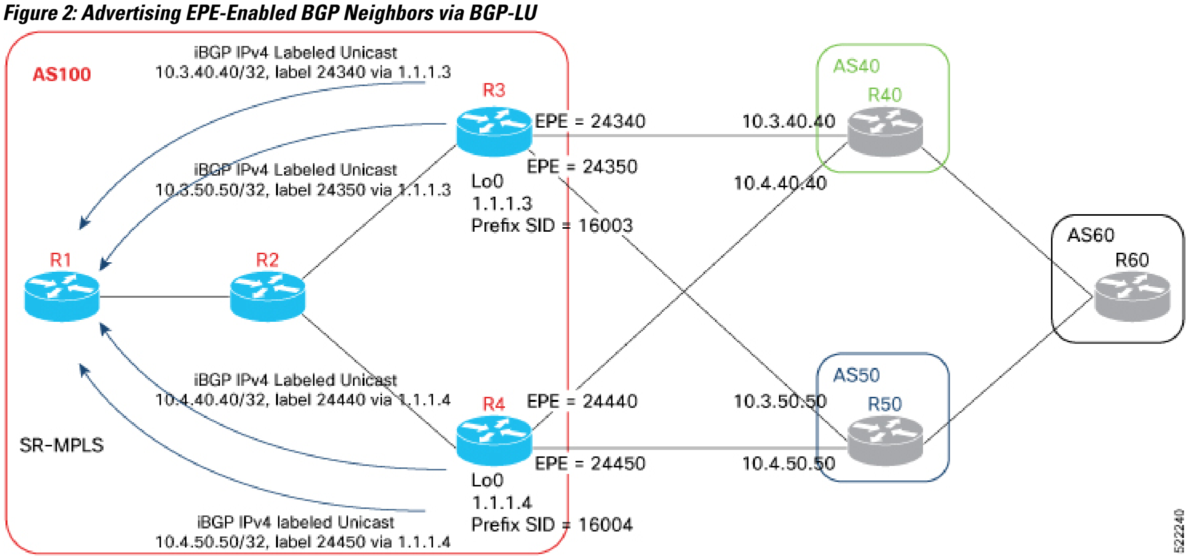
The following figure illustrates a Segment Routing network (AS100) connected to a pair of transit Autonomous Systems. The egress border routers (R3 and R4) have BGP Peering segments (EPE) enabled on their eBGP neighbors in AS40 and AS50. Prefixes are propagated inside AS100 via BGP. R3 and R4 maintain the BGP next-hops unchanged. In addition, BGP labeled unicast is enabled inside AS100 to advertise the IP address of these eBGP neighbors.

Figure 1: Solution Overview



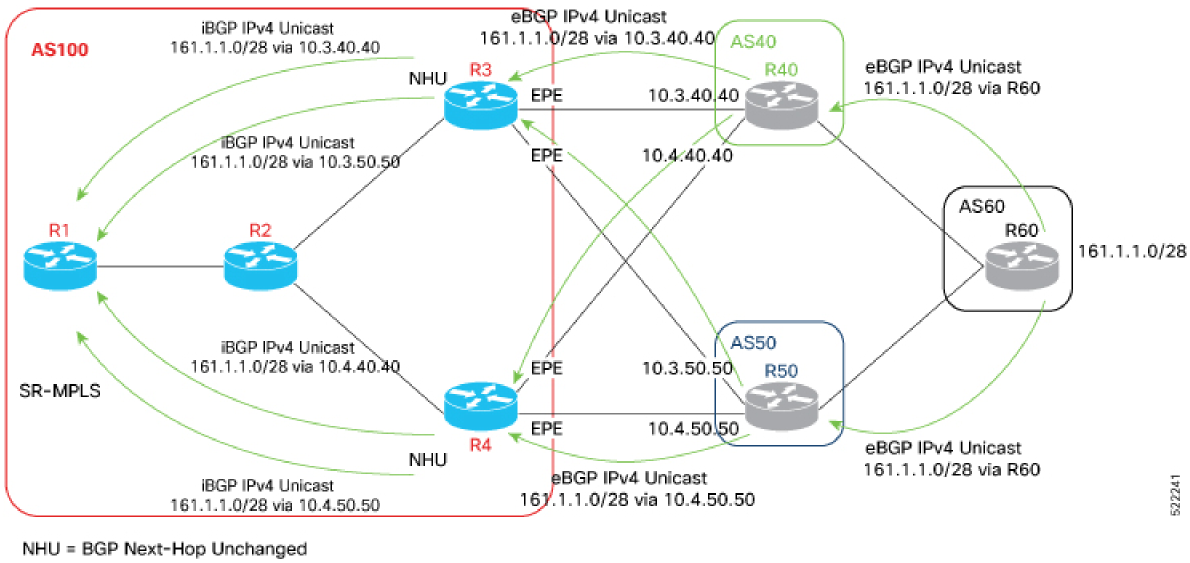
The figure below depicts the BGP-LU advertisements originated by R3 and R4 for the IP addresses of their eBGP neighbors. The figure also indicates the EPE label values assigned to each eBGP neighbor. Note that the local BGP-LU label on the egress border router is equal to the EPE label assigned to that neighbor.

Figure 2: Advertising EPE-Enabled BGP Neighbors via BGP-LU



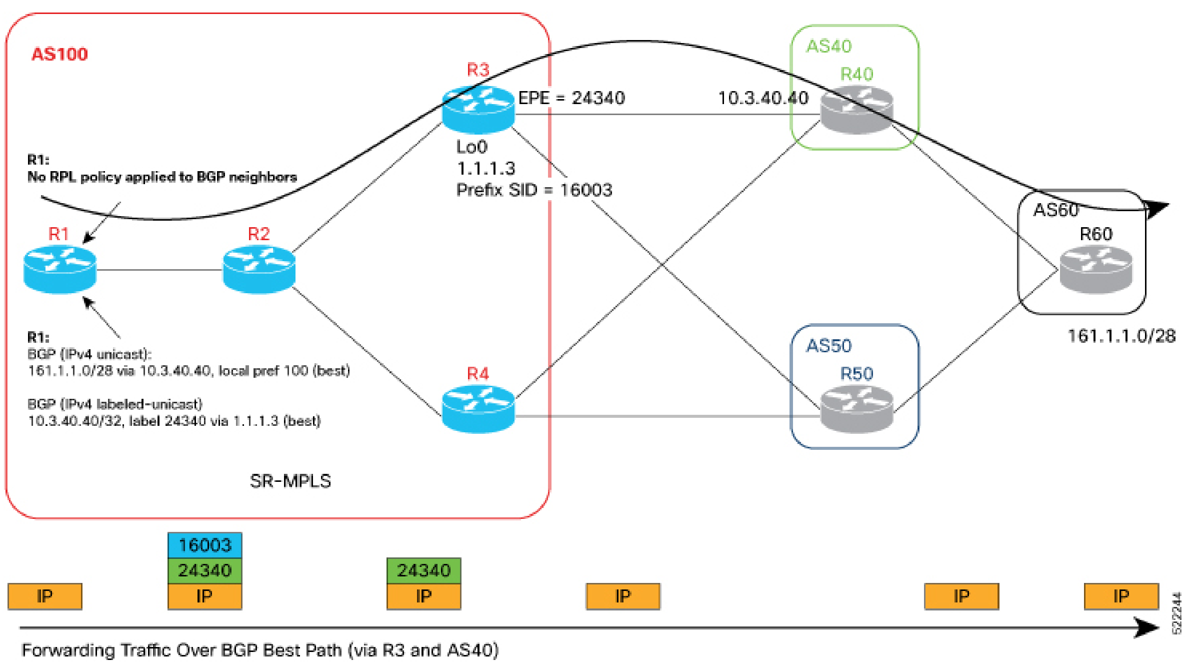
In the following figure, an overlay prefix 161.1.1.0/28 originating at AS60 is advertised inside AS100. Egress border routers are configured to advertise all of their paths. Note that the BGP next-hops are not modified. In this example, the ingress router in AS100 (R1) learns the overlay prefix via 4 paths (one for each eBGP neighbor).

Figure 3: Advertising Overlay Prefixes



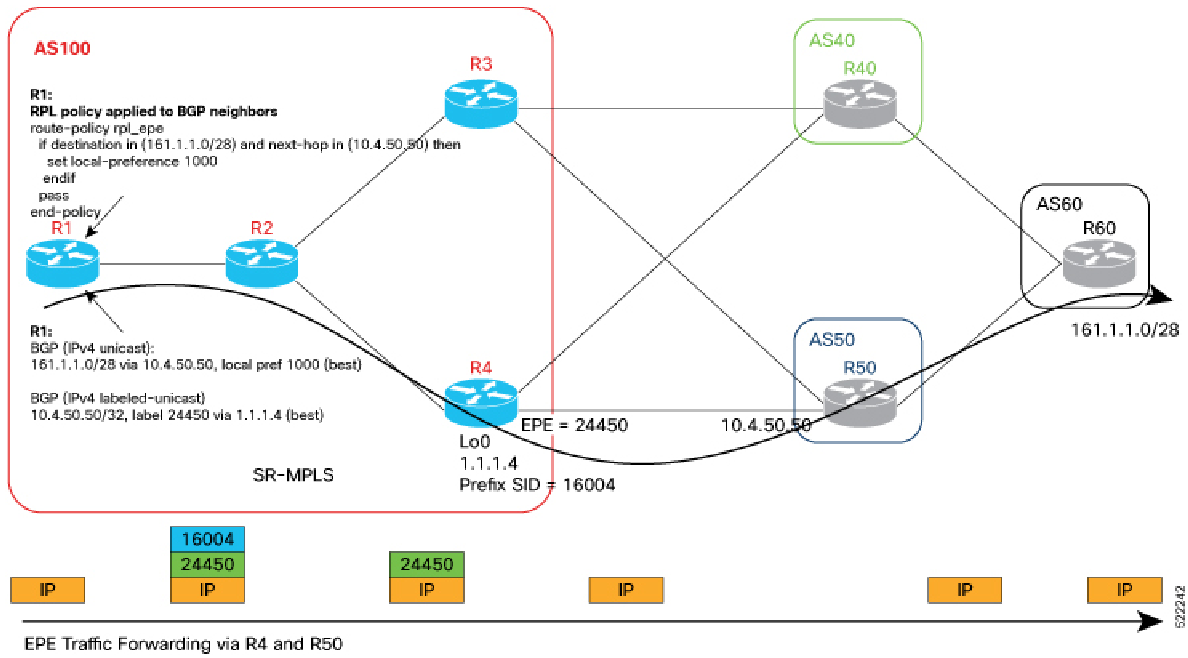
On ingress border router R1, and without any BGP policy modification, assume that BGP selects the path corresponding to AS40 as best path for the overlay prefix 161.1.1.0/28. Its BGP next-hop (10.3.40.40) is learned via BGP-LU from egress border router R3 (1.1.1.3) and with LU label 24340. This label is the EPE label assigned at R3 for the eBGP neighbor to AS40. The EPE local label is programmed as a POP-and-forward toward the interface connecting to AS40. Lastly, R3's loopback (1.1.1.3) and its prefix label (16003) are learned via IS-IS with SR extensions. As a result, incoming traffic matching the 161.1.1.0/28 route is encapsulated at R1 with two MPLS labels (bottom-of-stack label **24340** and top label **16003**) in order to send the traffic to R3 and then to AS40.

Figure 4: Forwarding Traffic Over BGP Best Path (via R3 and AS40)



When the operator wants to modify the exit egress border router and/or an exit AS for a given overlay prefix, a BGP policy can be applied at the ingress border router to influence the best-path selection. In our example, consider that the desired egress path to 161.1.1.0/28 is via R4 and then AS50 (instead of R3 and AS40, as shown in the previous figure). An RPL policy, for example, can be used to assign a higher BGP local preference to the desired path. As a result, incoming traffic matching the 161.1.1.0/28 route is now encapsulated at R1 with two MPLS labels (bottom-of-stack label **24450** and top label **16004**) in order to send the traffic to R4 and then to AS50.

Figure 5: Forwarding Traffic Over EPE Path (via R4 and AS50)



Usage Guidelines and Limitations

The following usage guidelines and limitations apply for this feature:

- BGPv4 and BGPv6 EPE-enabled neighbors are supported.
- BGP peering SIDs (EPE Peer-Node SIDs and Peer-Adjacencies SIDs) allocated dynamically or configured manually can be used as BGP-LU labels when advertising the IP address of an EPE-enabled BGP neighbor via BGP-LU.
- BGP Peer-Set SIDs are not supported.

Enabling Advertisement of EPE-Enabled BGP Neighbors via BGP-LU

To enable advertisement of EPE-enabled BGP neighbors via BGP-LU, use the **advertise epe-bgp labeled-unicast** command in router BGP address family configuration mode.

The following example shows how to enable advertisement of EPE-enabled BGP neighbors via BGP-LU:

```
RP/0/RP0/CPU0:R3(config)# router bgp 100
RP/0/RP0/CPU0:R3(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-af)# advertise epe-bgp labeled-unicast
```

Running Config

```
router bgp 100
  address-family ipv4 unicast
   advertise epe-bgp labeled-unicast
```

Use Case:

This section provides the router configuration and show command outputs of the scenario described in the overview above

Egress Border Router R3 Configuration

Configure the SRGB:

```
RP/0/RP0/CPU0:R3(config)# segment-routing
RP/0/RP0/CPU0:R3(config-sr)# global-block 16000 23999
RP/0/RP0/CPU0:R3(config-sr)# exit
```

Configure the Loopback address:

```
RP/0/RP0/CPU0:R3(config)# interface Loopback0
RP/0/RP0/CPU0:R3(config-if)# ipv4 address 1.1.1.3 255.255.255.255
RP/0/RP0/CPU0:R3(config-if)# exit
```

Configure MPLS Static on the egress interface connecting to the eBGP peer:

```
RP/0/RP0/CPU0:R3(config)# mpls static
RP/0/RP0/CPU0:R3(config-mpls-static)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:R3(config-mpls-static)# exit
```

Enable SR MPLS under IS-IS:

```
RP/0/RP0/CPU0:R3(config)# router isis 1
RP/0/RP0/CPU0:R3(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-isis-af)# segment-routing mpls
RP/0/RP0/CPU0:R3(config-isis-af)# metric-style wide
RP/0/RP0/CPU0:R3(config-isis-af)# exit
```

Configure prefix segment identifier (SID) value on the IS-IS enabled Loopback interface:

```
RP/0/RP0/CPU0:R3(config-isis)# interface Loopback0
RP/0/RP0/CPU0:R3(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-isis-if-af)# prefix-sid absolute 16003
RP/0/RP0/CPU0:R3(config-isis-if-af)# exit
RP/0/RP0/CPU0:R3(config-isis-if)# exit
```

Enable IS-IS in core-facing interface:

```
RP/0/RP0/CPU0:R3(config-isis)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:R3(config-isis-if)# point-to-point
RP/0/RP0/CPU0:R3(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-isis-if-af)# exit
RP/0/RP0/CPU0:R3(config-isis-if)# exit
RP/0/RP0/CPU0:R3(config-isis)# exit
RP/0/RP0/CPU0:R3(config)#
```

Configure a route policy to advertise all BGP paths:

```
RP/0/RP0/CPU0:R3(config)# route-policy rpl_advertise_all_paths
RP/0/RP0/CPU0:R3(config-rpl)# set path-selection all advertise
RP/0/RP0/CPU0:R3(config-rpl)# set path-selection backup 1 install multipath-protect
RP/0/RP0/CPU0:R3(config-rpl)# end-policy
```

Enable advertisement of EPE-enabled BGP neighbors via BGP-LU:

```
RP/0/RP0/CPU0:R3(config)# router bgp 100
RP/0/RP0/CPU0:R3(config-bgp)# bgp router-id 1.1.1.3
RP/0/RP0/CPU0:R3(config-bgp)# ibgp policy out enforce-modifications
RP/0/RP0/CPU0:R3(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-af)# advertise epe-bgp labeled-unicast
RP/0/RP0/CPU0:R3(config-bgp-af)# additional-paths receive
RP/0/RP0/CPU0:R3(config-bgp-af)# additional-paths send
RP/0/RP0/CPU0:R3(config-bgp-af)# additional-paths selection route-policy
rpl_advertise_all_paths
RP/0/RP0/CPU0:R3(config-bgp-af)# allocate-label all
RP/0/RP0/CPU0:R3(config-bgp-af)# exit
```

Enable IPv4 unicast and IPv4 labeled unicast address families on iBGP peer:

```
RP/0/RP0/CPU0:R3(config-bgp)# neighbor 1.1.1.1
RP/0/RP0/CPU0:R3(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:R3(config-bgp-nbr)# update-source Loopback0
RP/0/RP0/CPU0:R3(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# advertise local-labeled-route disable
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R3(config-bgp-nbr)# address-family ipv4 labeled-unicast
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R3(config-bgp-nbr)# exit
```

Enable EPE for the eBGP peers:

```
RP/0/RP0/CPU0:R3(config-bgp)# neighbor 10.3.40.40
RP/0/RP0/CPU0:R3(config-bgp-nbr)# remote-as 40
RP/0/RP0/CPU0:R3(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:R3(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# route-policy pass_all in
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# route-policy pass_all out
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R3(config-bgp-nbr)# exit

RP/0/RP0/CPU0:R3(config-bgp)# neighbor 10.3.50.50
RP/0/RP0/CPU0:R3(config-bgp-nbr)# remote-as 50
RP/0/RP0/CPU0:R3(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:R3(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# route-policy pass_all in
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# route-policy pass_all out
RP/0/RP0/CPU0:R3(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R3(config-bgp-nbr)# exit
RP/0/RP0/CPU0:R3(config-bgp)# exit
RP/0/RP0/CPU0:R3(config)# commit
```

Egress Border Router R3 Running Configuration

```
segment-routing
 global-block 16000 23999
!

interface Loopback0
 ipv4 address 1.1.1.3 255.255.255.255
```

```
mpls static
 interface GigabitEthernet0/0/0/0
!

router isis 1
 is-type level-2-only
 net 47.0000.0000.0003.00
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16003
  !
 !
 interface HundredGigE0/0/0/0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

route-policy rpl_advertise_all_paths
  set path-selection all advertise
  set path-selection backup 1 install multipath-protect
end-policy
!

router bgp 100
 bgp router-id 1.1.1.3
 ibgp policy out enforce-modifications
 address-family ipv4 unicast
  advertise epe-bgp labeled-unicast
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy rpl_advertise_all_paths
  allocate-label all
 !
 neighbor 1.1.1.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   advertise local-labeled-route disable
  !
  address-family ipv4 labeled-unicast
  !
 !
 neighbor 10.3.40.40
  remote-as 40
  egress-engineering
  address-family ipv4 unicast
   route-policy pass_all in
   route-policy pass_all out
  !
 !
 neighbor 10.3.50.50
  remote-as 50
  egress-engineering
  address-family ipv4 unicast
   route-policy pass_all in
   route-policy pass_all out
  !
```

```
  !
!
```

Egress Border Router R4 Configuration

The configuration of egress border router R4 follows the configuration of R3:

```
RP/0/RP0/CPU0:R4(config)# segment-routing
RP/0/RP0/CPU0:R4(config-sr)# global-block 16000 23999
RP/0/RP0/CPU0:R4(config-sr)# exit

RP/0/RP0/CPU0:R4(config)# interface Loopback0
RP/0/RP0/CPU0:R4(config-if)# ipv4 address 1.1.1.4 255.255.255.255
RP/0/RP0/CPU0:R4(config-if)# exit

RP/0/RP0/CPU0:R4(config)# mpls static
RP/0/RP0/CPU0:R4(config-mpls-static)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:R4(config-mpls-static)# exit

RP/0/RP0/CPU0:R4(config)# router isis 1
RP/0/RP0/CPU0:R4(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-isis-af)# segment-routing mpls
RP/0/RP0/CPU0:R4(config-isis-af)# metric-style wide
RP/0/RP0/CPU0:R4(config-isis-af)# exit

RP/0/RP0/CPU0:R4(config-isis)# interface Loopback0
RP/0/RP0/CPU0:R4(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-isis-if-af)# prefix-sid absolute 16004
RP/0/RP0/CPU0:R4(config-isis-if-af)# exit
RP/0/RP0/CPU0:R4(config-isis-if)# exit

RP/0/RP0/CPU0:R4(config-isis)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:R4(config-isis-if)# point-to-point
RP/0/RP0/CPU0:R4(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-isis-if-af)# exit
RP/0/RP0/CPU0:R4(config-isis-if)# exit
RP/0/RP0/CPU0:R4(config-isis)# exit

RP/0/RP0/CPU0:R4(config)# route-policy rpl_advertise_all_paths
RP/0/RP0/CPU0:R4(config-rpl)# set path-selection all advertise
RP/0/RP0/CPU0:R4(config-rpl)# set path-selection backup 1 install multipath-protect
RP/0/RP0/CPU0:R4(config-rpl)# end-policy

RP/0/RP0/CPU0:R4(config)# router bgp 100
RP/0/RP0/CPU0:R4(config-bgp)# bgp router-id 1.1.1.4
RP/0/RP0/CPU0:R4(config-bgp)# ibgp policy out enforce-modifications
RP/0/RP0/CPU0:R4(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-bgp-af)# advertise epe-bgp labeled-unicast
RP/0/RP0/CPU0:R4(config-bgp-af)# additional-paths receive
RP/0/RP0/CPU0:R4(config-bgp-af)# additional-paths send
RP/0/RP0/CPU0:R4(config-bgp-af)# additional-paths selection route-policy
rpl_advertise_all_paths
RP/0/RP0/CPU0:R4(config-bgp-af)# allocate-label all
RP/0/RP0/CPU0:R4(config-bgp-af)# exit

RP/0/RP0/CPU0:R4(config-bgp)# neighbor 1.1.1.1
RP/0/RP0/CPU0:R4(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:R4(config-bgp-nbr)# update-source Loopback0
RP/0/RP0/CPU0:R4(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# advertise local-labeled-route disable
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R4(config-bgp-nbr)# address-family ipv4 labeled-unicast
```

```
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R4(config-bgp-nbr)# exit

RP/0/RP0/CPU0:R4(config-bgp)# neighbor 10.4.40.40
RP/0/RP0/CPU0:R4(config-bgp-nbr)# remote-as 40
RP/0/RP0/CPU0:R4(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:R4(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# route-policy pass_all in
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# route-policy pass_all out
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R4(config-bgp-nbr)# exit

RP/0/RP0/CPU0:R4(config-bgp)# neighbor 10.4.50.50
RP/0/RP0/CPU0:R4(config-bgp-nbr)# remote-as 50
RP/0/RP0/CPU0:R4(config-bgp-nbr)# egress-engineering
RP/0/RP0/CPU0:R4(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# route-policy pass_all in
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# route-policy pass_all out
RP/0/RP0/CPU0:R4(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R4(config-bgp-nbr)# exit
RP/0/RP0/CPU0:R4(config-bgp)# exit
RP/0/RP0/CPU0:R4(config)# commit
```

Egress Border Router R4 Running Configuration

```
segment-routing
 global-block 16000 23999

interface Loopback0
 ipv4 address 1.1.1.4 255.255.255.255

mpls static
 interface GigabitEthernet0/0/0/0
!

router isis 1
 is-type level-2-only
 net 47.0000.0000.0004.00
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !

 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16004
  !
 !
 interface HundredGigE0/0/0/0
  point-to-point
  address-family ipv4 unicast
  !
 !

route-policy rpl_advertise_all_paths
  set path-selection all advertise
  set path-selection backup 1 install multipath-protect
end-policy
!

router bgp 100
 bgp router-id 1.1.1.4
 ibgp policy out enforce-modifications
```

```
 address-family ipv4 unicast
  advertise epe-bgp labeled-unicast
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy rpl_advertise_all_paths
  allocate-label all
 !
 neighbor 1.1.1.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   advertise local-labeled-route disable
  !
  address-family ipv4 labeled-unicast
  !
 !
 neighbor 10.4.40.40
  remote-as 40
  egress-engineering
  address-family ipv4 unicast
   route-policy pass_all in
   route-policy pass_all out
  !
 !
 neighbor 10.4.50.50
  remote-as 50
  egress-engineering
  address-family ipv4 unicast
   route-policy pass_all in
   route-policy pass_all out
  !
 !
!
```

Ingress Border Router R1 Configuration

Configure the SRGB:

```
RP/0/RP0/CPU0:R1(config)# segment-routing
RP/0/RP0/CPU0:R1(config-sr)# global-block 16000 23999
RP/0/RP0/CPU0:R1(config-sr)# exit
RP/0/RP0/CPU0:R1(config)#
```

Configure the Loopback addresses. Lo0 is advertised in IS-IS and used a BGP next-hop. Lo100 is advertised in BGP as an overlay prefix:

```
RP/0/RP0/CPU0:R1(config)# interface Loopback0
RP/0/RP0/CPU0:R1(config-if)# ipv4 address 1.1.1.1 255.255.255.255
RP/0/RP0/CPU0:R1(config-if)# exit

RP/0/RP0/CPU0:R1(config)# interface Loopback100
RP/0/RP0/CPU0:R1(config-if)# ipv4 address 151.1.1.1 255.255.255.255
RP/0/RP0/CPU0:R1(config-if)# exit
```

Enable SR MPLS under IS-IS:

```
RP/0/RP0/CPU0:R1(config)# router isis 1
RP/0/RP0/CPU0:R1(config-isis)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-isis-af)# metric-style wide
RP/0/RP0/CPU0:R1(config-isis-af)# segment-routing mpls
RP/0/RP0/CPU0:R1(config-isis-af)# exit
```

Configure prefix segment identifier (SID) value on the IS-IS enabled Loopback interface:

```
RP/0/RP0/CPU0:R1(config-isis)# interface Loopback0 address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-isis-if-af)# prefix-sid absolute 16001
RP/0/RP0/CPU0:R1(config-isis-if-af)# exit
RP/0/RP0/CPU0:R1(config-isis-if)# exit
```

Enable IS-IS in core-facing interface:

```
RP/0/RP0/CPU0:R1(config-isis)# interface HundredGigE0/0/0/0
RP/0/RP0/CPU0:R1(config-isis-if)# point-to-point
RP/0/RP0/CPU0:R1(config-isis-if)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-isis-if-af)# exit
RP/0/RP0/CPU0:R1(config-isis-if)# exit
RP/0/RP0/CPU0:R1(config-isis)# exit
```

Configure an RPL policy to prevent allocation of local label to overlay prefixes; such as Lo100 151.1.1.1/32:

```
RP/0/RP0/CPU0:R1(config)# prefix-set unlabelled_prefixes
RP/0/RP0/CPU0:R1(config-pfx)# 151.1.1.1/32
RP/0/RP0/CPU0:R1(config-pfx)# end-set
RP/0/RP0/CPU0:R1(config)# route-policy rpl_allocate_label
RP/0/RP0/CPU0:R1(config-rpl)# if destination in unlabelled_prefixes then
RP/0/RP0/CPU0:R1(config-rpl-if)# drop
RP/0/RP0/CPU0:R1(config-rpl-if)# else
RP/0/RP0/CPU0:R1(config-rpl-else)# pass
RP/0/RP0/CPU0:R1(config-rpl-else)# endif
RP/0/RP0/CPU0:R1(config-rpl)# end-policy
RP/0/RP0/CPU0:R1(config)#
```

Configure an RPL policy to influence the best-path selection by assigning a higher BGP local preference to the desired path. In this example, the desired egress exit path for prefix 161.1.1.0/28 is via R4 and then AS 50, and for prefix 161.1.1.1/32 is via R4 and then AS 40. Otherwise, the uninfluenced exit path for these prefixes is via R3:

```
RP/0/RP0/CPU0:R1(config)# route-policy rpl_epe
RP/0/RP0/CPU0:R1(config-rpl)# if destination in (161.1.1.0/28) and next-hop in (10.4.50.50)
 then
RP/0/RP0/CPU0:R1(config-rpl-if)# set local-preference 1000
RP/0/RP0/CPU0:R1(config-rpl-if)# elseif destination in (161.1.1.1/32) and next-hop in
(10.4.40.40) then
RP/0/RP0/CPU0:R1(config-rpl-elseif)# set local-preference 1000
RP/0/RP0/CPU0:R1(config-rpl-elseif)# endif
RP/0/RP0/CPU0:R1(config-rpl)# pass
RP/0/RP0/CPU0:R1(config-rpl)# end-policy
RP/0/RP0/CPU0:R1(config)#
```

Configure an RPL policy to advertise all candidate paths:

```
RP/0/RP0/CPU0:R1(config)# route-policy rpl_advertise_all_paths
RP/0/RP0/CPU0:R1(config-rpl)# set path-selection all advertise
RP/0/RP0/CPU0:R1(config-rpl)# end-policy
```

```
RP/0/RP0/CPU0:R1(config)# router bgp 100
RP/0/RP0/CPU0:R1(config-bgp)# bgp router-id 1.1.1.1
RP/0/RP0/CPU0:R1(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-af)# additional-paths receive
RP/0/RP0/CPU0:R1(config-bgp-af)# additional-paths send
RP/0/RP0/CPU0:R1(config-bgp-af)# additional-paths selection route-policy
rpl_advertise_all_paths
```

```
RP/0/RP0/CPU0:R1(config-bgp-af)# exit

RP/0/RP0/CPU0:R1(config-bgp)# neighbor 1.1.1.3
RP/0/RP0/CPU0:R1(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:R1(config-bgp-nbr)# update-source Loopback0
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# advertise local-labeled-route disable
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 labeled-unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R1(config-bgp-nbr)# exit

RP/0/RP0/CPU0:R1(config-bgp)# neighbor 1.1.1.4
RP/0/RP0/CPU0:R1(config-bgp-nbr)# remote-as 100
RP/0/RP0/CPU0:R1(config-bgp-nbr)# update-source Loopback0
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# advertise local-labeled-route disable
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 labeled-unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R1(config-bgp-nbr)# exit
RP/0/RP0/CPU0:R1(config-bgp)# exit
RP/0/RP0/CPU0:R1(config)# commit
```

Ingress Border Router R1 Running Configuration

```
segment-routing
 global-block 16000 23999

interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
interface Loopback100
 ipv4 address 151.1.1.1 255.255.255.255

router isis 1
 is-type level-2-only
 net 47.0000.0000.0001.00
 address-family ipv4 unicast
  metric-style wide
  segment-routing mpls
 !
 interface Loopback0
  address-family ipv4 unicast
   prefix-sid absolute 16001
  !
 !
 interface HundredGigE0/0/0/0
  point-to-point
  address-family ipv4 unicast
  !
 !
!

prefix-set unlabelled_prefixes
  151.1.1.1/32
end-set
!

route-policy rpl_allocate_label
  if destination in unlabelled_prefixes then
    drop
```

```
  else
    pass
  endif
end-policy
!

route-policy rpl_epe
  if destination in (161.1.1.0/28) and next-hop in (10.4.50.50) then
    set local-preference 1000
  elseif destination in (161.1.1.1/32) and next-hop in (10.4.40.40) then
    set local-preference 1000
  endif
  pass
end-policy
!

route-policy rpl_advertise_all_paths
  set path-selection all advertise
end-policy
!

router bgp 100
 bgp router-id 1.1.1.1
 ibgp policy out enforce-modifications
 address-family ipv4 unicast
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy rpl_advertise_all_paths
  network 151.1.1.1/32
  allocate-label route-policy rpl_allocate_label
 !
 neighbor 1.1.1.3
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   advertise local-labeled-route disable
  !
  address-family ipv4 labeled-unicast
  !
 !
 neighbor 1.1.1.4
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
   advertise local-labeled-route disable
  !
  address-family ipv4 labeled-unicast
  !
 !
!
```

The following sections depict the **show** command outputs associated with the Egress Border routers (R3, R4) and Ingress PE router (R1):

Egress Border Router R3 Output

The following commands show the BGP EPE labels allocated for eBGP neighbors 10.3.40.40 and 10.3.50.50 alongside their corresponding entries in the FIB:

```
RP/0/RP0/CPU0:R3# show bgp egress-engineering

 Egress Engineering Object: 10.3.40.40/32 (0x7fc163c62e80)
       EPE Type: Peer
```

```
         Nexthop: 10.3.40.40
         Version: 2, rn_version: 2
           Flags: 0x00000006
       Local ASN: 100
      Remote ASN: 40
       Local RID: 1.1.1.3
      Remote RID: 1.1.1.40
   Local Address: 10.3.40.3
       First Hop: 10.3.40.40
            NHID: 0
             IFH: 0x198
           Label: 24004, Refcount: 4
         rpc_set: 0x7fc14410ff18, ID: 1

  Egress Engineering Object: 10.3.50.50/32 (0x7fc163c62d88)
        EPE Type: Peer
         Nexthop: 10.3.50.50
         Version: 3, rn_version: 3
           Flags: 0x00000006
       Local ASN: 100
      Remote ASN: 50
       Local RID: 1.1.1.3
      Remote RID: 1.1.1.50
   Local Address: 10.3.50.3
       First Hop: 10.3.50.50
            NHID: 0
             IFH: 0x1a0
           Label: 24005, Refcount: 4
         rpc_set: 0x7fc144110088, ID: 2

RP/0/RP0/CPU0:R3# show mpls forwarding labels 24004
Thu Feb  3 22:11:18.459 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         No ID              Hu0/0/0/1    10.3.40.40      0

RP/0/RP0/CPU0:R3# show mpls forwarding labels 24005
Thu Feb  3 22:11:35.399 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24005  Pop         No ID              Hu0/0/0/2    10.3.50.50      0
```

The following output displays the BGP-LU prefixes used to advertise the EPE-enabled eBGP neighbors 10.3.40.40 and 10.3.50.50:

```
RP/0/RP0/CPU0:R3# show bgp ipv4 labeled-unicast
Thu Feb  3 22:11:57.865 UTC
BGP router identifier 1.1.1.3, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 4
BGP main routing table version 4
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.3.40.40/32      0.0.0.0                                0 i
*> 10.3.50.50/32      0.0.0.0                                0 i

Processed 2 prefixes, 2 paths
```

The details of the BGP-LU prefixes can be found below. Note that the EPE label is advertised in BGP-LU.

```
RP/0/RP0/CPU0:R3# show bgp ipv4 labeled-unicast 10.3.40.40/32
Thu Feb  3 22:12:18.210 UTC
BGP routing table entry for 10.3.40.40/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  3           3
    Local Label: 24004
Last Modified: Feb  3 19:13:07.039 for 02:59:11
Paths: (1 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Local
    0.0.0.0 from 0.0.0.0 (1.1.1.3)
      Origin IGP, localpref 100, valid, extranet, best, group-best
      Received Path ID 0, Local Path ID 1, version 3
      Origin-AS validity: not-found

RP/0/RP0/CPU0:R3# show bgp ipv4 labeled-unicast 10.3.50.50/32
Thu Feb  3 22:12:27.282 UTC
BGP routing table entry for 10.3.50.50/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  4           4
    Local Label: 24005
Last Modified: Feb  3 19:13:07.039 for 02:59:20
Paths: (1 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Local
    0.0.0.0 from 0.0.0.0 (1.1.1.3)
      Origin IGP, localpref 100, valid, extranet, best, group-best
      Received Path ID 0, Local Path ID 1, version 4
      Origin-AS validity: not-found
```

The output below depicts the BGP route and CEF details for an overlay prefix (161.1.1.0/28) learned via the EPE-enabled BGP neighbors:

```
RP/0/RP0/CPU0:R3# show bgp ipv4 unicast
Thu Feb  3 22:58:01.736 UTC
BGP router identifier 1.1.1.3, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 14
BGP main routing table version 14
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
```

```
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.3.40.40/32      0.0.0.0                                0 i
*> 10.3.50.50/32      0.0.0.0                                0 i
*>i151.1.1.1/32       1.1.1.1                  0    100      0 i
*> 161.1.1.0/28       10.3.40.40                             0 40 60 i
*                     10.3.50.50                             0 50 60 i

Processed 4 prefixes, 5 paths


RP/0/RP0/CPU0:R3# show bgp ipv4 unicast 161.1.1.0/28
Thu Feb  3 22:31:52.893 UTC
BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  6           6
Last Modified: Feb  3 22:28:56.039 for 00:02:56
Paths: (2 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  40 60
    10.3.40.40 from 10.3.40.40 (1.1.1.40)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 5
      Origin-AS validity: (disabled)
  Path #2: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  50 60
    10.3.50.50 from 10.3.50.50 (1.1.1.50)
      Origin IGP, localpref 100, valid, external, group-best, backup, add-path
      Received Path ID 0, Local Path ID 2, version 6
      Origin-AS validity: (disabled)


RP/0/RP0/CPU0:R3# show cef ipv4 161.1.1.0/28
Thu Feb 10 20:17:29.240 UTC
161.1.1.0/28, version 24, internal 0x5000001 0x40 (ptr 0x90684920) [1], 0x0 (0x0), 0x0 (0x0)

 Updated Feb 10 17:37:16.609
 Prefix Len 28, traffic index 0, precedence n/a, priority 4
   via 10.3.40.40/32, 5 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x90684c08 0x0], Internal 0x90211730
    next hop 10.3.40.40/32 via 10.3.40.40/32
   via 10.3.50.50/32, 4 dependencies, recursive, bgp-ext, backup [flags 0x6120]
    path-idx 1 NHID 0x0 [0x90685040 0x0]
    next hop 10.3.50.50/32 via 10.3.50.50/32
```

Egress Border Router R4 Output

The following outputs correspond to egress border router R4. They follow the same sequence shown for router R3.

```
RP/0/RP0/CPU0:R4# show bgp egress-engineering

Egress Engineering Object: 10.4.40.40/32 (0x7f84d2a4ae80)
       EPE Type: Peer
        Nexthop: 10.4.40.40
        Version: 2, rn_version: 2
          Flags: 0x00000006
      Local ASN: 100
     Remote ASN: 40
      Local RID: 1.1.1.4
     Remote RID: 1.1.1.40
  Local Address: 10.4.40.4
      First Hop: 10.4.40.40
           NHID: 0
            IFH: 0x198
          Label: 24004, Refcount: 4
        rpc_set: 0x7f84b010fdb8, ID: 1

 Egress Engineering Object: 10.4.50.50/32 (0x7f84d2a4ad88)
       EPE Type: Peer
        Nexthop: 10.4.50.50
        Version: 3, rn_version: 3
          Flags: 0x00000006
      Local ASN: 100
     Remote ASN: 50
      Local RID: 1.1.1.4
     Remote RID: 1.1.1.50
  Local Address: 10.4.50.4
      First Hop: 10.4.50.50
           NHID: 0
            IFH: 0x1a0
          Label: 24005, Refcount: 4
        rpc_set: 0x7f84b010ff28, ID: 2

RP/0/RP0/CPU0:R4# show mpls forwarding labels 24004
Thu Feb  3 22:34:55.059 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         No ID              Hu0/0/0/1    10.4.40.40      0

RP/0/RP0/CPU0:R4# show mpls forwarding labels 24005
Thu Feb  3 22:35:07.252 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24005  Pop         No ID              Hu0/0/0/2    10.4.50.50      0

RP/0/RP0/CPU0:R4# show bgp ipv4 labeled-unicast
Thu Feb  3 22:59:37.978 UTC
BGP router identifier 1.1.1.4, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 14
BGP main routing table version 14
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.4.40.40/32      0.0.0.0                                0 i
*> 10.4.50.50/32      0.0.0.0                                0 i

Processed 2 prefixes, 2 paths


RP/0/RP0/CPU0:R4# show bgp ipv4 labeled-unicast 10.4.40.40/32
Thu Feb  3 22:35:41.275 UTC
BGP routing table entry for 10.4.40.40/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  3           3
    Local Label: 24004
Last Modified: Feb  3 19:13:08.143 for 03:22:33
Paths: (1 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Labeled-unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Labeled-unicast paths to peers (in unique update groups):
    1.1.1.1
  Local
    0.0.0.0 from 0.0.0.0 (1.1.1.4)
      Origin IGP, localpref 100, valid, extranet, best, group-best
      Received Path ID 0, Local Path ID 1, version 3
      Origin-AS validity: not-found


RP/0/RP0/CPU0:R4# show bgp ipv4 labeled-unicast 10.4.50.50/32
Thu Feb  3 22:35:53.259 UTC
BGP routing table entry for 10.4.50.50/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  4           4
    Local Label: 24005
Last Modified: Feb  3 19:13:08.143 for 03:22:45
Paths: (1 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Labeled-unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Labeled-unicast paths to peers (in unique update groups):
    1.1.1.1
  Local
    0.0.0.0 from 0.0.0.0 (1.1.1.4)
      Origin IGP, localpref 100, valid, extranet, best, group-best
      Received Path ID 0, Local Path ID 1, version 4
      Origin-AS validity: not-found


RP/0/RP0/CPU0:R4# show bgp ipv4 unicast
Thu Feb  3 23:00:32.470 UTC
BGP router identifier 1.1.1.4, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 14
```

```
BGP main routing table version 14
BGP NSR Initial initsync version 2 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*> 10.4.40.40/32      0.0.0.0                                0 i
*> 10.4.50.50/32      0.0.0.0                                0 i
*>i151.1.1.1/32       1.1.1.1                  0    100      0 i
*> 161.1.1.0/28       10.4.40.40                             0 40 60 i
*                     10.4.50.50                             0 50 60 i

Processed 4 prefixes, 5 paths

RP/0/RP0/CPU0:R4# show bgp ipv4 unicast 161.1.1.0/28
Thu Feb  3 22:36:09.266 UTC
BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  6           6
Last Modified: Feb  3 22:28:56.143 for 00:07:13
Paths: (2 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  40 60
    10.4.40.40 from 10.4.40.40 (1.1.1.40)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 5
      Origin-AS validity: (disabled)
  Path #2: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  50 60
    10.4.50.50 from 10.4.50.50 (1.1.1.50)
      Origin IGP, localpref 100, valid, external, group-best, backup, add-path
      Received Path ID 0, Local Path ID 2, version 6
      Origin-AS validity: (disabled)
```

Ingress Border Router R1 Output

This section includes the outputs corresponding to ingress border router R1.

R1 learns the eBGP neighbor IP addresses via BGP-LU. In the details for each neighbor prefix, observe that the advertised BGP-LU label corresponds to the EPE label at the egress border router (R3 or R4).

```
RP/0/RP0/CPU0:R1# show bgp ipv4 labeled-unicast
Thu Feb 10 20:18:59.645 UTC
BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 8
BGP main routing table version 8
```

```
BGP NSR Initial initsync version 5 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*>i10.3.40.40/32      1.1.1.3                       100      0 i
*>i10.3.50.50/32      1.1.1.3                       100      0 i
*>i10.4.40.40/32      1.1.1.4                       100      0 i
*>i10.4.50.50/32      1.1.1.4                       100      0 i

Processed 4 prefixes, 4 paths


RP/0/RP0/CPU0:R1# show bgp ipv4 labeled-unicast 10.3.40.40/32
Thu Feb  3 23:01:57.912 UTC
BGP routing table entry for 10.3.40.40/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 15          15
    Local Label: 24004
Last Modified: Feb  3 22:47:43.539 for 00:14:14
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    1.1.1.3 (metric 30) from 1.1.1.3 (1.1.1.3)
      Received Label 24004
      Origin IGP, localpref 100, valid, internal, best, group-best, labeled-unicast
      Received Path ID 1, Local Path ID 1, version 15


RP/0/RP0/CPU0:R1# show bgp ipv4 labeled-unicast 10.3.50.50/32
Thu Feb  3 23:02:09.173 UTC
BGP routing table entry for 10.3.50.50/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 16          16
    Local Label: 24005
Last Modified: Feb  3 22:47:43.539 for 00:14:25
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    1.1.1.3 (metric 30) from 1.1.1.3 (1.1.1.3)
      Received Label 24005
      Origin IGP, localpref 100, valid, internal, best, group-best, labeled-unicast
      Received Path ID 1, Local Path ID 1, version 16


RP/0/RP0/CPU0:R1# show bgp ipv4 labeled-unicast 10.4.40.40/32
Thu Feb  3 23:02:18.843 UTC
BGP routing table entry for 10.4.40.40/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 17          17
    Local Label: 24006
Last Modified: Feb  3 22:47:43.539 for 00:14:35
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
```

```
  Not advertised to any peer
  Local
    1.1.1.4 (metric 30) from 1.1.1.4 (1.1.1.4)
      Received Label 24004
      Origin IGP, localpref 100, valid, internal, best, group-best, labeled-unicast
      Received Path ID 1, Local Path ID 1, version 17

RP/0/RP0/CPU0:R1# show bgp ipv4 labeled-unicast 10.4.50.50/32
Thu Feb  3 23:02:27.622 UTC
BGP routing table entry for 10.4.50.50/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 18          18
    Local Label: 24007
Last Modified: Feb  3 22:47:43.539 for 00:14:44
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local
    1.1.1.4 (metric 30) from 1.1.1.4 (1.1.1.4)
      Received Label 24005
      Origin IGP, localpref 100, valid, internal, best, group-best, labeled-unicast
      Received Path ID 1, Local Path ID 1, version 18

RP/0/RP0/CPU0:R1# show isis segment-routing label table

IS-IS 1 IS Label Table
Label         Prefix                   Interface
----------    ----------------         ---------
16001         1.1.1.1/32               Loopback0
16002         1.1.1.2/32
16003         1.1.1.3/32
16004         1.1.1.4/32
```

The following show commands depict the RIB and CEF outputs for the loopbacks of R3 and R4 learned vis ISIS-SR:

```
RP/0/RP0/CPU0:R1# show route 1.1.1.3/32

Routing entry for 1.1.1.3/32
  Known via "isis 1", distance 115, metric 30, labeled SR, type level-2
  Installed Feb 10 17:36:12.497 for 02:43:40
  Routing Descriptor Blocks
    10.1.2.2, from 1.1.1.3, via HundredGigE0/0/0/0
      Route metric is 30
  No advertising protos.

RP/0/RP0/CPU0:R1# show route 1.1.1.4/32

Routing entry for 1.1.1.4/32
  Known via "isis 1", distance 115, metric 30, labeled SR, type level-2
  Installed Feb 10 17:37:02.171 for 02:42:59
  Routing Descriptor Blocks
    10.1.2.2, from 1.1.1.4, via HundredGigE0/0/0/0
      Route metric is 30
  No advertising protos.

RP/0/RP0/CPU0:R1# show cef 1.1.1.3/32
```

```
1.1.1.3/32, version 18, labeled SR, internal 0x1000001 0x8110 (ptr 0x90cd33a0) [1], 0x0
(0x90c3eb10), 0xa28 (0x91a18378)
 Updated Feb 10 17:36:12.506
 local adjacency to HundredGigE0/0/0/0

 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 10.1.2.2/32, HundredGigE0/0/0/0, 7 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x91de84d8 0x0]
    next hop 10.1.2.2/32
    local adjacency
     local label 16003      labels imposed {16003}

RP/0/RP0/CPU0:R1# show cef 1.1.1.4/32

1.1.1.4/32, version 20, labeled SR, internal 0x1000001 0x8110 (ptr 0x90cd32c8) [1], 0x0
(0x90c3eb58), 0xa28 (0x91a18408)
 Updated Feb 10 17:37:02.176
 local adjacency to HundredGigE0/0/0/0

 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 10.1.2.2/32, HundredGigE0/0/0/0, 7 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x91de84d8 0x0]
    next hop 10.1.2.2/32
    local adjacency
     local label 16004      labels imposed {16004}
```

Next, we observe the BGP table for overlay prefixes at R1. In this usecase, we use prefix 161.1.1.0/28 as an overlay prefix learned from AS 40 and AS 50. Note that all BGP paths are present at R1 with a BGP next-hop unchanged. By default and without any BGP policy applied, the BGP best-path is the path from NH 10.3.40.40 (AS 40 via R3).

```
RP/0/RP0/CPU0:R1# show bgp

BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 21
BGP main routing table version 21
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*>i10.3.40.40/32      1.1.1.3                        100      0 i
*>i10.3.50.50/32      1.1.1.3                        100      0 i
*>i10.4.40.40/32      1.1.1.4                        100      0 i
*>i10.4.50.50/32      1.1.1.4                        100      0 i
*> 151.1.1.1/32       0.0.0.0                  0         32768 i
*>i161.1.1.0/28       10.3.40.40                     100      0 40 60 i
* i                   10.3.50.50                     100      0 50 60 i
* i                   10.4.40.40                     100      0 40 60 i
* i                   10.4.50.50                     100      0 50 60 i

Processed 6 prefixes, 9 paths
```

```
RP/0/RP0/CPU0:R1# show bgp ipv4 unicast 161.1.1.0/28

BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  8           8
Last Modified: Feb 10 17:38:09.280 for 02:42:57
Paths: (4 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  40 60
    10.3.40.40 (metric 30) from 1.1.1.3 (1.1.1.3)
      Origin IGP, localpref 100, valid, internal, best, group-best
      Received Path ID 1, Local Path ID 1, version 8
  Path #2: Received by speaker 0
  Not advertised to any peer
  50 60
    10.3.50.50 (metric 30) from 1.1.1.3 (1.1.1.3)
      Origin IGP, localpref 100, valid, internal, group-best, add-path
      Received Path ID 2, Local Path ID 4, version 8
  Path #3: Received by speaker 0
  Not advertised to any peer
  40 60
    10.4.40.40 (metric 30) from 1.1.1.4 (1.1.1.4)
      Origin IGP, localpref 100, valid, internal, add-path
      Received Path ID 1, Local Path ID 2, version 8
  Path #4: Received by speaker 0
  Not advertised to any peer
  50 60
    10.4.50.50 (metric 30) from 1.1.1.4 (1.1.1.4)
      Origin IGP, localpref 100, valid, internal, add-path
      Received Path ID 2, Local Path ID 3, version 8
```

A ping and traceroute to the overlay prefix confirms that the traffic is directed to R3 (prefix SID 16003) and then to AS 40 (EPE label 24004 for the eBGP neighbor to AS 40 at R3).

```
RP/0/RP0/CPU0:R1# ping 161.1.1.1 source 151.1.1.1 count 10
Thu Feb  3 23:20:48.911 UTC
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 161.1.1.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 30/36/54 ms


RP/0/RP0/CPU0:R1# traceroute 161.1.1.1 source 151.1.1.1
Thu Feb  3 23:20:53.630 UTC

Type escape sequence to abort.
Tracing the route to 161.1.1.1

 1  10.1.2.2 [MPLS: Labels 16003/24004 Exp 0] 49 msec  45 msec  42 msec
 2  10.2.3.3 [MPLS: Label 24004 Exp 0] 42 msec  37 msec  37 msec
 3  10.3.40.40 44 msec  37 msec  41 msec
 4  10.40.60.60 47 msec *  55 msec
```

Now, we proceed to apply a BGP route-policy that would modify BGP best-path selection and choose instead the path from NH 10.4.50.50 (AS 50 via R4).

```
RP/0/RP0/CPU0:R1(config)# route-policy rpl_epe
RP/0/RP0/CPU0:R1(config-rpl)# if destination in (161.1.1.0/28) and next-hop in (10.4.50.50)
 then
RP/0/RP0/CPU0:R1(config-rpl-if)# set local-preference 1000
```

```
RP/0/RP0/CPU0:R1(config-rpl-if)# elseif destination in (161.1.1.1/32) and next-hop in
(10.4.40.40) then
RP/0/RP0/CPU0:R1(config-rpl-elseif)# set local-preference 1000
RP/0/RP0/CPU0:R1(config-rpl-elseif)# endif
RP/0/RP0/CPU0:R1(config-rpl)# pass
RP/0/RP0/CPU0:R1(config-rpl)# end-policy
RP/0/RP0/CPU0:R1(config)#

RP/0/RP0/CPU0:R1(config)# router bgp 100
RP/0/RP0/CPU0:R1(config-bgp)# neighbor 1.1.1.3
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# route-policy rpl_epe in
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# exit
RP/0/RP0/CPU0:R1(config-bgp-nbr)# exit
RP/0/RP0/CPU0:R1(config-bgp)# neighbor 1.1.1.4
RP/0/RP0/CPU0:R1(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)# route-policy rpl_epe in
RP/0/RP0/CPU0:R1(config-bgp-nbr-af)#
```

Observe the new BGP best-path selected for the overlay prefix via NH 10.4.50.50:

```
RP/0/RP0/CPU0:R1# show bgp

BGP router identifier 1.1.1.1, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 20
BGP main routing table version 20
BGP NSR Initial initsync version 7 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
*>i10.3.40.40/32      1.1.1.3                       100      0 i
*>i10.3.50.50/32      1.1.1.3                       100      0 i
*>i10.4.40.40/32      1.1.1.4                       100      0 i
*>i10.4.50.50/32      1.1.1.4                       100      0 i
*> 151.1.1.1/32       0.0.0.0                  0         32768 i
* i161.1.1.0/28       10.3.40.40                    100      0 40 60 i
* i                   10.3.50.50                    100      0 50 60 i
* i                   10.4.40.40                    100      0 40 60 i
*>i                   10.4.50.50                   1000      0 50 60 i

Processed 6 prefixes, 9 paths


RP/0/RP0/CPU0:R1# show bgp ipv4 unicast 161.1.1.0/28

BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 20          20
Last Modified: Feb  3 23:13:30.539 for 00:01:33
Paths: (4 available, best #4)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  40 60
    10.3.40.40 (metric 30) from 1.1.1.3 (1.1.1.3)
```

```
      Origin IGP, localpref 100, valid, internal, group-best, add-path
      Received Path ID 1, Local Path ID 4, version 20
  Path #2: Received by speaker 0
  Not advertised to any peer
  50 60
    10.3.50.50 (metric 30) from 1.1.1.3 (1.1.1.3)
      Origin IGP, localpref 100, valid, internal, add-path
      Received Path ID 2, Local Path ID 3, version 8
  Path #3: Received by speaker 0
  Not advertised to any peer
  40 60
    10.4.40.40 (metric 30) from 1.1.1.4 (1.1.1.4)
      Origin IGP, localpref 100, valid, internal, add-path
      Received Path ID 1, Local Path ID 2, version 8
  Path #4: Received by speaker 0
  Not advertised to any peer
  50 60
    10.4.50.50 (metric 30) from 1.1.1.4 (1.1.1.4)
      Origin IGP, localpref 1000, valid, internal, best, group-best
      Received Path ID 2, Local Path ID 1, version 20
```

A ping and traceroute to the overlay prefix confirms that after the RPL policy is applied, the traffic is directed instead to R4 (prefix SID 16004) and then to AS 50 (EPE label 24005 for the eBGP neighbor to AS 50 at R4).

```
RP/0/RP0/CPU0:R1# ping 161.1.1.1 source 151.1.1.1 count 10
Thu Feb  3 23:17:43.812 UTC
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 161.1.1.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 30/35/50 ms


RP/0/RP0/CPU0:R1# traceroute 161.1.1.1 source 151.1.1.1
Thu Feb  3 23:18:01.656 UTC

Type escape sequence to abort.
Tracing the route to 161.1.1.1

 1  10.1.2.2 [MPLS: Labels 16004/24005 Exp 0] 50 msec  42 msec  45 msec
 2  10.2.4.4 [MPLS: Label 24005 Exp 0] 50 msec  42 msec  42 msec
 3  10.4.50.50 46 msec  44 msec  44 msec
 4  10.50.60.60 51 msec  *  54 msec
```

IP Lookup Fallback for BGP Peering (EPE) Segments

Table 4: Feature History Table

Feature Name	Release	Description
IP Lookup Fallback for BGP Peering (EPE) Segments	Release 7.3.3	BGP peering segments/SIDs are part of the Segment Routing Centralized BGP Egress Peer Engineering solution (BGP-EPE). A BGP-EPE-enabled border router allocates and programs BGP peering SIDs (EPE labels) to steer traffic over a specific external interface/BGP neighbor. This feature allows a BGP-EPE-enabled border router to pop the EPE label and forward traffic based on an IP-based lookup when a BGP neighbor fails. Traffic arriving with the EPE label assigned to a failed neighbor is forwarded based on a destination IP address lookup to allow traffic to be forwarded over a different directly connected external peer.

BGP peering segments/SIDs are part of the Segment Routing Centralized BGP Egress Peer Engineering solution (BGP-EPE), as described in IETF RFC 9087. A BGP-EPE-enabled border router allocates and programs BGP peering SIDs (EPE labels) to steer traffic over a specific external interface/BGP neighbor.

This feature allows a BGP-EPE-enabled border router to pop the EPE label and forward traffic based on an IP-based lookup when a BGP neighbor fails. Traffic arriving with the EPE label assigned to a failed neighbor is forwarded based on a destination IP address lookup to allow traffic to be forwarded over a different directly connected external peer.

Usage Guidelines and Limitations

The following usage guidelines and limitations apply for this feature:

- IP Lookup Fallback for BGP peering SIDs (EPE Peer-Node SIDs and Peer-Adjacencies SIDs) allocated dynamically or configured manually is supported.
- BGPv4 and BGPv6 EPE-enabled neighbors are supported
- Sub-second convergence is supported upon failure of EPE-enabled BGP neighbor with interface peering.
- Sub-second convergence is supported upon failure of EPE-enabled BGP neighbor with loopback peering over a single interface.
- Sub-second convergence is not supported upon failure of EPE-enabled BGP neighbor with loopback peering over more than one interface.
- IP Lookup Fallback for BGP Peer-Set SIDs is not supported

- MPLS egress path counters for BGP peering SIDs are not supported when IP Lookup Fallback is enabled

Enabling IP Lookup Fallback for BGP Peering (EPE) Segments

To guaranteed convergence, configure a route policy on the ingress border router to advertise all BGP paths. For example:

```
RP/0/RP0/CPU0:R1(config)# route-policy INSTALL_BACKUP
RP/0/RP0/CPU0:R1(config-rpl)# set path-selection all advertise
RP/0/RP0/CPU0:R1(config-rpl)# set path-selection backup 1 install multipath-protect
RP/0/RP0/CPU0:R1(config-rpl)# end-policy

RP/0/RP0/CPU0:R1(config)# router bgp 100
RP/0/RP0/CPU0:R1(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R1(config-bgp-af)# additional-paths selection route-policy INSTALL_BACKUP
RP/0/RP0/CPU0:R1(config-bgp-af)# exit
RP/0/RP0/CPU0:R1(config-bgp)# exit
RP/0/RP0/CPU0:R1(config)#
```

To enable IP lookup fallback for EPE segments, use the **epe backup enable** command in router BGP address family configuration mode.

To retain the local label of the primary path after reconvergence for the specified amount of time, use the **retain local-label** *minutes* command in router BGP address family configuration mode. The range of *minutes* is from 3 to 60.

The following example shows how to enable IP lookup fallback for EPE segments associated with BGPv4 EPE-enabled neighbors:

```
RP/0/RP0/CPU0:R3(config)# router bgp 100
RP/0/RP0/CPU0:R3(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:R3(config-bgp-af)# epe backup enable
RP/0/RP0/CPU0:R3(config-bgp-af)# retain local-label 6
```

Running Config

```
router bgp 100
  address-family ipv4 unicast
   epe backup enable
   retain local-label 6
```

Verification

The following outputs display the forwarding entries for the EPE MPLS labels (24004 and 24005) at an egress border router **before** the IP Lookup Fallback for EPE feature is enabled. Observe that no backup is programmed.

```
RP/0/RP0/CPU0:R3# show mpls forwarding labels 24004 24005

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         No ID              Hu0/0/0/1    10.3.40.40      0
24005  Pop         No ID              Hu0/0/0/2    10.3.50.50      0


RP/0/RP0/CPU0:R3# show mpls forwarding labels 24004 24005 detail
```

```
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Pop         No ID              Hu0/0/0/1    10.3.40.40      0
     Updated: Feb 10 18:38:16.116
     Path Flags: 0x6000 [  ]
     Version: 16, Priority: 3
     Label Stack (Top -> Bottom): { Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 0/0, MTU: 1500
     Outgoing Interface: HundredGigE0/0/0/1 (ifhandle 0x00000198)
     Packets Switched: 0

24005  Pop         No ID              Hu0/0/0/2    10.3.50.50      0
     Updated: Feb 10 18:38:16.116
     Path Flags: 0x6000 [  ]
     Version: 17, Priority: 3
     Label Stack (Top -> Bottom): { Imp-Null }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0
     MAC/Encaps: 0/0, MTU: 1500
     Outgoing Interface: HundredGigE0/0/0/2 (ifhandle 0x000001a0)
     Packets Switched: 0
```

The following output depicts the BGP table for an overlay prefix including its primary and backup path.

```
RP/0/RP0/CPU0:R3# show bgp ipv4 unicast 161.1.1.0/28

BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  6           6
Last Modified: Feb  3 22:28:56.039 for 00:02:56
Paths: (2 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  40 60
    10.3.40.40 from 10.3.40.40 (1.1.1.40)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 5
      Origin-AS validity: (disabled)
  Path #2: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  50 60
    10.3.50.50 from 10.3.50.50 (1.1.1.50)
      Origin IGP, localpref 100, valid, external, group-best, backup, add-path
      Received Path ID 0, Local Path ID 2, version 6
      Origin-AS validity: (disabled)


RP/0/RP0/CPU0:R3# show cef 161.1.1.0/28 detail

161.1.1.0/28, version 26, internal 0x5000001 0x40 (ptr 0x90cd2920) [1], 0x0 (0x0), 0x0 (0x0)

 Updated Feb 17 20:35:32.438
 Prefix Len 28, traffic index 0, precedence n/a, priority 4
  gateway array (0x90aa9a58) reference count 1, flags 0x102010, source rib (7), 0 backups
```

```
                [1 type 3 flags 0x48441 (0x90b5a148) ext 0x0 (0x0)]
   LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
   gateway array update type-time 1 Feb 17 20:35:32.438
  LDI Update time Feb 17 20:35:32.438

   Level 1 - Load distribution: 0
   [0] via 10.3.40.40/32, recursive

    via 10.3.40.40/32, 3 dependencies, recursive, bgp-ext [flags 0x6020]
     path-idx 0 NHID 0x0 [0x90cd3250 0x0], Internal 0x9081e550
     next hop 10.3.40.40/32 via 10.3.40.40/32

     Load distribution: 0 (refcount 1)

     Hash  OK  Interface                 Address
     0     Y   HundredGigE0/0/0/1        10.3.40.40

    via 10.3.50.50/32, 2 dependencies, recursive, bgp-ext, backup [flags 0x6120]
     path-idx 1 NHID 0x0 [0x90cd3178 0x0]
     next hop 10.3.50.50/32 via 10.3.50.50/32
```

The following outputs display the forwarding entries for the EPE MPLS labels at an egress border router **after** the IP Lookup Fallback for EPE feature is enabled. Observe that a backup path is now programmed for an EPE local label.

```
RP/0/RP0/CPU0:R3# show mpls forwarding labels 24004 24005
Thu Feb 10 18:40:34.655 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Unlabelled  No ID              Hu0/0/0/1    10.3.40.40      0
       Aggregate   No ID              default                      0            (!)
24005  Unlabelled  No ID              Hu0/0/0/2    10.3.50.50      0
       Aggregate   No ID              default                      0            (!)

RP/0/RP0/CPU0:R3# show mpls forwarding labels 24004 24005 detail

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24004  Unlabelled  No ID              Hu0/0/0/1    10.3.40.40      0
     Updated: Feb 10 18:40:25.476
     Path Flags: 0x6000 [  ]
     Version: 18, Priority: 3
     Label Stack (Top -> Bottom): { Unlabelled }
     NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 1, Weight: 0
     MAC/Encaps: 14/14, MTU: 1500
     Outgoing Interface: HundredGigE0/0/0/1 (ifhandle 0x00000198)
     Packets Switched: 0

       Aggregate   No ID              default                      0            (!)
     Updated: Feb 10 18:40:25.476
     Path Flags: 0x100 [  BKUP, NoFwd ]
     Label Stack (Top -> Bottom): { }
     MAC/Encaps: 0/0, MTU: 0
     Packets Switched: 0
24005  Unlabelled  No ID              Hu0/0/0/2    10.3.50.50      0
     Updated: Feb 10 18:40:25.482
     Path Flags: 0x6000 [  ]
     Version: 19, Priority: 3
     Label Stack (Top -> Bottom): { Unlabelled }
```

```
      NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 1, Weight: 0
      MAC/Encaps: 14/14, MTU: 1500
      Outgoing Interface: HundredGigE0/0/0/2 (ifhandle 0x000001a0)
      Packets Switched: 0

        Aggregate   No ID              default                      0             (!)
      Updated: Feb 10 18:40:25.482
      Path Flags: 0x100 [  BKUP, NoFwd ]
      Label Stack (Top -> Bottom): { }
      MAC/Encaps: 0/0, MTU: 0
      Packets Switched: 0


RP/0/RP0/CPU0:R3# show bgp ipv4 unicast 161.1.1.0/28
Thu Feb  3 22:31:52.893 UTC
BGP routing table entry for 161.1.1.0/28
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                  6           6
Last Modified: Feb  3 22:28:56.039 for 00:02:56
Paths: (2 available, best #1)
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  40 60
    10.3.40.40 from 10.3.40.40 (1.1.1.40)
      Origin IGP, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 5
      Origin-AS validity: (disabled)
  Path #2: Received by speaker 0
  Advertised IPv4 Unicast paths to peers (in unique update groups):
    1.1.1.1
  50 60
    10.3.50.50 from 10.3.50.50 (1.1.1.50)
      Origin IGP, localpref 100, valid, external, group-best, backup, add-path
      Received Path ID 0, Local Path ID 2, version 6
      Origin-AS validity: (disabled)


RP/0/RP0/CPU0:R3# show cef 161.1.1.0/28 detail

161.1.1.0/28, version 24, internal 0x5000001 0x40 (ptr 0x90684920) [1], 0x0 (0x0), 0x0 (0x0)

 Updated Feb 10 17:37:16.610
 Prefix Len 28, traffic index 0, precedence n/a, priority 4
  gateway array (0x9045ba58) reference count 1, flags 0x102010, source rib (7), 0 backups
               [1 type 3 flags 0x48441 (0x9050c148) ext 0x0 (0x0)]
  LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
  gateway array update type-time 1 Feb 10 17:37:16.610
 LDI Update time Feb 10 17:37:16.623

  Level 1 - Load distribution: 0
  [0] via 10.3.40.40/32, recursive

   via 10.3.40.40/32, 5 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x90684c08 0x0], Internal 0x90211730
    next hop 10.3.40.40/32 via 10.3.40.40/32

    Load distribution: 0 (refcount 1)
```

```
 Hash  OK  Interface                 Address
 0     Y   HundredGigE0/0/0/1        10.3.40.40

via 10.3.50.50/32, 4 dependencies, recursive, bgp-ext, backup [flags 0x6120]
 path-idx 1 NHID 0x0 [0x90685040 0x0]
 next hop 10.3.50.50/32 via 10.3.50.50/32
```

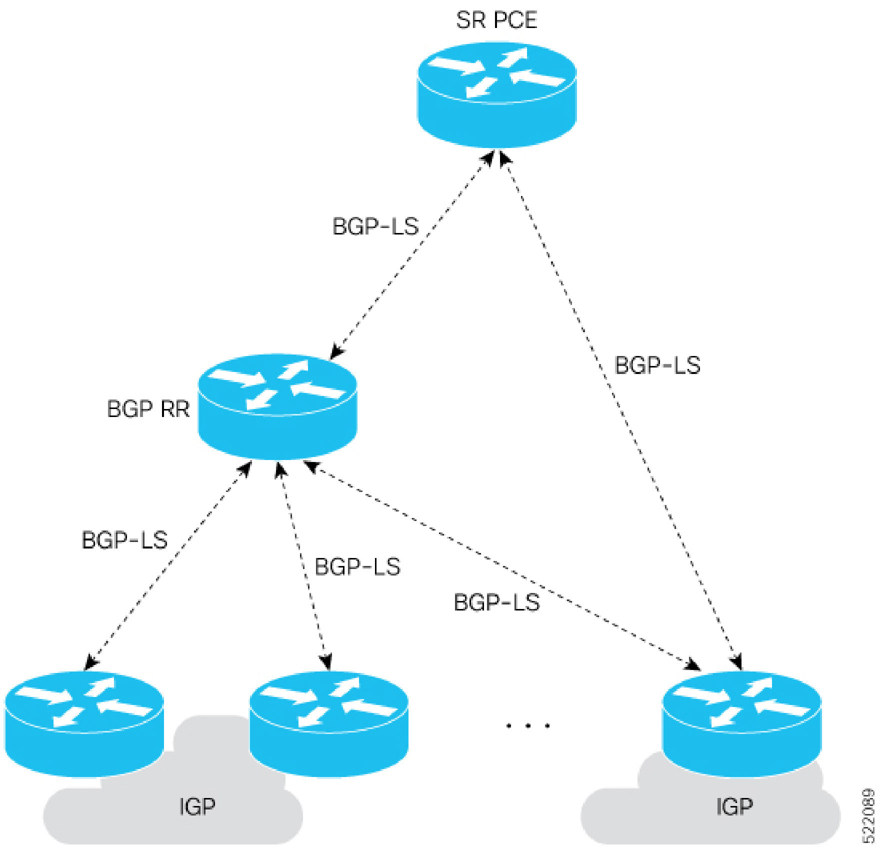
Configure BGP Link-State

BGP Link-State (LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) originally defined to carry interior gateway protocol (IGP) link-state information through BGP. The BGP Network Layer Reachability Information (NLRI) encoding format for BGP-LS and a new BGP Path Attribute called the BGP-LS attribute are defined in RFC7752. The identifying key of each Link-State object, namely a node, link, or prefix, is encoded in the NLRI and the properties of the object are encoded in the BGP-LS attribute.

The BGP-LS Extensions for Segment Routing are documented in RFC9085.

BGP-LS applications like an SR Path Computation Engine (SR-PCE) can learn the SR capabilities of the nodes in the topology and the mapping of SR segments to those nodes. This can enable the SR-PCE to perform path computations based on SR-TE and to steer traffic on paths different from the underlying IGP-based distributed best-path computation.

The following figure shows a typical deployment scenario. In each IGP area, one or more nodes (BGP speakers) are configured with BGP-LS. These BGP speakers form an iBGP mesh by connecting to one or more route-reflectors. This way, all BGP speakers (specifically the route-reflectors) obtain Link-State information from all IGP areas (and from other ASes from eBGP peers).



Usage Guidelines and Limitations

- BGP-LS supports IS-IS and OSPFv2.
- The identifier field of BGP-LS (referred to as the Instance-ID) identifies the IGP routing domain where the NLRI belongs. The NLRIs representing link-state objects (nodes, links, or prefixes) from the same IGP routing instance must use the same Instance-ID value.
- When there is only a single protocol instance in the network where BGP-LS is operational, we recommend configuring the Instance-ID value to **0**.
- Assign consistent BGP-LS Instance-ID values on all BGP-LS Producers within a given IGP domain.
- NLRIs with different Instance-ID values are considered to be from different IGP routing instances.
- Unique Instance-ID values must be assigned to routing protocol instances operating in different IGP domains. This allows the BGP-LS Consumer (for example, SR-PCE) to build an accurate segregated multi-domain topology based on the Instance-ID values, even when the topology is advertised via BGP-LS by multiple BGP-LS Producers in the network.
- If the BGP-LS Instance-ID configuration guidelines are not followed, a BGP-LS Consumer may see duplicate link-state objects for the same node, link, or prefix when there are multiple BGP-LS Producers deployed. This may also result in the BGP-LS Consumers getting an inaccurate network-wide topology.

- The following table defines the supported extensions to the BGP-LS address family for carrying IGP topology information (including SR information) via BGP. For more information on the BGP-LS TLVs, refer to Border Gateway Protocol - Link State (BGP-LS) Parameters.

Table 5: IOS XR Supported BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs

TLV Code Point	Description	Produced by IS-IS	Produced by OSPFv2	Produced by BGP
256	Local Node Descriptors	X	X	—
257	Remote Node Descriptors	X	X	—
258	Link Local/Remote Identifiers	X	X	—
259	IPv4 interface address	X	X	—
260	IPv4 neighbor address	X		
261	IPv6 interface address	X	—	—
262	IPv6 neighbor address	X	—	—
263	Multi-Topology ID	X	—	—
264	OSPF Route Type	—	X	—
265	IP Reachability Information	X	X	—
266	Node MSD TLV	X	X	—
267	Link MSD TLV	X	X	—
512	Autonomous System	—	—	X
513	BGP-LS Identifier	—	—	X
514	OSPF Area-ID	—	X	—
515	IGP Router-ID	X	X	—
516	BGP Router-ID TLV	—	—	X
517	BGP Confederation Member TLV	—	—	X
1024	Node Flag Bits	X	X	—
1026	Node Name	X	X	—
1027	IS-IS Area Identifier	X	—	—
1028	IPv4 Router-ID of Local Node	X	X	—
1029	IPv6 Router-ID of Local Node	X	—	—
1030	IPv4 Router-ID of Remote Node	X	X	—
1031	IPv6 Router-ID of Remote Node	X	—	—
1034	SR Capabilities TLV	X	X	—
1035	SR Algorithm TLV	X	X	—
1036	SR Local Block TLV	X	X	—

TLV Code Point	Description	Produced by IS-IS	Produced by OSPFv2	Produced by BGP
1039	Flex Algo Definition (FAD) TLV	X	X	—
1044	Flex Algorithm Prefix Metric (FAPM) TLV	X	X	—
1088	Administrative group (color)	X	X	—
1089	Maximum link bandwidth	X	X	—
1090	Max. reservable link bandwidth	X	X	—
1091	Unreserved bandwidth	X	X	—
1092	TE Default Metric	X	X	—
1093	Link Protection Type	X	X	—
1094	MPLS Protocol Mask	X	X	—
1095	IGP Metric	X	X	—
1096	Shared Risk Link Group	X	X	—
1099	Adjacency SID TLV	X	X	—
1100	LAN Adjacency SID TLV	X	X	—
1101	PeerNode SID TLV	—	—	X
1102	PeerAdj SID TLV	—	—	X
1103	PeerSet SID TLV	—	—	X
1114	Unidirectional Link Delay TLV	X	X	—
1115	Min/Max Unidirectional Link Delay TLV	X	X	—
1116	Unidirectional Delay Variation TLV	X	X	—
1117	Unidirectional Link Loss	X	X	—
1118	Unidirectional Residual Bandwidth	X	X	—
1119	Unidirectional Available Bandwidth	X	X	—
1120	Unidirectional Utilized Bandwidth	X	X	—
1122	Application-Specific Link Attribute TLV	X	X	—
1152	IGP Flags	X	X	—
1153	IGP Route Tag	X	X	—
1154	IGP Extended Route Tag	X	—	—
1155	Prefix Metric	X	X	—
1156	OSPF Forwarding Address	—	X	—
1158	Prefix-SID	X	X	—
1159	Range	X	X	—

TLV Code Point	Description	Produced by IS-IS	Produced by OSPFv2	Produced by BGP
1161	SID/Label TLV	X	X	—
1170	Prefix Attribute Flags	X	X	—
1171	Source Router Identifier	X	—	—
1172	L2 Bundle Member Attributes TLV	X	—	—
1173	Extended Administrative Group	X	X	—

Exchange Link State Information with BGP Neighbor

The following example shows how to exchange link-state information with a BGP neighbor:

```
Router# configure
Router(config)# router bgp 1
Router(config-bgp)# neighbor 10.0.0.2
Router(config-bgp-nbr)# remote-as 1
Router(config-bgp-nbr)# address-family link-state link-state
Router(config-bgp-nbr-af)# exit
```

IGP Link-State Database Distribution

A given BGP node may have connections to multiple, independent routing domains. IGP link-state database distribution into BGP-LS is supported for both OSPF and IS-IS protocols in order to distribute this information on to controllers or applications that desire to build paths spanning or including these multiple domains.

To distribute IS-IS link-state data using BGP-LS, use the **distribute link-state** command in router configuration mode.

```
Router# configure
Router(config)# router isis isp
Router(config-isis)# distribute link-state instance-id 32
```

To distribute OSPFv2 link-state data using BGP-LS, use the **distribute link-state** command in router configuration mode.

```
Router# configure
Router(config)# router ospf 100
Router(config-ospf)# distribute link-state instance-id 32
```

Configurable Filters for IS-IS advertisements to BGP-Link State

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Configurable Filters for IS-IS Advertisements to BGP-Link State	Release 7.10.1	This feature allows you to configure a route map to filter IS-IS route advertisements to BGP-Link State (LS). It also provides a per-area configuration knob to disable IS-IS advertisements for external and propagated prefixes. This configuration of filters hence reduces the amount of redundant data for external and interarea prefixes sent to the BGP - LS clients. The feature introduces **exclude-external**, **exclude-interarea**, and **route-policy** *name* optional keywords in the **distribute link-state** command.

In a large IS-IS network, there are multiple routers in different areas distributing their link-state databases through BGP-LS. In addition, other protocols, such as OSPF do their own BGP-LS reporting and have routes that are redistributed into IS-IS. This can result in substantial amounts of redundant data for external and interarea prefixes which are sent to the BGP-LS clients only to be discarded.

Rather than sending redundant information, this feature provides the option of limiting the prefixes for which IS-IS TLV information is sent to BGP-LS.

There are three options to filter prefix Type-Length-Values (TLVs) that are reported in BGP-LS and the operators can specify these options on a per-level basis:

- **exclude-external**: Omits information for external prefixes that are redistributed from a different protocol or instance. These are identified by the “X” bit set in its Extended Reachability Attribute Flags or the ‘X’ bit of TLVs 236 and 237.
- **exclude-interarea**: Omits information for interarea prefixes and summaries. These are identified by the ‘R’ bit set in their Extended Reachability Attribute Flags or the ‘up or down’ bit set in TLVs 135, 235, 236, and 237.
- **route-policy***name*: Allows specification of a route-policy to provide filtering based on a set of destination prefixes.

The filtering is implemented at the point where the individual prefix TLVs are read from a label-switched path to generate updates to BGP-LS. It does not affect the advertisement of a node or the link information.

Configure Filters for IS-IS Advertisements to BGP-LS

Configuration Example

You can configure any of these filters for IS-IS advertisements to BGP-LS:

```
Router#config
Router(config)#router isis 1
```

```
Router(config-isis)#distribute link-state exclude-external
Router(config-isis)#commit

Router#config
Router(config)#router isis 1
Router(config-isis)#ddistribute link-state exclude-interarea
Router(config-isis)#commit

Router# config
Router(config)# router isis 1
Router(config-isis)#distribute link-state route-policy isis-rp-1
Router(config-isis)#commit
```

Note This feature does not introduce any new failure modes to IS-IS.

Running Configuration

To check the filter for IS-IS advertisements to BGP-LS, you can run the following command:

```
Router# show running-config
router isis 1
 distribute link-state exclude-external
 commit
 !
!

router isis 1
 distribute link-state exclude-interarea
 commit
 !
!

router isis 1
 distribute link-state route-policy isis-rp-1
 commit
 !
!
```

Configure BGP Proxy Prefix SID

Table 7: Feature History Table

Feature Name	Release	Description
BGP Proxy Prefix SID	Release 7.3.2	This feature is a BGP extension to signal BGP prefix-SIDs. This feature allows you to attach BGP prefix SID attributes for remote prefixes learned over BGP labeled unicast (LU) sessions and propagate them as SR prefixes using BGP LU. This allows an LSP towards non-SR endpoints to use segment routing global block in the SR domain.

To support segment routing, Border Gateway Protocol (BGP) requires the ability to advertise a segment identifier (SID) for a BGP prefix. A BGP-Prefix-SID is the segment identifier of the BGP prefix segment in a segment routing network. BGP prefix SID attribute is a BGP extension to signal BGP prefix-SIDs. However, there may be routers which do not support BGP extension for segment routing. Hence, those routers also do not support BGP prefix SID attribute and an alternate approach is required.

BGP proxy prefix SID feature allows you to attach BGP prefix SID attributes for remote prefixes learnt from BGP labeled unicast (LU) neighbours which are not SR-capable and propagate them as SR prefixes. This allows an LSP towards non SR endpoints to use segment routing global block in a SR domain. Since BGP proxy prefix SID uses global label values it minimizes the use of limited resources such as ECMP-FEC and provides more scalability for the networks.

BGP proxy prefix SID feature is implemented using the segment routing mapping server (SRMS). SRMS allows the user to configure SID mapping entries to specify the prefix-SIDs for the prefixes. The mapping server advertises the local SID-mapping policy to the mapping clients. BGP acts as a client of the SRMS and uses the mapping policy to calculate the prefix-SIDs.

Configuration Example:

This example shows how to configure the BGP proxy prefix SID feature for the segment routing mapping server.

```
RP/0/RSP0/CPU0:router(config)# segment-routing
RP/0/RSP0/CPU0:router(config-sr)# mapping-server
RP/0/RSP0/CPU0:router(config-sr-ms)# prefix-sid-map
RP/0/RSP0/CPU0:router(config-sr-ms-map)# address-family ipv4
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 1.1.1.1/32 10 range 200
RP/0/RSP0/CPU0:router(config-sr-ms-map-af)# 192.168.64.1/32 400 range 300
```

This example shows how to configure the BGP proxy prefix SID feature for the segment-routing mapping client.

```
RP/0/RSP0/CPU0:router(config)# router bgp 1
RP/0/RSP0/CPU0:router(config-bgp)# address-family ip4 unicast
RP/0/RSP0/CPU0:router(config-bgp-af)# segment-routing prefix-sid-map
```

Verification

These examples show how to verify the BGP proxy prefix SID feature.

```
RP/0/RSP0/CPU0:router# show segment-routing mapping-server prefix-sid-map ipv4
 detail
Prefix
1.1.1.1/32
    SID Index:      10
    Range:          200
    Last Prefix:    1.1.1.200/32
    Last SID Index: 209
    Flags:
Number of mapping entries: 1


RP/0/RSP0/CPU0:router# show bgp ipv4 labeled-unicast 192.168.64.1/32

BGP routing table entry for 192.168.64.1/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                117         117
  Local Label: 16400
Last Modified: Oct 25 01:02:28.562 for 00:11:45Paths: (2 available, best #1)
 Advertised to peers (in unique update groups):
   201.1.1.1
 Path #1: Received by speaker 0  Advertised to peers (in unique update groups):
    201.1.1.1
  Local
   20.0.101.1 from 20.0.101.1 (20.0.101.1)      Received Label 61
   Origin IGP, localpref 100, valid, internal, best, group-best, multipath, labeled-unicast

   Received Path ID 0, Local Path ID 0, version 117
  Prefix SID Attribute Size: 7
  Label Index: 1

 RP/0/RSP0/CPU0:router# show route ipv4 unicast 192.68.64.1/32 detail

Routing entry for 192.168.64.1/32
  Known via "bgp 65000", distance 200, metric 0, [ei]-bgp, labeled SR, type internal
  Installed Oct 25 01:02:28.583 for 00:20:09
  Routing Descriptor Blocks
    20.0.101.1, from 20.0.101.1, BGP multi path
      Route metric is 0
      Label: 0x3d (61)
      Tunnel ID: None
      Binding Label: None
      Extended communities count: 0
      NHID:0x0(Ref:0)
   Route version is 0x6 (6)
  Local Label: 0x3e81 (16400)
  IP Precedence: Not Set
  QoS Group ID: Not Set
  Flow-tag: Not Set
  Fwd-class: Not Set
  Route Priority: RIB_PRIORITY_RECURSIVE (12) SVD Type RIB_SVD_TYPE_LOCAL
  Download Priority 4, Download Version 242
  No advertising protos.


RP/0/RSP0/CPU0:router# show cef ipv4 192.168.64.1/32 detail
192.168.64.1/32, version 476, labeled SR, drop adjacency, internal 0x5000001 0x80 (ptr
0x71c42b40) [1], 0x0 (0x71c11590), 0x808 (0x722b91e0)
 Updated Oct 31 23:23:48.733
```

```
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
 Extensions: context-label:16400
  gateway array (0x71ae7e78) reference count 3, flags 0x7a, source rib (7), 0 backups
                [2 type 5 flags 0x88401 (0x722eb450) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0x71c11590, sh-ldi=0x722eb450]
  gateway array update type-time 3 Oct 31 23:49:11.720
 LDI Update time Oct 31 23:23:48.733
 LW-LDI-TS Oct 31 23:23:48.733
   via 20.0.101.1/32, 0 dependencies, recursive, bgp-ext [flags 0x6020]
    path-idx 0 NHID 0x0 [0x7129a294 0x0]
    recursion-via-/32
    unresolved
     local label 16400
     labels imposed {ExpNullv6}


RP/0/RSP0/CPU0:router# show bgp labels
BGP router identifier 2.1.1.1, local AS number 65000
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0000000   RD version: 245
BGP main routing table version 245
BGP NSR Initial initsync version 16 (Reached)
BGP NSR/ISSU Sync-Group versions 245/0
BGP scan interval 60 secs

Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        Rcvd Label      Local Label
*>i1.1.1.1/32         1.1.1.1         3               16010
*> 2.1.1.1/32         0.0.0.0         nolabel         3
*> 192.68.64.1/32      20.0.101.1      2               16400
*> 192.68.64.2/32      20.0.101.1      2               16401
```

BGP Best Path Computation using SR Policy Paths

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
BGP Best Path Computation using SR Policy Paths	Release 7.5.2 Release 7.3.4	BGP best-path selection is modified for a prefix when at least one of its paths resolves over the next hop using SR policies (SR policy in "up" state). Under this condition, paths not steered over an SR policy (those using native next-hop resolution) are considered ineligible during best-path selection. You can thus control the best path selection in order to steer traffic, preferably or exclusively, over SR policies with the desired SLA. This feature introduces the **bgp bestpath sr-policy {force | prefer}** command.

BGP selects the best path from the available pool of paths such as iBGP, eBGP, color, or noncolor paths with native next hop and SR policy next hop. BGP uses either native next hop or an SR policy next hop for best path computation. However, BGP might not consider SR policy next hop for best path computation due to other factors in best path selection. By default, BGP considers a native next hop for the best path computation during the failure.

For more information, see Best path calculation algorithm.

When multiple advertisements of the same BGP prefix are received where some have extended community color, SRTE headend with BGP multi-path enabled installs multiple routes with or without extended community color. It may be required to exclude the path resolving over native next hop SR policy paths from BGP best path selection when a prefix has multiple paths in the presence of one BGP path with the extended community color that is resolved over the SR policy.

You may want to use the egress PE to exit a domain using local preference or other attributes before the next hop metric selection. In such scenarios, when SR policy of the primary path fails, the best path is resolved over a regular IGP next hop that is the default mode of operation. Traffic doesn't select the backup path with SR policy, instead traffic moves to native LSP on the primary path.

The BGP Best Path Computation using SR Policy Paths feature allows the BGP to use the path with SR policy as the best-path, backup, and multipath.

When this feature is enabled, some paths are marked as an ineligible path for BGP best path selection. Existing BGP best path selection order is applied to the eligible paths.

Use either of the following modes for the BGP to select the SR policy path as the best path for the backup path:

- Force mode: When force mode is enabled, only SR policy paths are considered for best path calculation. Use the **bgp bestpath sr-policy force** command to enable this mode.

  In a network, when at least one path has an active SR policy, the following paths are marked as ineligible for best path selection:

  - iBGP paths with noncolor or color paths with SR policy that isn't active.
  - eBGP with color and SR policy isn't active.
  - eBGP noncolor paths

> **Note** Local and redistributed BGP paths are always eligible for best path selection.

- Prefer mode: When prefer mode is enabled, SR policy paths and eBGP noncolor paths are eligible for best path calculation.

  Use the **bgp bestpath sr-policy prefer** command to enable this mode.

  In a network, when at least one path has an active SR policy, the following paths are marked as ineligible for best path selection:

  - iBGP paths with noncolor or color paths with SR policy that isn't active.
  - eBGP with color and SR policy isn't active.

> **Note** Local and redistributed BGP paths are always eligible for best path selection.

Configure BGP Best Path Computation using SR Policy Paths

To enable the feature, perform the following tasks on the ingress PE router that is the head-end of SR policy:

- Configure route policy.
- Configure SR policy.
- Configure BGP with either prefer or force mode.

Configuration Example

Configure route policies on the egress PE router:

```
Router(config)#extcommunity-set opaque color9001
Router(config-ext)#9001 co-flag 01
Router(config-ext)#end-set
Router(config)#extcommunity-set opaque color9002
Router(config-ext)#9002 co-flag 01
Router(config-ext)#end-set
Router(config)#commitC

Router(config)#route-policy for9001
Router(config-rpl)#set extcommunity color color9001
```

```
Router(config-rpl)# pass
Router(config-rpl)#end-policy

Router(config)#route-policy for9002
Router(config-rpl)#set extcommunity color color9002
Router(config-rpl)#pass
Router(config-rpl)#end-policy
Router(config)#commit


Router#configure
Router(config)#route-policy add_path
Router(config-rpl)#set path-selection backup 1 install multipath-protect advertise
multipath-protect-advertise
Router(config-rpl)#end-policy

Router(config)#route-policy pass-all
Router(config-rpl)#pass
Router(config-rpl)#end-policy
Router(config)#commit
```

Configure SR policy on the egress PE router:

```
Router#configure
Router(config)#segment-routing
Router(config-sr)#traffic-eng
Router(config-sr-te)#segment-list SL201
Router(config-sr-te-sl)#index 1 mpls label 25000
Router(config-sr-te-sl)#policy POLICY_9001
Router(config-sr-te-policy)#binding-sid mpls 47700
Router(config-sr-te-policy)#color 9001 end-point ipv6 ::
Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 10
Router(config-sr-te-policy-path-pref)#explicit segment-list SL201
Router(config-sr-te-sl)#policy POLICY_9002
Router(config-sr-te-policy)#binding-sid mpls 47701
Router(config-sr-te-policy)#color 9002 end-point ipv6 ::
Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 10
Router(config-sr-te-policy-path-pref)#explicit segment-list SL201
Router(config-sr-te-policy-path-pref)#commit
```

Configure BGP on the Egress PE router:

```
Router(config)#router bgp 100
Router(config-bgp)#nsr
Router(config-bgp)#bgp router-id 10.1.1.2
Router(config-bgp)#bgp best-path sr-policy force
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths eibgp 25
Router(config-bgp-af)#additional-paths receive
Router(config-bgp-af)#additional-paths send
Router(config-bgp-af)#additional-paths selection route-policy add_path
Router(config-bgp-af)#redistribute connected
Router(config-bgp-af)#redistribute static
Router(config-bgp-af)#allocate-label all
Router(config-bgp-af)#commit
Router(config-bgp-af)#exit
Router(config-bgp)#neighbor 31::2
Router(config-bgp-nbr)#remote-as 2
Router(config-bgp-nbr)#address-family ipv6 unicast
Router(config-bgp-nbr-af)#route-policy for9001 in
Router(config-bgp-nbr-af)#route-policy pass-all out
```

```
Router(config-bgp-nbr-af)#commit
Router(config-bgp-nbr-af)#exit
Router(config-bgp)#neighbor 32::2
Router(config-bgp-nbr)#remote-as 2
Router(config-bgp-nbr)#address-family ipv6 unicast
Router(config-bgp-nbr-af)#route-policy for9002 in
Router(config-bgp-nbr-af)#route-policy pass-all out
Router(config-bgp-nbr-af)#commit
```

Verification

The following show output shows that when the **force** option is enabled, the configured SR policy path is selected as the best path instead of the default best path.

```
Router#show bgp ipv6 unicast 2001:DB8::1 brief
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, r RIB-failure, S stale, N Nexthop-discard
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop            Metric LocPrf Weight Path
* 2001:DB8::1         10:1:1::55                     100      0 2 i
* i                   10:1:1::55                     100      0 2 i

*                     30::2                                   0 2 I
*>                    31::2 C:9001                            0 2 I
*                     32::2 C:9002                            0 2 I
Router#
```

Use the following command to compare the best paths:

```
Router#show bgp ipv6 unicast 2001:DB8::1 bestpath-compare
BGP routing table entry for 2001:DB8::1
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               7641        7641
    Flags: 0x240232b2+0x20050000; multipath; backup available;
Last Modified: Dec  7 03:43:57.200 for 00:34:48
Paths: (24 available, best #4)
  Advertised IPv6 Unicast paths to update-groups (with more than one peer):
    0.3 0.4
  Advertised IPv6 Unicast paths to peers (in unique update groups):
    10.1.1.55
  Path #1: Received by speaker 0
  Flags: 0x2000000000020005, import: 0x20
  Flags2: 0x00
  Not advertised to any peer
  2
    10:1:1::55 (metric 30) from 10.1.1.55 (10.1.1.55), if-handle 0x00000000
      Origin IGP, localpref 100, valid, internal
      Received Path ID 1, Local Path ID 0, version 0
      Extended community: Color[CO-Flag]:8001[01]
      Non SR-policy path is ignored due to config knob
  Path #2: Received by speaker 0
  Flags: 0x2000000000020005, import: 0x20
  Flags2: 0x00
  Not advertised to any peer
  2
    10:1:1::55 (metric 30) from 10.1.1.55 (10.1.1.55), if-handle 0x00000000
      Origin IGP, localpref 100, valid, internal
      Received Path ID 3, Local Path ID 0, version 0
      Extended community: Color[CO-Flag]:8002[01]
      Non SR-policy path is ignored due to config knob
  Path #3: Received by speaker 0
  Flags: 0x3000000000060001, import: 0x20
```

```
  Flags2: 0x00
  Advertised IPv6 Unicast paths to update-groups (with more than one peer):
    0.4
  Advertised IPv6 Unicast paths to peers (in unique update groups):
    10.1.1.55
  2
    30::2 from 30::2 (198.51.100.1), if-handle 0x00000000
      Origin IGP, localpref 100, weight 65534, valid, external, backup, add-path
      Received Path ID 0, Local Path ID 2, version 7641
      Origin-AS validity: (disabled)
      Non SR-policy path is ignored due to config knob
  Path #4: Received by speaker 0
  Flags: 0xb000000001070001, import: 0x20
  Flags2: 0x00
  Advertised IPv6 Unicast paths to update-groups (with more than one peer):
    0.3 0.4
  Advertised IPv6 Unicast paths to peers (in unique update groups):
    10.1.1.55
  2
    31::2 C:9001 (bsid:48900) from 31::2 (198.51.100.2), if-handle 0x00000000
      Origin IGP, localpref 100, valid, external, best, group-best, multipath
      Received Path ID 0, Local Path ID 1, version 7641
      Extended community: Color[CO-Flag]:9001[01]
      Origin-AS validity: (disabled)
      SR policy color 9001, ipv6 null endpoint, up, not-registered, bsid 48900

      best of AS 2, Overall best
  Path #5: Received by speaker 0
  Flags: 0xb000000000030001, import: 0x20
  Flags2: 0x00
  Not advertised to any peer
  2
    32::2 C:9002 (bsid:48901) from 32::2 (198.51.100.3), if-handle 0x00000000
      Origin IGP, localpref 100, valid, external, multipath
      Received Path ID 0, Local Path ID 0, version 0
      Extended community: Color[CO-Flag]:9002[01]
      Origin-AS validity: (disabled)
      SR policy color 9002, up, not-registered, bsid 48901
      Higher router ID than best path (path #4)
```

Use the **show bgp process** command to verify which mode is enabled.

In the following example, you see that the **force** mode is enabled.

```
Router#show bgp process
BGP Process Information:
BGP is operating in STANDALONE mode
Autonomous System number format: ASPLAIN
Autonomous System: 100
Router ID: 10.1.1.2 (manually configured)
Default Cluster ID: 10.1.1.2
Active Cluster IDs:  10.1.1.2
Fast external fallover enabled
Platform Loadbalance paths max: 64
Platform RLIMIT max: 8589934592 bytes
Maximum limit for BMP buffer size: 1638 MB
Default value for BMP buffer size: 1228 MB
Current limit for BMP buffer size: 1228 MB
Current utilization of BMP buffer limit: 0 B
Neighbor logging is enabled
Enforce first AS enabled
Use SR-Policy admin/metric of color-extcomm Nexthop during path comparison: disabled
SR policy path force is enabled
Default local preference: 100
```

```
Default keepalive: 60
Non-stop routing is enabled
Slow peer detection enabled
ExtComm Color Nexthop validation: RIB

Update delay: 120
Generic scan interval: 60
Configured Segment-routing Local Block: [0, 0]
In use Segment-routing Local Block: [15000, 15999]
Platform support mix of sr-policy and native nexthop: Yes

Address family: IPv4 Unicast
Dampening is not enabled
Client reflection is enabled in global config
Dynamic MED is Disabled
Dynamic MED interval : 10 minutes
Dynamic MED Timer : Not Running
Dynamic MED Periodic Timer : Not Running
Scan interval: 60
Total prefixes scanned: 33
Prefixes scanned per segment: 100000
Number of scan segments: 1
Nexthop resolution minimum prefix-length: 0 (not configured)
IPv6 Nexthop resolution minimum prefix-length: 0 (not configured)
Main Table Version: 12642
Table version synced to RIB: 12642
Table version acked by RIB: 12642
IGP notification: IGPs notified
RIB has converged: version 2
RIB table prefix-limit reached ?  [No], version 0
Permanent Network Unconfigured

Node               Process     Nbrs Estb Rst Upd-Rcvd Upd-Sent Nfn-Rcv Nfn-Snt
node0_RSP1_CPU0    Speaker       53    3   2      316      823       0      53
```

SRv6 Double Recursion for Multi-Layer BGP Underlay

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
SRv6 Double Recursion for Multi-Layer BGP Underlay	Release 24.4.1	

Feature Name	Release Information	Feature Description
		Introduced in this release on: Fixed Systems (8100, 8200); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) The feature introduces support to SRv6 double recursion where a network service such as BGP VPN (Layer 2/Layer 3) requires multiple layers of resolution, specifically where one routing layer resolves over another before reaching its final destination. You can achieve double recursion by collapsing the underlay, which typically involves protocols like IGP or BGP in the packet forwarding chain, you can achieve three level load balancing, allowing an even distribution of traffic across multiple layers of the network stack. The feature is supported on the ingress Provider Edge (PE) router. Previously, SRv6 supported only two levels of load balancing, which works for traditional service provider setups. The feature introduces these changes: **CLI:** • **tag-map tag** *<value>* **map forwarding-hierarchy level-2-used-as-nexthop** • The **show cef ipv6** *ipv6-prefixes* and **show cef ipv4** *ipv4-prefixes* commands are enhanced to include the Layer 2 prefix information, which resolve as nexthop Layer 3 prefixes. **Yang Data Models:** • `Cisco-IOS-XR-um-router-rib-cfg:router` (see GitHub, Yang Data Models Navigator) data

Feature Name	Release Information	Feature Description
		model.

SRv6 Double Recursion and Load Balancing

SRv6 double recursion refers to scenarios where a network service such as BGP VPN (Layer 2/Layer 3) requires multiple layers of resolution, specifically where one routing layer resolves over another before reaching its final destination.

The SRv6 double recursion feature is designed to support three level load balancing by collapsing the underlay, which typically involves protocols like IGP or BGP in the forwarding chain. This involves specific configurations in RIB, BGP, and static routes to indicate IPv4 and IPv6 unicast routes with Layer 2 prefixes, which resolve as nexthop Layer 3 prefixes.

Without this feature, the traffic drops the Layer 3 prefixes leading to packet loss.

Key Benefits of SRv6 Double Recursion

- Support for complex network scenarios: In traditional service provider (SP) networks, BGP VPN services typically resolve directly over IGP reachability, meaning there is a straightforward, single-level resolution path from the service to the IGP underlay. However, with the evolution of networking, especially in data centers, new architectures have emerged where BGP underlay networks are used. In these scenarios, BGP VPN services resolve over a BGP underlay, which in turn resolves over IGP or directly connected routes.
- Support for enhanced load balancing: SRv6 double recursion allows the routing platform to balance traffic more efficiently across all available paths, enhancing overall network performance and reducing congestion. Proper load balancing at each level of recursion helps in distributing traffic evenly, avoiding bottlenecks, and ensuring that packets take optimal paths even in complex topologies.
- Improved network flexibility: Double recursion allows advanced routing solutions in SP and data center networks, accommodating complex use cases beyond single-layer limitations.

Single and Double Recursion in BGP VPN Services

To optimize the network routing efficiency, it is important to understand the resolution processes of BGP VPN services.

- Single Recursion: BGP VPN service → IGP reachability. In single recursion, the BGP VPN service directly resolves to IGP (Interior Gateway Protocol) reachability. Single recursion is required when a BGP VPN service can directly resolve to an IGP route without any intermediary steps.
- Double Recursion: BGP VPN service → BGP underlay reachability → IGP reachability. In double recursion, the BGP VPN service first resolves over a BGP underlay, which then leads to IGP reachability. Double recursion is necessary in cases where a BGP VPN service cannot resolve directly to an IGP route but instead needs to traverse a BGP underlay first. This creates a layered resolution process that standard single-level load balancing cannot handle effectively.

Usage Guidelines and Limitations for SRv6 Double Recursion

The following usage guidelines and limitations apply:

- Set the locator prefix for IPv6 prefixes that are Layer 2 prefixes.
- If the collapsed BGP paths is a combination of IGP IPv6 and SRv6, the router filters out only the IPv6 paths.
- You cannot use a combination of BGP-SU (BGP Service Unicast) and BGP-IP paths at Layer 2.
- For a collapsed chain, there must be an encapsulation ID on BGP if IGP is SRv6.

Configure SRv6 Double Recursion for Multi-Layer BGP Underlay

To configure SRv6 double recursion for multi-layer BGP underlay:

- Enable the hardware support for BGP-LU.
- Configure the BGP-SR or BGP-IP IPv4 and IPv6 unicast routes in the RIB, to ensure that these Layer 2 prefixes could be used as nexthop for Layer 3 prefixes.
- Assign or map specific tags to Layer 2 routes in the RIB for IPv4 and IPv6 unicast routes in the VRF.

Before you begin

- For Layer 2 IPv6 prefixes, set the locator prefix using the **prefix-set** command in RIB.
- Configure the set of IPv4 and IPv6 Layer 2 prefixes, which resolve as nexthop Layer 3 prefixes using the **prefix-set** command.

Procedure

Step 1 Enable the hardware support for BGP-LU to allow the ingress PE router to advertise and forward the MPLS-labelled unicast routes.

Example:

```
Router#config
Router(config)#hw-module profile cef bgplu enable
```

Step 2 Configure the IPv4 and IPv6 unicast routes in the RIB, ensuring that these Layer 2 prefixes could be used as nexthop for Layer 3 prefixes.

Example:

```
Router#config
Router(config)#router bgp 100
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#table-policy level2-ipv4-policy
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#table-policy level2-ipv6-policy
Router(config-bgp-af)#commit
```

Step 3 Assign the required tag to the IPv4 and IPv6 unicast routes.

For these IPv4 and IPv6 routes, the RIB assigns the tags and forwards the route information to the Forwarding Information Base (FIB), indicating that these Layer 2 prefixes resolve as nexthop for Layer 3 prefixes.

Example:

IPv4 Layer 2 prefixes:

```
Router(config)#route-policy level2-ipv4-policy
Router(config-rpl)#if destination in level2_prefixes-ipv4 then
Router(config-rpl-if)#set tag 100
Router(config-rpl-if)#else
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
Router(config)#commit
```

Example:

IPv6 Layer 2 prefixes:

```
Router(config)#route-policy level2-ipv6-policy
Router(config-rpl)#if destination in level2_prefixes-ipv6 then
Router(config-rpl-if)#set tag 100
Router(config-rpl-if)#endif
Router(config-rpl)#if destination in level2_locators then
Router(config-rpl-if)#set locator-prefix
Router(config-rpl-if)#else
Router(config-rpl-else)#pass
Router(config-rpl-else)#endif
Router(config-rpl)#end-policy
```

Step 4 Map the tags in the RIB for the IPv4 and IPv6 unicast routes.

In the following example, the RIB maps all the IPv4 and IPv6 routes tagged with the value 100, which indicates that these routes resolve as nexthop for Layer 3 prefixes. The RIB adds the *FIB_UPDATE_ROUTE_FLAG_EXTN_LVL2_HAS_DEPENDENT* flag when it sends the route update to the FIB.

Example:

```
Router#config
Router(config)#router rib
Router(config-rib)#tag-map tag 100 map forwarding-hierarchy level-2-used-as-nexthop
Router(config-rib)#commit
```

Step 5 Verify the running configuration using the **show running-config** command.

Example:

```
hw-module profile cef bgplu enable
!
router bgp 100
 address-family ipv4 unicast
  table-policy level2-ipv4-policy
 !
 address-family ipv6 unicast
  table-policy level2-ipv6-policy
 !
!
router rib
 tag-map tag 100 map forwarding-hierarchy level-2-used-as-nexthop
 !
!
```

Step 6 Use the **show cef ipv4** and **show cef ipv6** commands to verify the Layer 2 collapsed prefixes.

In the following example, the IPv6 Layer 2 prefixes, which resolve as nexthop for Layer 3 prefixes are collapsed. This is indicated by the **collapsed** keyword in the output. The SRv6 SID lists indicate the different encapsulation layers or hierarchy.

Example:

```
Router#show cef ipv6 2001:DB8:A:B::1/64
Thu Jun  6 12:48:52.399 EDT
2001:DB8:A:B::1/64, version 8, SRv6 Headend, internal 0x1000001 0x0 (ptr 0x63851c98) [1], 0x1400
(0x63851da0), 0x0 (0x638b2128)
 Updated Jun  6 12:41:10.589
 Prefix Len 64, traffic index 0, precedence n/a, priority 0, encap-id 0x11deadbeef
  gateway array (0x61e1a798) reference count 1, flags 0x10, source rib (7), 0 backups
                [2 type 3 flags 0x40008501 (0x63853e38) ext 0x0 (0x0) (collapsed)]
  LW-LDI[type=3, refc=1, ptr=0x63851da0, sh-ldi=0x63853e38]
  gateway array update type-time 1 Jun  6 12:41:10.589
 LDI Update time Jun  6 12:41:10.629
 LW-LDI-TS Jun  6 12:41:10.629
 Accounting: Disabled
   via 2001:DB8::1/128, 1 dependency, recursive [flags 0x3000000]
    path-idx 0 NHID 0x0 [0x63a2c098 0x0]
    next hop 2001:DB8::1/128 via  2001:DB8::1
    SRv6 H.Encaps.Red SID-list {2001:DB8:1:e002::}
     SRv6 H.Insert.Red SID-list {}
      SRv6 H.Insert.Red SID-list {2001:DB8:3:: 2001:DB8:4::}
   via 2001:DB8::1/128, 1 dependency, recursive [flags 0x3000000]
    path-idx 1 NHID 0x0 [0x63a2c270 0x0]
    next hop 2001:DB8::1/128 via 2001:DB8::1
    SRv6 H.Encaps.Red SID-list {2001:DB8:1:e002::}
       SRv6 H.Insert.Red SID-list {}

    Load distribution: 0 1 2 2 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   UNKNOWN intf 0x00000013   10::2
    1     Y   UNKNOWN intf 0x00000014   20::2
    2     Y   UNKNOWN intf 0x00000013   10::2
    3     Y   UNKNOWN intf 0x00000013   10::2
```

Step 7 For IPv6 or IPv4 static routes, use the **router static** command to configure and map the tags in RIB:

Example:

```
Router#config
Router(config)#router static
Router(config-static)#address-family ipv6 unicast
Router(config-static-afi)#2001:DB8:8::/48 4::4 tag 100
Router(config-static-afi)#commit
```