Label Allocation and MPLS Support

This chapter provides comprehensive guidance on BGP labeled unicast (BGP LU) features, enabling scalable MPLS transport across diverse network topologies, including multi-area and multi-AS environments. It details various BGP LU implementations like RSVP-TE, IPv6, MPLS IP POP, and fast convergence with PIC edge. You will also learn how to optimize label allocation and steer BGP control-plane traffic over IP-only paths for enhanced network efficiency.

BGP labeled unicast

The BGP labeled unicast is a routing feature that

- provides MPLS transport between Provider Edge (PE) routers separated by multiple IGP boundaries or autonomous systems
- uses autonomous system border routers (ASBRs) to advertise PE loopback prefixes and their MPLS label bindings via iBGP between area border routers (ABRs) and eBGP between ASBRs, and
- supports multihop eBGP between PEs in different ASes to exchange VPN routes and enables services like 6PE over BGP LU connectivity.

BGP labeled unicast key components

- PE routers are routers at the edge of a provider network that connect to customer networks.
- ASBRs connect different autonomous systems.
- iBGP and eBGP are internal and external Border Gateway Protocol sessions used for route exchange within and between ASes.

Table 1: Feature History Table

Feature Name	Release	Description
BGP Labeled Unicast	Release 24.4.1	Introduced in this release on: Fixed Systems (8700)(select variants only*). This feature enables seamless MPLS transport between Provider Edge (PE) routers across multiple IGP boundaries or autonomous systems by utilizing Autonomous Systems Border Routers (ASBRs) to advertise loopback prefixes and MPLS label bindings through iBGP and eBGP. *This feature is now supported on Cisco 8712-MOD-M Routers.

BGP labeled unicast overview

BGP labeled unicast, also known as unified MPLS, helps you scale MPLS transport across complex network topologies involving multiple IGP areas and AS boundaries. By using BGP LU, you reduce the scale of IGP labeled prefixes and adjacencies, which improves network efficiency.

To avoid unintended scale reduction on routers not configured for BGP LU, you must configure the hw-module command before enabling BGP LU. After configuring this command, you need to restart the router for the changes to take effect.

For example, you can run 6PE and other MPLS VPN services between PEs separated by multiple ASes or IGP areas, simplifying route advertisement and label distribution.

This approach helps you manage large-scale MPLS networks more effectively by lowering the overhead on IGP and BGP processes while maintaining connectivity and service capabilities.

Guidelines for BGP labeled unicast

To ensure proper configuration and compatibility, follow these guidelines and restrictions for the BGP LU feature:

- Use only per-VRF label mode on Cisco 8000; other label modes are not supported.
- Use Label Distribution Protocol (LDP) or Segment Routing (SR) for the transport underlay; do not use Traffic Engineering (TE).
- Do not enable the BGP Prefix Independent Convergence (PIC) edge feature.
- Do not configure L3VPN or 6VPE over BGP LU.
- Enable the BGP Prefix Independent Convergence (PIC) core feature only if needed.
- Do not use the deprecated **label-allocation-mode** command in IOS XR Release 7.4.1 or later; use the **label mode** command under the configured address-family instead.

- Avoid using the deprecated **label-allocation-mode** command to ensure compatibility with current releases.

Features supported by BGP labeled unicast

BGP labeled unicast includes the following supported features:

- BGP LU with inter-AS Option C
- 6PE over MPLS transport using LDP or SR
- BGP PIC core
- L3VPN and L2VPN services over BGP LU using LDP or SR transport

How MPLS connectivity for BGP labeled unicast across multiple OSPF areas works

In large service provider networks, MPLS with BGP labeled unicast enables scalable connectivity across multiple OSPF areas by using label switching and iBGP-based label distribution. This process is essential for connecting remote sites and maintaining efficient, reliable routing in complex environments.

Summary

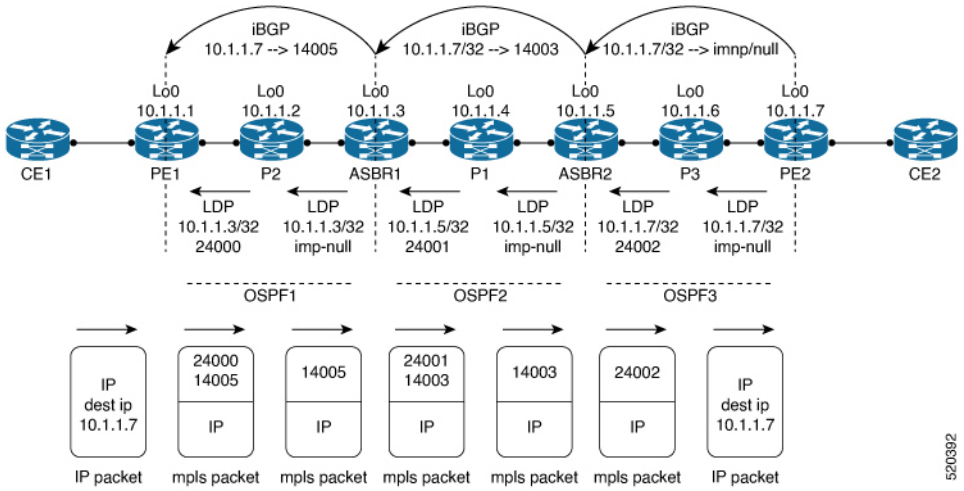
Effective BGP labeled unicast implementation across diverse OSPF areas relies on the coordinated functions of these key components:

- PE1 and PE2: Provider edge routers that establish connectivity across OSPF areas.
- OSPF areas (OSPF1, OSPF2, OSPF3): Separate OSPF instances running in different areas to support routing.
- Label Distribution Protocol (LDP): Provides transport label switching between OSPF areas.
- iBGP: Internal BGP used to advertise labels and loopback addresses between PE2 and PE1 via intermediate routers.
- ASBR1 and ASBR2: Autonomous system border routers that assign and swap labels during packet forwarding.
- P1, P2, P3: Intermediate routers in the MPLS network forwarding iBGP and MPLS traffic.

This coordinated interplay of edge, core, and routing protocols facilitates robust and scalable MPLS transport for BGP labeled unicast across disparate OSPF areas.

Workflow

Figure 1: BGP labeled unicast (Intra-Autonomous System) control plane and data plane



These stages describe how the BGP labeled unicast (intra-autonomous aystem) control plane and data plane works:

1. Establishing iBGP connectivity
   - Actor: PE2, intermediate routers (P3, ASBR2, P1, ASBR1, P2), PE1
   - Action: PE2 sends iBGP updates to PE1 through the path P3 → ASBR2 → P1 → ASBR1 → P2. PE1 learns PE2's loopback address to establish connectivity.
2. Label advertisement and allocation
   - Actor: PE2, ASBR2, ASBR1, PE1
   - Actions:
     - PE2 advertises its loopback address 10.1.1.7 with a BGP label (implicit null) via iBGP to ASBR2.
     - ASBR2 assigns a local label 14003 and advertises it to ASBR1.
     - ASBR1 assigns label 14005 and advertises it to PE1.
     - PE1 learns the prefix and label 14005, with ASBR1 as the BGP next hop.
3. Packet forwarding from PE1 to PE2
   - Actor: PE1, ASBR1, ASBR2
   - Actions:
     - PE1 sends traffic to PE2 with two labels: the BGP-LU label 14005 and the transport LDP label 24000 on top.
     - The transport LDP label carries the packet to ASBR1.
     - ASBR1 swaps the BGP-LU label from 14005 to 14003, applies transport LDP label 24001, and forwards the packet to ASBR2.

- ASBR2 uses an implicit null BGP-LU label and pushes transport label 24002 to forward the packet to PE2.

Result

This process enables PE1 and PE2 to communicate across multiple OSPF areas using MPLS with label switching, ensuring efficient and scalable inter-area connectivity.

How inter-AS connectivity works using eBGP

Summary

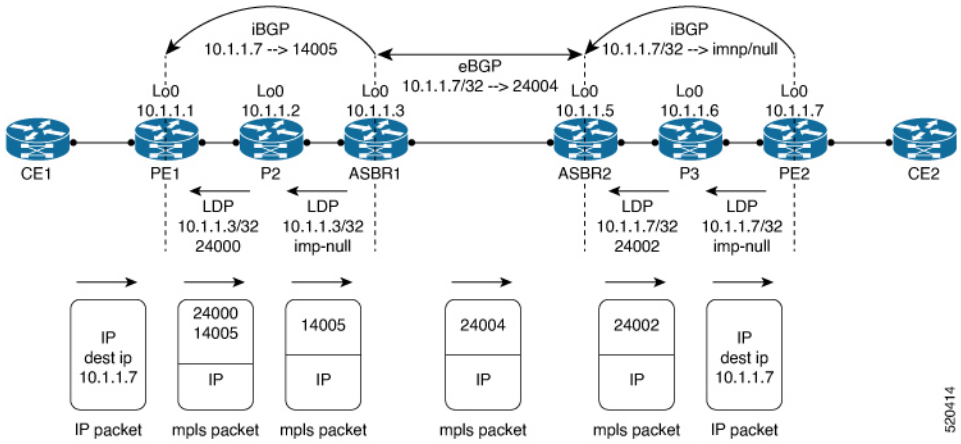
The key components involved in the process are:

- PE1 and PE2: Provider Edge routers serving as the source and destination for IP packets.
- ASBR1 and ASBR2: Autonomous System Border Routers performing eBGP peering, label advertisement, allocation, and swapping.
- LDP (Label Distribution Protocol): Used to signal transport labels across the MPLS path.
- BGP-LU (BGP Labeled Unicast): Facilitates label route advertisement between routers across and within AS boundaries.
- IGP (Interior Gateway Protocol): Supports local MPLS path computation where applicable.

This process enables reliable MPLS connectivity across multiple autonomous systems. eBGP connects ASBRs at AS boundaries for route and label exchange. By combining BGP-LU, LDP, and correct label operations, providers ensure efficient IP packet forwarding between PE routers over a multi-AS MPLS backbone. Understanding label allocation, advertisement, and swapping across ASBRs is key to inter-AS MPLS success.

Workflow

Figure 2: BGP labeled unicast (Intra-Autonomous System Option C) control plane and data plane



These stages describe how inter-AS connectivity using eBGP works:

1. Label advertisement by PE2

- Actor: PE2
- Action: PE2 advertises the BGP-LU label (implicit null) to ASBR2 via iBGP.

2. Label allocation and advertisement by ASBR2
   - Actor: ASBR2
   - Actions:
     - ASBR2 prefers the IGP MPLS path with LDP label 24002.
     - It allocates a local label 24004 for loopback address 10.1.1.7.
     - ASBR2 advertises label 24004 to ASBR1.

3. Label creation and advertisement by ASBR1
   - Actor: ASBR1
   - Actions
     - ASBR1 creates a local label 14005.
     - It advertises label 14005 to PE1

4. Packet forwarding from PE1 to PE2
   - Actor: PE1, ASBR1, ASBR2
   - Actions
     - PE1 sends IP packets with BGP label 14005 and transport label 24000 to ASBR1.
     - ASBR1 swaps the BGP-LU label 14005 to 24004.
     - ASBR2 pushes the LDP label 24002 and forwards the packet to PE2.

Result

This process ensures that IP packets from PE1 are efficiently delivered to PE2 across autonomous system boundaries using eBGP and MPLS label switching, enabling robust and scalable inter-AS connectivity.

How MPLS connectivity with multihop eBGP between multiple ASes works

Summary

The key components involved in the process are:

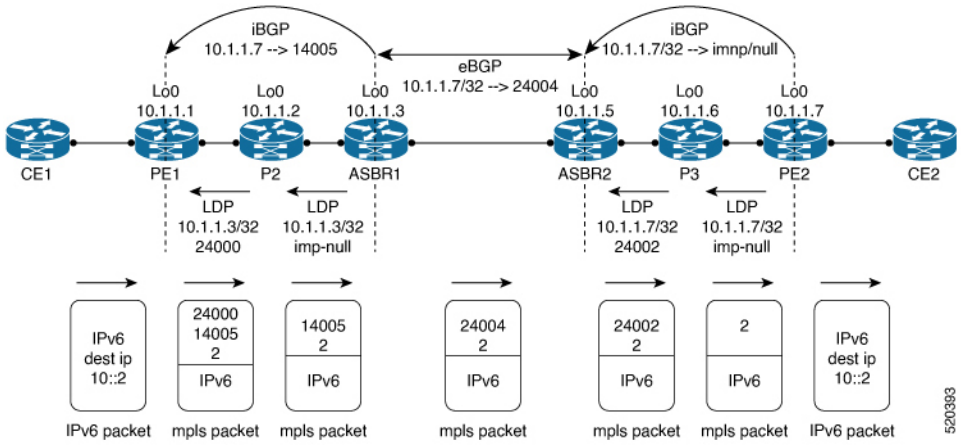
- PE1 router: Initiates label imposition on IPv6 packets and exchanges 6PE routes with PE2 using Multihop eBGP.
- PE2 router: Advertises IPv6 prefixes with explicit null labels and performs IPv6 lookup to forward packets.
- ASBR1 router: Swaps BGP Label Unicast (LU) labels and forwards packets between ASes.

- ASBR2 router: Adds Label Distribution Protocol (LDP) labels and forwards packets toward PE2.
- P3 router: Pops the top LDP label before delivering packets to PE2.

This process describes how PE1 and PE2 use MPLS with multihop eBGP across multiple ASes to enable seamless IPv6 forwarding, focusing on 6PE route exchange and the sequence of label operations.

Workflow

Figure 3: 6PE over BGP LU (inter-AS option C) control plane and data plane



These stages describe how MPLS connectivity with multihop eBGP between multiple ASes works:

1. Establishing a multihop eBGP session and exchanging routes:

   PE1 and PE2 establish a multihop eBGP session across multiple ASes to exchange 6PE routes with associated labels.

2. Advertising IPv6 prefixes by PE2:

   PE2 advertises an IPv6 prefix (e.g., 10::2/128) with the 6PE label set to the IPv6 explicit null label.

3. IPv6 packet arrival at PE1:

   An IPv6 packet destined for 10::2/128 arrives at PE1.

4. Label imposition at PE1:

   The PE1 router imposes labels on the packet in the following order:

   - First, the 6PE label with value 2 (IPv6 explicit null).
   - Next, the BGP label 14005.
   - Finally, the next-hop LDP label 14005 for the BGP LU next hop.

5. Label swapping at ASBR1:

   ASBR1 receives the packet, swaps the BGP-LU label from 14005 to 24004, and forwards it to ASBR2.

6. LDP label imposition by ASBR2:

   ASBR2 adds an LDP label on top of the 6PE label 2 and forwards the packet to P3.

7. Label popping at P3:

   P3 pops the top LDP label so that PE2 receives the packet with only the 6PE explicit null label remaining.

8. IPv6 lookup and forwarding at PE2:

   PE2 performs an IPv6 lookup on the packet and forwards it to the final destination.

Result

This process enables seamless IPv6 packet forwarding between PE routers across multiple ASes by leveraging MPLS with multihop eBGP and label stacking. The correct sequence of label imposition, swapping, and removal ensures that IPv6 traffic is efficiently routed and delivered end to end.

Configure BGP labeled unicast

Enable BGP LU to support labeled IPv6 unicast routing on your router.

BGP labeled unicast extends BGP to distribute labeled routes, allowing routers to forward IPv6 packets using MPLS labels. This capability is essential for scalable and flexible IPv6 routing with MPLS label switching.

Before you begin

- Ensure you have administrative access to the router.
- Confirm the router supports BGP-LU and the required hardware module features.

Follow these steps to configure BGP labeled unicast:

Procedure

Step 1 Enable the BGP-LU hardware profile.

Example:

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

Step 2 Restart the router to activate the BGP-LU hardware profile.

Step 3 Enable the BGP-LU feature and configure the BGP router with IPv6 unicast address family settings to support labeled unicast routing.

Example:

```
Router(config)# router bgp 1
Router(config-bgp)# bgp router-id 2001:DB8::1
Router(config-bgp)# address-family ipv6 unicast
Router(config-bgp-af)# redistribute connected route-policy set-lbl-idx
Router(config-bgp-af)# allocate-label all
Router(config-bgp-af)# exit
```

Step 4 Configure a BGP neighbor with an IPv6 address, enabling the labeled-unicast address family and applying inbound and outbound route policies.

Example:

```
Router(config-bgp)# neighbor 2001:DB8::2
Router(config-bgp)# remote-as 1
Router(config-bgp)# update-source Loopback 0
Router(config-bgp)# address-family ipv6 labeled-unicast
```

```
Router(config-bgp)# route-policy pass-all in
Router(config-bgp)# route-policy pass-all out
```

Step 5 Use the **show running-config** to verify the running configuration.

Example:

```
Router# show running-config

hw-module profile cef bgplu enable
!
router bgp 1
 bgp router-id 2001:DB8::1
 address-family ipv6 unicast
 redistribute connected route-policy set-lbl-idx
 allocate-label all
!
 neighbor 2001:DB8::2
 remote-as 1
 update-source Loopback0

!
 address-family ipv6 labeled-unicast
 route-policy pass-all in
 route-policy pass-all out
!
```

BGP labeled unicast is enabled on your router, supporting labeled IPv6 unicast routing with the specified neighbor, and the hardware profile is active.

What to do next

Restart the router after enabling the hardware module profile if you have not already done so to ensure all changes take effect.

BGP labeled unicast over RSVP-TE

A BGP labeled unicast over RSVP-TE is a routing feature that

- enables routers to forward BGP labeled unicast traffic to the BGP-LU next-hop router through RSVP-TE tunnels,
- allows network administrators to select the tunnel path for traffic transport based on your requirements, and
- differentiates traffic by routing packets destined for the tunnel destination address exclusively through Autoroute Announce (AA) tunnels, while routing all other traffic through Forwarding-Adjacency (FA) tunnels.

Table 2: Feature History Table

Feature Name	Release Information	Feature Description

BGP Labeled Unicast over RSVP-TE	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700[ASIC:K100]); Modular Systems (8800 [LC ASIC: P100]) (select variants only*). *This feature is now supported on: • 8212-32FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 8712-MOD-M
BGP Labeled Unicast over RSVP-TE	Release 7.11.1	You can now steer the MPLS traffic as per your requirement instead of relying on what the IGP directs. This feature extends the BGP Labeled Unicast (LU) functionality over RSVP-TE protocol. BGP LU advertises label bindings while RSVP-TE establishes the traffic engineering paths that you specify. This feature allows the provider Edge (PE) routers to forward incoming traffic using the label bindings along the specific path reserved using RSVP-TE. This ability to provide explicit routing ensures optimal use of your network resources. The feature introduces these changes: CLI: • **hw-module profile cef bgplu-over-rsvpte enable** YANG Data Models: • `Cisco-IOS-XR-npu-hw-profile-cfg.yang` (see GitHub, YANG Data Models Navigator)

How BGP labeled unicast path selection and tunnel protection work

BGP-LU with RSVP-TE enables network administrators to select explicit tunnel paths for traffic, assures traffic engineering, and protects service continuity using recovery mechanisms.

Summary

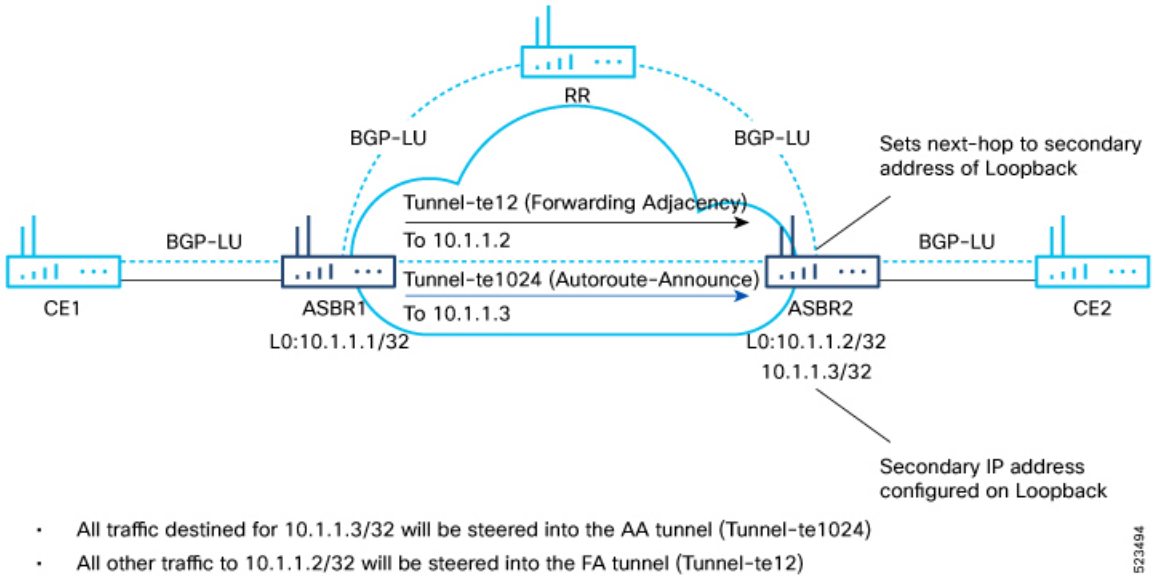
The key components involved in the process are:

- ASBR1: Establishes RSVP-TE tunnels to ASBR2 and forwards traffic to CE2 based on the next-hop address received from ASBR2.
- ASBR2: Receives BGP-LU prefixes and sets next-hop addresses.
- CE1/CE2: Edge routers exchanging traffic.
- RSVP-TE: Tunneling technique for traffic protection.
- Fast Reroute (FRR): Provides resiliency in case of failures.

This process uses BGP-LU and RSVP-TE tunnels to ensure reliable, protected traffic forwarding between customer edge routers across ASBRs, enabling explicit path selection, traffic engineering, and service continuity.

Workflow

Figure 4: BGP labeled unicast over RSVP-TE



These stages describe the process by which BGP-LU and RSVP-TE tunnels are used to establish connectivity, select forwarding paths, and protect traffic between CE1 and CE2 through ASBR1 and ASBR2, ensuring reliable and resilient network service.

1. Establish BGP-LU connections: ASBR1 connects to CE1 and ASBR2 connects to CE2 using BGP labeled unicast.
2. Configure RSVP-TE tunnels: ASBR1 is configured with FA and AA tunnels to ASBR2's primary and secondary IP addresses, respectively.
3. Set BGP-LU next-hop: ASBR2 sets BGP-LU next-hop prefixes from CE2 to the secondary IP address.
4. Forward packets based on next-hop: ASBR1 forwards traffic for CE2 via the AA tunnel based on the next-hop prefix.
5. Select preferred next-hop path: If two paths are learned (RSVP-TE and regular), the regular next-hop path is chosen.
6. Provide failure protection with FRR: Fast Reroute (FRR) protects against link and node failures during forwarding.

Result

Traffic between CE1 and CE2 is forwarded reliably and efficiently, with automatic protection against failures to maintain continuous network service.

Guidelines for BGP LU over RSVP-TE

- Do not configure BGP-LU over RSVP-TE simultaneously with BGP-LU (over NH) and Class-based forwarding (CBF). You must disable BGP-LU and CBF before enabling BGP-LU over RSVP-TE to avoid error messages.
- BGP-LU over RSVP-TE is not supported on Q100-based line cards.
- BGP-LU SR-TE is not supported.
- L3VPN, 6PE, and 6VPE services are not supported with BGP-LU over RSVP-TE.
- Use LDP or SR as the transport underlay. Do not use Traffic Engineering (TE) as the transport underlay.
- Reaching ASBR (BGP next-hop) through both regular next-hop and RSVP-TE is not supported.

Configure BGP-LU over RSVP-TE

Use this task when you need to deploy BGP-LU over RSVP-TE, optimize path selection with Forwarding-Adjacency (FA) and Autoroute Announce (AA) tunnels, and maintain protection against link or node failures.

Enable routers to forward BGP labeled unicast (BGP-LU) traffic through RSVP-TE tunnels, allowing you to select optimal tunnel paths and ensure traffic continuity with fast reroute (FRR) protection.

Before you begin

- Verify existing BGP-LU and Class-Based Forwarding (CBF) configurations are not active.
- Ensure you have appropriate router access and privileges.

Follow these steps to configure BGP-LU over RSVP-TE:

Procedure

Step 1 Disable BGP-LU and CBF configurations:

Example:

```
Router(config)# no hw-module profile cef bgplu enable
Router(config)# no hw-module profile cef cbf enable
```

Step 2 Enable BGP-LU over RSVP-TE and optionally increase tunnel capacity:

Example:

```
Router(config)# hw-module profile cef bgplu-over-rsvpte enable
Router(config)# hw-module profile cef te-tunnel highscale-no-ldp-over-te
```

Step 3 Configure the loopback interface:

Example:

```
Router(config)# interface Loopback1001
Router(config-if)# ipv4 address 10.10.10.10 255.255.255.255
Router(config-if)# exit
```

Step 4 Configure the tunnel interface:

Example:

```
Router(config)# interface tunnel-te1
Router(config-if)# ipv4 unnumbered Loopback0
Router(config-if)# autoroute announce
Router(config-if)# exit
Router(config)# destination 10.10.10.11
Router(config)# path-option 1 dynamic
```

Step 5 Configure the BGP router and address families:

Example:

```
Router(config)# router bgp 100
Router(config-bgp)# bgp router-id 10.10.10.10
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp)# allocate-label all unlabeled-path
Router(config-bgp)# exit
Router(config-bgp)# address-family ipv6 unicast
Router(config-bgp)# exit
```

Step 6 Configure the BGP neighbor:

Example:

```
Router(config-bgp)# neighbor 10.0.0.1
Router(config-bgp-nbr)# remote-as 200
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family ipv4 labeled-unicast
Router(config-bgp-nbr-af)# route-policy PASS-ALL in
Router(config-bgp-nbr-af)# route-policy PASS-ALL out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# exit
```

Step 7 Configure MPLS LDP:

Example:

```
Router(config)# mpls ldp
Router(config-ldp)# router-id 10.1.1.1
Router(config-ldp)# interface tunnel-te1
Router(config-ldp)# exit
```

Step 8 Reload the router to apply the hardware module profile commands.

BGP-LU over RSVP-TE is now active, optimal tunnel paths are selected, and traffic is protected by fast reroute capabilities.

What to do next

Continue monitoring network performance and verify reroute operation during link or node failure events.

Verify BGP labeled unicast over RSVP-TE

Verify that BGP labeled unicast traffic forwards correctly through RSVP-TE tunnels. This ensures your configuration works as intended, enabling optimized traffic transport and fast reroute protection for reliable network performance.

Procedure

Step 1 Verify the configuration:

Example:

```
Router# show running configuration

Router configuration:
!
hw-module profile cef bgplu-over-rsvpte enable
!
router bgp 200
 nsr
 bgp router-id 10.1.1.1
 mpls activate
  interface Bundle-Ether10
  interface Bundle-Ether40
  interface Bundle-Ether100
  interface Bundle-Ether101
  interface HundredGigE0/0/0/22
 !
 bgp graceful-restart
 ibgp policy out enforce-modifications
 address-family ipv4 unicast
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy INSTALL_BACKUP
  network 10.1.1.5/32
  allocate-label all unlabeled-path
 !
 neighbor 10.1.4.1
  remote-as 200
  bfd fast-detect
  bfd multiplier 3
  bfd minimum-interval 100
  update-source Loopback0
  address-family ipv4 labeled-unicast
   next-hop-self
   soft-reconfiguration inbound always
   !
 neighbor 10.1.5.1
  remote-as 200
  bfd fast-detect
  bfd multiplier 3
  bfd minimum-interval 100
  update-source Loopback0
  address-family ipv4 labeled-unicast
   next-hop-self
   soft-reconfiguration inbound always
  !
  !
 neighbor 10.1.6.1
  remote-as 200
  bfd fast-detect
  bfd multiplier 3
  bfd minimum-interval 100
  address-family ipv4 labeled-unicast
   next-hop-self
   route-policy PASS-ALL in
   route-reflector-client
```

```
   route-policy PASS-ALL out
  !
Enabling LDP (to assign labels to the tunnel):

mpls ldp
 router-id 10.1.1.1
 address-family ipv4
  label
   local
    allocate for ldp-acl
   !
  !
router isis core
 is-type level-2-only
 net 49.1111.0000.0001.00
 nsr
 nsf cisco
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng igp-intact
 !
 address-family ipv6 unicast
  metric-style wide
  maximum-paths 64
 !
 interface Bundle-Ether40
  circuit-type level-2-only
  point-to-point
  address-family ipv4 unicast
   metric 10
  !
  address-family ipv6 unicast
   metric 10
  !
   interface Bundle-Ether100
  circuit-type level-2-only
  point-to-point
  address-family ipv4 unicast
   metric 10
  !
  address-family ipv6 unicast
   metric 10
  !
 interface Bundle-Ether101
  circuit-type level-2-only
  point-to-point
  address-family ipv4 unicast
   metric 10
  !
  address-family ipv6 unicast
   metric 10
  !
Tunnel Configuration:

interface tunnel-te141
 description PE1-PE4
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000000
 autoroute announce
 !
 destination 10.1.4.1
```

```
 fast-reroute
 path-protection
 !
 path-option 1 explicit name R1-R4-141
!
interface tunnel-te142
 description PE1-PE4
 ipv4 unnumbered Loopback0
 shutdown
 signalled-bandwidth 1000000
 autoroute announce
 !
 destination 10.1.4.1
 fast-reroute
 path-option 1 explicit name R1-R4-142
!
interface tunnel-te13641
 ipv4 unnumbered Loopback0
 signalled-bandwidth 1000000
 autoroute announce
 !
 destination 10.1.4.1
 path-option 1 explicit name R1-R3-R6-R4-Phy protected-by 2
 path-option 2 explicit name R1-R3-R6-R4-Bundle
!
!

mpls traffic-eng
 interface Bundle-Ether10
 !
 interface Bundle-Ether100
  backup-path tunnel-te 13641
 !
 interface Bundle-Ether101
  backup-path tunnel-te 13641
```

Step 2 Verify the details of route paths:

Example:

```
Router# show cef 209.165.200.225/27

Tue Jun  6 13:59:39.649 UTC
201.1.1.10/32, version 838761, internal 0x5000001 0x40 (ptr 0xb6848370) [1], 0x600 (0xb67bc1d8),
0xa08 (0xbbc3c0d8)
 Updated Jun  6 13:56:34.879
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
  gateway array (0xc020eac8) reference count 3, flags 0x100078, source rib (7), 0 backups
                [2 type 5 flags 0x441 (0xc1807b38) ext 0x0 (0x0)]
  LW-LDI[type=5, refc=3, ptr=0xb67bc1d8, sh-ldi=0xc1807b38]
  gateway array update type-time 1 Jun  6 13:56:34.879
 LDI Update time Jun  6 13:56:34.879
 LW-LDI-TS Jun  6 13:56:34.879
   via 10.1.4.1/32, 60047 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x97518b90 0x0]
    recursion-via-/32
    next hop 10.1.4.1/32 via 24000/0/21
     local label 36112
     next hop 10.1.4.1/32 tt141        labels imposed {ImplNull 34184}
     next hop 10.1.4.1/32 tt142        labels imposed {ImplNull 34184}
     next hop 10.1.4.1/32 tt13641      labels imposed {ImplNull 34184}
   via 10.1.5.1/32, 30045 dependencies, recursive, backup [flags 0x6100]
    path-idx 1 NHID 0x0 [0x97524fc0 0x0]
    recursion-via-/32
    next hop 10.1.5.1/32 via 24002/0/21
```

```
     local label 36112
     next hop 10.1.5.1/32 tt13651      labels imposed {ImplNull 39146}

    Load distribution: 0 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   recursive                 24000/0
```

Example:

```
Router# show route 10.1.4.1
Tue Jun  6 14:02:31.653 UTC

Routing entry for 10.1.4.1/32
  Known via "isis core", distance 115, metric 20, type level-2
  Installed Jun  6 13:59:07.013 for 00:03:24
  Routing Descriptor Blocks
    10.1.4.1, from 10.1.4.1, via tunnel-te141
      Route metric is 20
    10.1.4.1, from 10.1.4.1, via tunnel-te142
      Route metric is 20
    10.1.4.1, from 10.1.4.1, via tunnel-te13641
      Route metric is 20
  No advertising protos.
```

Example:

```
Router# show route summary
Wed May 31 17:47:01.203 UTC
Route Source                      Routes     Backup     Deleted     Memory(bytes)
connected                         536        2          0           116248
local                             539        0          0           116424
local LSPV                        1          0          0           216
local SMIAP                       1          0          0           216
application fib_mgr               0          0          0           0
static                            4          0          0           904
bgp 200                           48152      60         0           11936632
te-client                         0          0          0           0
isis core                         14056       534        0           4088288
dagr                              0          0          0           0
vxlan                             0          0          0           0
Total                             61364      596        0           16202240
```

Step 3 Verify the details of LSP tunnel:

Example:

```
Router# show mpls forwarding prefix 209.165.200.225/27
Tue Jun  6 14:00:17.601 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
36112  34184       209.165.200.225/27                10.1.4.1        0
       39146       209.165.200.225/27                10.1.5.1        0
```

Step 4 Verify the contents of the Fast Reroute (FRR) database:

Example:

```
show mpls traffic-eng fast-reroute database
Tue Jun  6 14:01:59.907 UTC
Tunnel head FRR information:
Tunnel       Out Intf : Label   FRR Intf : Label   Status
------------ ------------------ ------------------ -------
tt141        BE100:Pop          tt13641:Pop        Ready
tt142        BE101:Pop          tt13641:Pop        Ready
```

Step 5 Verify the forwarding information on tunnels:

Example:

```
Router# show mpls traffic-eng forwarding tunnel-id 141
Mon Jun  5 23:46:04.961 UTC
P2P tunnels:

Tunnel ID                 Ingress IF     Egress IF      In lbl  Out lbl        Backup
------------------------- -------------- -------------- ------- -------------- -------
10.1.1.1 141_10                        -          BE100 81920   3              tt13641
Displayed 1 tunnel heads, 0 label P2P rewrites
Displayed 0 tunnel heads, 0 label P2MP rewrites
```

Step 6 Verify the utilization of banks in the NPU resources:

Example:

```
Router# show grid pool 2 bank 13
Wed May 31 17:46:56.848 UTC

Bank Ptr                     : 0x308d069d38
Bank ID                      : 13
Pool                         : GLIF (id 2)
Bank Start                   : 530295
Bank End                     : 589823
Max Bank Size                : 59529
Max Resource Pages           : 1861
Available resource IDs       : 11375 (19.108% free)
Bank statistics:                          Success      Error         (since last clear)
  Resource IDs reserved                     51728          0           51728          0
  Resource IDs returned                      3574          0            3574          0
Client                       : lsd
  Resource IDs reserved                         2          0               2          0
  Resource IDs returned                         0          0               0          0
current usage                : 2
Client                       : rib-v4
  Resource IDs reserved                     51726          0           51726          0
  Resource IDs returned                      3574          0            3574          0
```

BGP labeled unicast version 6

BGP labeled unicast version 6 is a routing feature that

- enable MPLS transport between Provider Edge (PE) routers separated by multiple IGP boundaries (intra-AS) or autonomous systems (inter-AS)
- use autonomous system border routers (ASBRs) to advertise PE loopback prefixes and their MPLS label bindings, and
- support iBGP between area border routers (ABRs) and eBGP between ASBRs for route exchange.

Key considerations for BGP LU and VPN services across autonomous systems

Multihop eBGP can be used between PEs in different autonomous systems to exchange VPN routes. Services such as 6PE operate between PEs that have BGP Labeled Unicast connectivity.

BGP LU feature reduces scale requirements for IGP labeled prefixes and adjacencies. To prevent potential scaling issues when BGP Labeled Unicasts are not configured, you must configure the **hw-module** command before enabling BGP LU and restart the router for the changes to take effect.

BGP LU v6 extends this functionality to IPv6 networks, allowing you to apply the same MPLS transport principles over IPv6.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
BGP Labeled Unicast Version 6	Release 7.3.16	This feature extends the BGP Labeled Unicast (LU) functionality over IPv6. This feature provides connectivity between PEs to run services, such as L3VPN and 6PVE. This feature allows the PEs to transport traffic across autonomous systems (AS) boundaries. BGP LU allows you to transport MPLS traffic across IGP boundaries. By advertising loopbacks and label bindings across IGP boundaries routers communicate with other routers in remote areas that do not share the same local IGP.

Limitations for BGP labeled unicast version 6

You must understand the limitations of BGP Labeled Unicast Version 6 (BGP LU v6) to ensure proper deployment and avoid unsupported configurations that may cause network issues.

- 6VPE over BGP LU is not supported
- Inter-Address Family Identifier (AFI) is not supported.
- BGP PIC core feature is not supported.
- Coexistence of 6PE with the same neighbor is not supported.
- Coexistence of BGP LU v6 IPv6 unicast address family is not supported.
- VPNv6 over BGP LU v6 and Inter-AS Option-C with BGP LU v6 are not supported.
- Link-local addresses are not supported.
- Cases where BGP LU itself is the transport are not supported.
- Carrier Supporting Carrier version 6 is not supported.

Configure BGP labeled unicast version 6

Enable and configure BGP LU v6 on the hardware module, set up the BGP router and router ID, configure IPv6 unicast address family with route redistribution and label allocation, establish BGP neighbor and session parameters, and apply route policies for IPv6 labeled-unicast.

Use this task to enable labeled IPv6 routing with BGP LU v6, ensuring proper neighbor configuration and route policy application for efficient labeled-unicast forwarding.

Before you begin

Verify hardware module supports BGP LU and that you have the necessary router and neighbor information.

Follow these steps to enable and configure BGP LU v6:

Procedure

Step 1 Enable BGP labeled unicast on the hardware module

Example:

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

Step 2 Configure the BGP router and router ID:

Example:

```
Router(config)# router bgp 1
Router(config-bgp)# bgp router-id 2001:DB8::1
```

Step 3 Configure the IPv6 unicast address family and redistribute connected routes with label allocation:

Example:

```
Router(config-bgp)# address-family ipv6 unicast
Router(config-bgp-af)# redistribute connected route-policy set-lbl-idx
Router(config-bgp-af)# allocate-label all
Router(config-bgp-af)# exit
```

Step 4 Configure the BGP neighbor and session parameters:

Example:

```
Router(config-bgp)# neighbor 2001:DB8::2
Router(config-bgp)# remote-as 1
Router(config-bgp)# update-source Loopback 0
```

Step 5 Configure the IPv6 labeled-unicast address family and apply route policies:

Example:

```
Router(config-bgp)# address-family ipv6 labeled-unicast
Router(config-bgp)# route-policy pass-all in
Router(config-bgp)# route-policy pass-all out
Router(config-bgp)# commit
```

Step 6 Verify the configuration:

Example:

```
Router# show running-config
hw-module profile cef bgplu enable
router bgp 1
 bgp router-id 2001:DB8::1
 address-family ipv6 unicast
  redistribute connected route-policy set-lbl-idx
  allocate-label all
  exit
 neighbor 2001:DB8::2
 remote-as 1
 update-source Loopback 0
  address-family ipv6 labeled-unicast
```

```
   route-policy pass-all in
   route-policy pass-all out
```

Step 7 Use the **show hw-module profile cef** to verify that the BGP LU has been configured.

Example:

```
Router# show hw-module profile cef
Thu Jun 17 00:06:32.974 UTC
-------------------------------------------------------------------------------------
Knob                          Status        Applied   Action
-------------------------------------------------------------------------------------
BGPLU                        Configured      Yes       None
LPTS ACL                     Unconfigured    Yes       None
Dark Bandwidth               Unconfigured    Yes       None
MPLS Per Path Stats          Unconfigured    Yes       None
Tunnel TTL Decrement         Unconfigured    Yes       None
High-Scale No-LDP-Over-TE    Unconfigured    Yes       None
IPv6 Hop-limit Punt          Unconfigured    Yes       None
IP Redirect Punt             Unconfigured    Yes       None
```

Step 8 Use the **show cef ipv6** command to verify the details of route paths along with the BGP and transport label information.

Example:

```
Router# show cef ipv6 192:168:9::80/128

Wed Jun 16 07:42:04.789 UTC
192:168:9::80/128, version 27, internal 0x5000001 0x40 (ptr 0x93f2d478) [1], 0x0 (0x93ef6cc0), 0xa08
 (0x9460a8a8)
 Updated Jun 16 07:36:00.189
 Prefix Len 128, traffic index 0, precedence n/a, priority 4, encap-id 0x1001000000001
   via 10:0:1::51/128, 3 dependencies, recursive [flags 0x6000]
    path-idx 0 NHID 0x0 [0x94720660 0x0]
    recursion-via-/128
    next hop 10:0:1::51/128 via 16061/0/21
     next hop fe80::7af8:c2ff:fee4:20c0/128 Hu0/0/0/27   labels imposed {16061 25001}
/*
16061 - Transport Label
25001 – BGP Label
*/
```

Step 9 Use the **show bgp ipv6 unicast labels** to verify the BGP LU version 6 routes and BGP label information in BGP process.

Example:

```
Router# show bgp ipv6 unicast labels
Wed Jun 16 07:34:58.968 UTC
BGP router identifier 10.0.1.50, local AS number 1
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0xe0800000   RD version: 6
BGP main routing table version 6
BGP NSR Initial initsync version 3 (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        Rcvd Label      Local Label
*> 192:168::/64        192:168:1::70   nolabel         24006
*>i192:168:9::80/128   10:0:1::51      25001           nolabel
```

```
Processed 2 prefixes, 2 paths
```

BGP LU v6 is enabled and configured, allowing labeled IPv6 routing with proper neighbor and route policy settings.

What to do next

Reload the router for the **hw-module profile cef bgplu enable** command to take effect.

BGP labeled unicast MPLS IP POP

BGP Labeled Unicast MPLS IP POP is a routing feature that

- enables efficient forwarding of IPv4 unicast traffic using labeled routes
- uses BGP to distribute labeled prefixes with implicit NULL labels, and
- applies only the transport LDP label on packets before forwarding them into the MPLS core.

This mechanism optimizes label stacking and improves forwarding efficiency in MPLS networks.

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
BGP Labeled Unicast MPLS IP POP Support	Release 7.3.1	This feature is based on the BGP labeled Unicast feature. This feature enables a router to send unicast traffic to the destination from BGP labeled unicast using implicit NULL label. Implicit null label avoids adding or removing rewrites for neighbor flaps.

BGP labeled unicast with implicit NULL label and transport LDP label forwarding in MPLS core networks

- IP POP Support allows you to forward IP packets with minimal label stacking, improving network efficiency.
- For example, in a topology where client A sends IPv4 unicast traffic to client B, IP POP Support enables the Provider Edge (PE) router to add only the transport LDP label on top of the IP packet before forwarding it into the MPLS core.
- This reduces complexity compared to traditional MPLS forwarding, which may use multiple stacked labels.
- Understanding IP POP Support helps you design and troubleshoot MPLS networks that optimize label usage and forwarding performance.

Configure BGP labeled unicast on MPLS IP pop Support

Enable and configure BGP labeled unicast on PE1 and PE3 to optimize IPv4 labeled-unicast forwarding by reducing label overhead and simplifying MPLS label stacking.

This configuration allows PE1 to learn destination prefixes from PE3 via BGP labeled unicast using implicit NULL labels. PE1 then applies only the transport LDP label on IPv4 packets before forwarding them into the MPLS core, enhancing forwarding efficiency.

Before you begin

- Ensure hardware modules on PE1 and PE3 support BGP labeled unicast and the hardware module profile can be enabled.
- Confirm PE1 and PE3 have proper connectivity and reachability, including configured loopback interfaces for BGP update sources.
- Make sure BGP is operational on both routers with correct autonomous system numbers.

Configure PE1:

Procedure

Step 1 Enable BGP labeled unicast on the hardware module:

Example:

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

Step 2 Configure BGP router and enable features

Example:

```
Router(config)# router bgp 200
Router(config-bgp)# nsr
Router(config-bgp)# bgp router-id 192.168.70.24
Router(config-bgp)# bgp graceful-restart
Router(config-bgp)# ibgp policy out enforce-modifications
```

Step 3 Configure IPv4 unicast address family with maximum paths and label allocation:

Example:

```
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ibgp 8
Router(config-bgp-af)# network 101.101.1.0/24
Router(config-bgp-af)# network 101.101.2.0/24
Router(config-bgp-af)# allocate-label all
Router(config-bgp-af)# exit
```

Step 4 Configure BGP neighbor and session parameters:

Example:

```
Router(config-bgp)# neighbor 10.3.3.3
Router(config-bgp)# remote-as 200
Router(config-bgp)# update-source Loopback0
```

Step 5 Configure IPv4 labeled unicast address family and set the next-hop:

Example:

```
Router(config-bgp)# address-family ipv4 labeled-unicast
Router(config-bgp)# next-hop self
```

Step 6 Verify the configuration on the PE1 router. The output should show the bgplu profile enabled and correct BGP labeled unicast settings.

Example:

```
Router# show running-config
hw-module profile bgplu enable
router bgp 200
 nsr
 bgp router-id 192.168.70.24
 bgp graceful-restart
 ibgp policy out enforce-modifications
 address-family ipv4 unicast
  maximum-paths ibgp 8
  network 101.101.1.0/24
  network 101.101.2.0/24
  allocate-label all
!
 neighbor 10.3.3.3
 remote-as 200
 update-source Loopback0
 address-family ipv4 labeled-unicast
 next-hop self
```

Step 7 Verify the configuration on the PE2 router. The output should show the bgplu profile enabled and correct BGP labeled unicast settings.

Example:

```
Router# show running-config
hw-module profile bgplu enable
router bgp 200
 nsr
 bgp router-id 192.168.70.25
 bgp graceful-restart
 ibgp policy out enforce-modifications
 address-family ipv4 unicast
  maximum-paths ibgp 8
  network 103.101.1.0/24
  network 103.101.2.0/24
  allocate-label all
!
neighbor 10.1.1.1
remote-as 1
update-source Loopback0
address-family ipv4 labeled-unicast
next-hop self
```

Step 8 Verify if the feature has been configured.

Example:

```
Router# show bgp ipv4 unicast labels
  Network          Next Hop    Rcvd Label      Local Label
*>i103.101.1.0/24   10.3.3.3         3            24006
```

Step 9 Restart the router so hardware module configuration takes effect.

PE routers use BGP labeled unicast with implicit NULL labels and apply only the transport LDP label before forwarding IPv4 packets into the MPLS core, resulting in improved forwarding efficiency.

What to do next

Monitor BGP and MPLS operations to ensure stable label distribution and end-to-end forwarding.

Convergence for BGP labeled unicast PIC edge

A convergence for BGP labeled unicast PIC edge is a routing resiliency feature that

- allows you to store both primary and backup paths in the Routing Information Base (RIB), Forwarding Information Base (FIB), and Cisco Express Forwarding,
- enables the router to switch immediately to the backup path when a failure is detected, and
- provides subsecond convergence and fast failover for BGP-labeled prefixes.

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Convergence for BGP Labeled Unicast PIC Edge	Release 7.7.1	This feature improves the convergence time of BGP labeled unicast (LU) routes to subseconds when an ingress provider edge router fails or loses PE router connectivity, and another PE router needs to be connected. This feature minimizes traffic drops when the primary paths fail for the BGP LU routes.

Table 6: BGP LU PIC edge vs. standard BGP LU

	BGP LU PIC edge	Standard BGP LU
Backup path preprogrammed	Yes	No
Failover speed	Subsecond	Slower, may take seconds
Service disruption	Minimal	Possible disruption

Note You must ensure that your edge iBGP devices, such as ingress PEs and ASBRs, support BGP PIC and receive backup BGP next hops for PIC edge features to function.

How BGP labeled unicast PIC edge ensures fast convergence

This process applies to provider networks using BGP labeled unicast (LU) prefix independent convergence (PIC) edge to ensure rapid failover and maintain uninterrupted traffic during path or router failures.

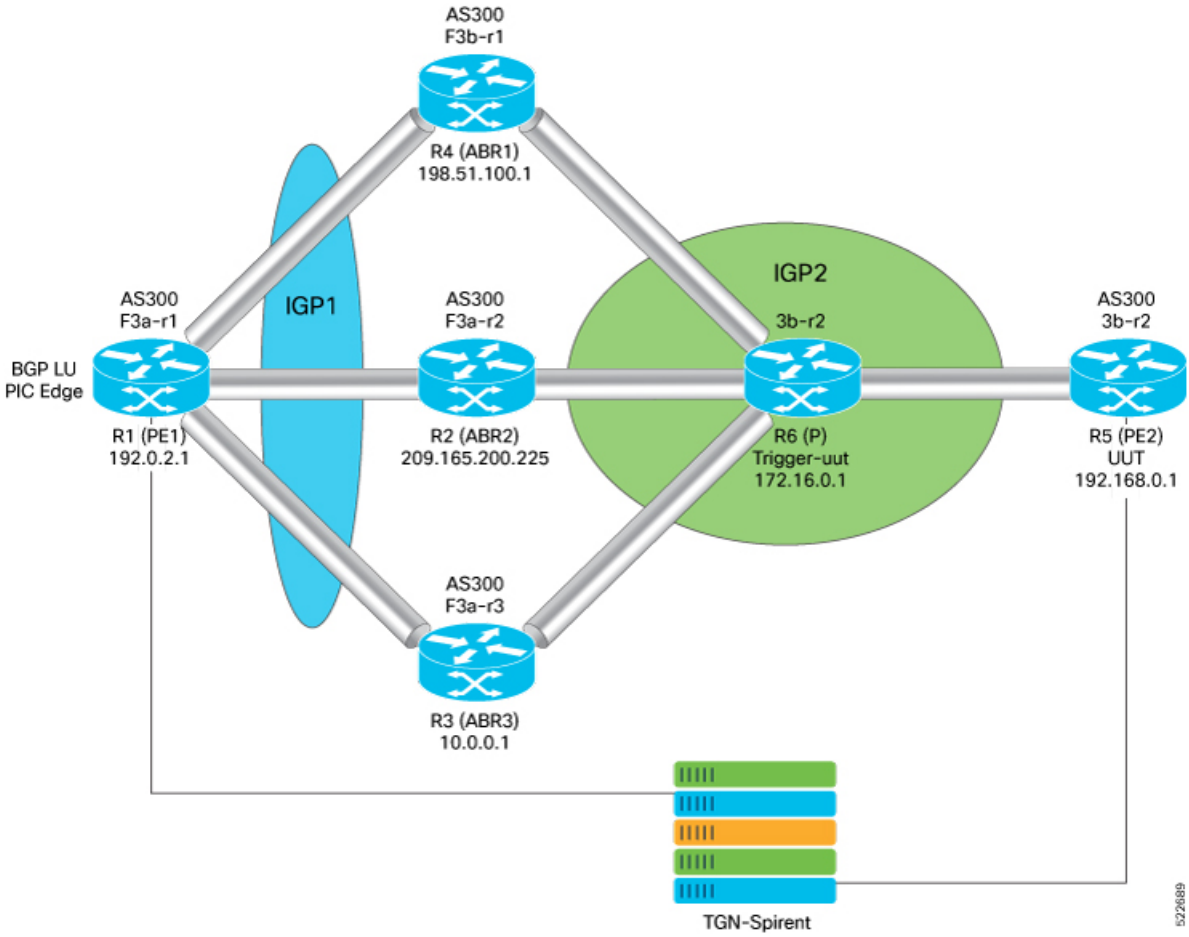
Summary

- PE1 (Provider Edge Router): Enables BGP LU PIC edge, learns BGP LU prefixes, programs primary and backup paths, detects failures, and switches traffic to the backup path when needed.
- PE2 (Remote Provider Edge Router): Provides BGP LU prefixes to PE1 and serves as a backup or alternate next hop.
- ABR1, ABR2, ABR3 (Area Border Routers): Serve as primary and backup transit paths for traffic between PE1 and PE2; if one ABR fails, another ABR takes over as the active path.

This process enables rapid failover in provider networks using BGP LU PIC edge by preprogramming primary and backup paths, ensuring fast, automatic rerouting and uninterrupted traffic during failures.

Workflow

Figure 5: BGP labeled unicast prefix independent convergence edge



These stages describe how BGP labeled unicast (LU) PIC edge enables fast convergence and failover in the provider network work.

1. Enable BGP LU PIC edge on the provider edge router (PE1).
2. PE1 learns BGP LU prefixes from the remote provider edge router (PE2).

3. PE1 programs both the primary and backup paths through area border routers (ABR1, ABR2, and ABR3) into the forwarding information base (FIB).
4. Program PE2 as the backup or alternate next hop in the FIB of PE1.
5. Under normal conditions, PE1 forwards traffic to the customer edge (CE) device through the primary ABR.
6. If the primary ABR fails, PE1 immediately activates the preprogrammed backup path and forwards traffic through an alternate ABR.
7. When PE1 detects that PE2 is not reachable via the primary ABR, it switches the BGP next hop for the prefix to the alternate ABR without delay.
8. The switchover to the backup path occurs in less than a second, even if multiple BGP prefixes are updated at once.

Result

This process ensures subsecond convergence and uninterrupted traffic flow during ABR or path failures in the network.

Supported features and limitations of BGP LU PIC edge

Supported features:

- BGP LU PIC edge supports BGP multipaths, allowing routers to install multiple internal and external BGP paths in the forwarding table.
- Multiple paths enable BGP to load balance traffic across several links.

Limitation::

- Convergence time is independent of the BGP LU route scale.

Configure convergence for BGP labeled unicast prefix independent convergence edge

Configure BGP LU PIC edge and multipath to enable fast convergence and backup path installation.

Use this task when you need to configure BGP labeled unicast PIC edge and multipath on Cisco routers to ensure fast convergence and automatic traffic failover in service provider or large enterprise networks.

Before you begin

- Ensure you have access to the router prompt in configuration mode.
- Verify that BGP is enabled and address families are configured as needed.

Procedure

Step 1 Configure BGP labeled unicast and attach a route-policy to BGP address families.

Example:

Apply the route policy to the BGP address family.

```
Router(config)# route-policy BGP-PIC-EDGE
Router(config-rpl)# set path-selection backup 1 install
Router(config-rpl)# end-policy
```

Apply the route policy to the BGP address family.

```
Router(config)# router bgp <ASN>
Router(config-bgp)# bgp router-id <Router-ID>
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# additional-paths receive
Router(config-bgp-af)# additional-paths send
Router(config-bgp-af)# additional-paths selection route-policy BGP-PIC-EDGE
```

Step 2 Configure BGP labeled unicast multipath and attach a route-policy to BGP address families:

Example:

Define the multipath route policy.

```
Router(config)# route-policy BGP-PIC-EDGE-MULTIPATH
Router(config-rpl)# set path-selection backup 1 install multipath-protect
Router(config-rpl)# end-policy
```

Apply the multipath route policy to the BGP address family.

```
Router(config)# router bgp <ASN>
Router(config-bgp)# bgp router-id <Router-ID>
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ibgp 2
Router(config-bgp-af)# additional-paths receive
Router(config-bgp-af)# additional-paths send
Router(config-bgp-af)# additional-paths selection route-policy BGP-PIC-EDGE-MULTIPATH
```

Step 3 Use the **show running-config** to verify if the feature has been configured.

Example:

```
Router# show running-config
route-policy BGP-PIC-EDGE
 set path-selection backup 1 install
 end-policy
router bgp 200
 bgp router-id 192.168.1.0
 address-family ipv4 unicast
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy BGP-PIC-EDGE

route-policy BGP-PIC-EDGE-MULTIPATH
 set path-selection backup 1 install multipath-protect
 end-policy
router bgp 200
 bgp router-id 192.168.1.0
 address-family ipv4 unicast
  maximum-paths ibgp 2
  additional-paths receive
  additional-paths send
  additional-paths selection route-policy BGP-PIC-EDGE-MULTIPATH
```

Step 4 Use the **show cef** to verify that the backup path is established.

Example:

```
Router# show cef 192.0.2.1/32

192.168.0.0/32, version 31, internal 0x5000001 0x40 (ptr 0x901d2370) [1], 0x0 (0x90d2beb8), 0xa08
(0x91c74378)
 Prefix Len 32, traffic index 0, precedence n/a, priority 4
   via 203.0.113.1/32, 3 dependencies, recursive [flags 0x6000]  << Primary Path
    path-idx 0 NHID 0x0 [0x90319650 0x0]
    recursion-via-/32
    next hop 192.51.100.1/32 via 24006/0/21
    next hop 209.165.200.225/32 Hu0/0/0/25   labels imposed {24002 24000}
    next hop 10.0.0.1/32 Hu0/0/0/26   labels imposed {24002 24000}
   via 203.0.113.2/32, 2 dependencies, recursive, backup [flags 0x6100]  << Backup Path
    path-idx 1 NHID 0x0 [0x903197b8 0x0]
    recursion-via-/32
    next hop 209.165.200.225/32 via 24005/0/21
    next hop 192.51.100.1/32 Hu0/0/0/25   labels imposed {24001 24000}
    next hop 10.0.0.1/32 Hu0/0/0/26   labels imposed {24001 24000}
```

BGP labeled unicast PIC edge and multipath are configured, enabling the router to install backup paths and achieve fast, automatic convergence and failover during link or path failures.

Exclusion of label allocation for non-advertised routes

An exclusion of label allocation for non-advertised routes is a label management feature that

- allows control of label allocation based on route advertisement status
- enables assignment of labels only to routes that are advertised to BGP peers, and
- prevents label allocation for routes that are not advertised.

Route Advertisement Control Using Community Attributes on PE-ASBRs

- This feature is configured on Provider Edge (PE) routers acting as autonomous system border routers (ASBRs).

  The community attribute is set to either no-advertise or no-export in route-policy configuration mode for routes that are not advertised to peers.
- After applying the community attribute in the route policy, the router does not allocate a label to those routes.

Guidelines for applying BGP community attributes

Apply the BGP community attributes as follows:

- Use `no-advertise` for both eBGP and iBGP routes.
- Use `no-export` only for eBGP routes.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Exclusion of Label Allocation for Non-Advertised Routes	Release 7.10.1	We have enabled better label space management and hardware resource utilization by making MPLS label allocation more flexible. This flexibility means you can now assign these labels to only those routes that are advertised to their peer routes, ensuring better label space management and hardware resource utilization. Prior to this release, label allocation was done regardless of whether the routes being advertised. This resulted in inefficient use of label space.

Exclude label allocation for non-advertised routes

Prevent label allocation to routes that are not advertised to any BGP peer by using the no-advertise community.

Use this task when you need to configure BGP on a Provider Edge (PE) router so that non-advertised routes do not receive labels.

Before you begin

- Verify that BGP is already configured on the device.
- Identify the BGP neighbors and address families that require this configuration.
- Plan the route policies and community sets you need to implement.

\

Procedure

Step 1 Configure the community set for no-advertise.

Example:

```
Router(config)# community-set no-advertise
Router(config-comm)# no-advertise
Router(config-comm)# end-set
```

Step 2 Configure a route policy to set the no-advertise community.

Example:

```
Router(config)# route-policy set-no-advertise
Router(config-rpl)# set community no-advertise additive
Router(config-rpl)# end-policy
```

Step 3 Apply the route policy as an inbound policy to the BGP neighbor.

Example:

```
Router(config)# router bgp 1
Router(config-bgp)# neighbor 192.0.2.1
Router(config-bgp-nbr)# remote-as 1
Router(config-bgp-nbr)# update-source Loopback0
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy set-no-advertise in
Router(config-bgp-nbr-af)# route-policy pass_all out
Router(config-bgp-nbr-af)# commit
```

Step 4 Use the **show bgp vpnv6 unicast rd** command to verify if the community parameter is set to *no-advertise*.

Example:

```
Router# show bgp vpnv6 unicast rd 2001:DB8:0:ABCD::1
BGP routing table entry for 0:ABCD::1 Route Distinguisher: 2001:DB8
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker               19207        19207
Paths: (1 available, best #1, not advertised to any peer)
  Not advertised to any peer
  Path #1: Received by speaker 0
  Not advertised to any peer
  Local, (Received from a RR-client)
    192.0.2.254 from 192.0.2.1 (192.0.2.1)
      Received Label 16
      Origin IGP, metric 3, localpref 3, aigp metric 3, valid, internal, best, group-best,
import-candidate, not-in-vrf
      Received Path ID 0, Local Path ID 1, version 19207
Community: 1:1 no-advertise
      Extended community: Color:3333 RT:2001:DB8
      AIGP set by inbound policy metric
      Total AIGP metric 3
```

Steering of BGP control-plane traffic over IP paths

Steering of BGP control-plane traffic over IP paths is a traffic engineering feature that

- allows selection of an IP-only transport path for BGP control-plane traffic instead of using the default MPLS LSP
- separates BGP control-plane traffic from labeled and regular IP traffic, and
- reduces complexity and risk by isolating BGP session traffic from MPLS transport paths.

Steering BGP control-plane traffic over IP-only paths in MPLS networks

In a typical underlay network, the transport label-switched path (LSP) is established using MPLS protocols such as Segment Routing MPLS, Label Distribution Protocol (LDP), or Service Layer API. By default, the transport LSP carries all traffic—including labeled packets, IP packets, and BGP control-plane traffic—toward the underlay destination. Routing BGP control-plane traffic over MPLS LSPs can introduce operational complexity and risk, potentially leading to network instability.

With the steering feature, you can configure BGP control-plane traffic to use an IP-only path created by the IS-IS protocol. The MPLS path continues to determine BGP next hops for data-plane traffic, while the IP-only path is used exclusively for BGP control-plane packets.

Before you enable this feature, you create a new VRF to manage IP-only routing tables. After configuration, IS-IS generates an IP-only route entry in the Routing Information Base (RIB), which is then downloaded to the Forwarding Information Base (FIB) in the VRF. This separate VRF topology allows the router to resolve locally generated BGP control-plane traffic independently from the MPLS transport.

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Steering of BGP Control-Plane Traffic over IP Path	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 88-LC1-36EH+A8:B12 • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 8212-48FH-M • 8711-32FH-M
Steering of BGP Control-Plane Traffic over IP Path	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100]) This feature support is now extended to the Cisco 8712-MOD-M routers.

Steering of BGP Control-Plane Traffic over IP Path	Release 24.2.11	You can now steer the BGP control-plane traffic through an IP-only transport path even when MPLS Link State Packets (LSPs) are configured for BGP neighbor reachability. This feature allows you to keep the BGP control-plane traffic independent of the data plane traffic, enabling you to have more granular control over your network traffic. The feature introduces these changes: CLI: New Commands: • **table ip-only activate vrf** • **tcp ip-only-preferred** Modified Commands: • The **distribute-list** command is modified with a new **ip-only** keyword. YANG Data Models: New XPaths for • Cisco-IOS-XR-clns-isis-cfg.yang • Cisco-IOS-XR-ipv4-bgp-cfg.yang • Cisco-IOS-XR-ip-rib-cfg.yang • Cisco-IOS-XR-um-router-bgp-cfg.yang • Cisco-IOS-XR-um-router-isis-cfg.yang (see GitHub, YANG Data Models Navigator)

Configure the router to steer BGP control-plane traffic over an IP-only path

Configure the router to direct BGP control-plane traffic over an IS-IS IP-only path instead of the default MPLS LSP.

Use this task when you want to separate BGP control-plane traffic from MPLS transport paths in an underlay network. Steering BGP control-plane packets over an IP-only path can help reduce complexity and improve the stability of BGP sessions in large-scale or high-availability environments.

Before you begin

- Confirm that BGP and IS-IS are enabled and configured.
- Identify the ASN, remote AS, loopback interface, and process IDs you need.

Procedure

Step 1 Configure a VRF for the IP-only path.

Example:

```
Router(config)# vrf ip_only
Router(config-vrf)# fallback-vrf default
Router(config-vrf)# address-family ipv4 unicast
Router(config-vrf-af)# exit
Router(config-vrf)# address-family ipv6 unicast
Router(config-vrf-af)# exit
```

Step 2 Activate the IP-only table in RIB configuration.

Example:

```
Router(config)# router rib
Router(config-rib)# table ip-only activate vrf ip_only
```

Step 3 Configure the BGP neighbor group to use IP-only steering.

Example:

```
Router(config)# router bgp <ASN>
Router(config-bgp)# neighbor-group ip-only
Router(config-bgp-nbrgrp)# remote-as <Remote-AS>
Router(config-bgp-nbrgrp)# update-source <Loopback-Interface>
Router(config-bgp-nbrgrp)# tcp ip-only-preferred
```

Step 4 (Optional) Configure prefix-list and distribute-list for IS-IS.

Example:

```
Router(config)# ipv4 prefix-list v4-host-only
Router(config-ipv4_pfx)# 10 permit 0.0.0.0/0 eq 32
Router(config-ipv4_pfx)# exit
Router(config)# router isis 1
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# distribute-list ip-only prefix-list v4-host-only in
```

Step 5 (Optional) Configure a route-policy for IP-only steering.

Example:

```
Router(config)# route-policy rpl-isis-ip-only
Router(config-rpl)# if not destination in (192.0.2.1 192.0.2.2 192.0.2.3) then
Router(config-rpl-if)# drop
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl)# end-policy
Router(config)# router isis 1
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# distribute-list ip-only route-policy isis-ip-only in
```

Step 6 Use the **show running-config router rib** command to verify if the feature is enabled.

Example:

```
Router# show running-config router rib
Wed Mar 27 06:39:01.233 UTC
router rib
 table ip-only activate vrf ip_only
!
```

Step 7 Verify the IS-IS IP-only local RIB entries:

Example:

```
 Router# show isis route ip-only

Wed Jul 26 09:24:56.422 PDT
```

```
IS-IS 1 IPv4 Unicast routes

Codes: L1 - level 1, L2 - level 2, ia - interarea (leaked into level 1)
       df - level 1 default (closest attached router), su - summary null
       C - connected, S - static, R - RIP, B - BGP, O - OSPF
       E - EIGRP, A - access/subscriber, M - mobile, a - application
       i - IS-IS (redistributed from another instance)

Maximum parallel path count: 8

L2 10.2.1.0/24 [20/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
L2 10.3.1.0/24 [120/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
L2 10.4.1.0/24 [130/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
L2 10.1.0.101/32 [20/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
L2 10.1.0.102/32 [30/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
L2 10.1.0.103/32 [130/115]
     via 10.1.1.101, GigabitEthernet0/0/0/2, r101, Weight: 0
```

Step 8 Use the **show tcp detail pcb** command to verify that BGP is using the IP-only option and check the TCP session details for the neighbor.

Example:

```
Router# show tcp detail pcb 0x00007f733000d618 location 0/rP1/CPU0

Tue Dec 12 09:20:56.163 UTC

================================================================
Connection state is ESTAB, I/O status: 0, socket status: 0
Established at Tue Dec 12 07:25:24 2023

PCB 0x00007f733000d618, SO 0x7f733000d158, TCPCB 0x7f733000d8c8, vrfid 0x60000000,
Pak Prio: Medium, TOS: 192, TTL: 255, Hash index: 1575
Local host: 10.1.1.1, Local port: 179 (Local App PID: 24619)
Foreign host: 10.4.4.4, Foreign port: 50026
(Local App PID/instance/SPL_APP_ID: 24619/1/0)

Current send queue size in bytes: 0 (max 24576)
Current receive queue size in bytes: 0 (max 32768)  mis-ordered: 0 bytes
Current receive queue size in packets: 0 (max 0)

   Timer          Starts    Wakeups         Next(msec)
Retrans           1735         0               0
SendWnd             0          0               0
TimeWait            0          0               0
AckHold           1733      1668               0
KeepAlive           0          0               0
PmtuAger            0          0               0
GiveUp              0          0               0
Throttle            0          0               0
FirstSyn            0          0               0

   iss: 2670304720  snduna: 2670348690  sndnxt: 2670348690
sndmax: 2670348690  sndwnd: 32768       sndcwnd: 3720
   irs: 2277543107  rcvnxt: 2277587077  rcvwnd: 32331   rcvadv: 2277619845

SRTT: 232 ms,  RTTO: 300 ms,  RTV: 7 ms,  KRTT: 0 ms
minRTT: 0 ms,  maxRTT: 248 ms
```

```
ACK hold time: 200 ms, Keepalive time: 0 sec, SYN waittime: 30 sec
Giveup time: 0 ms, Retransmission retries: 0, Retransmit forever: FALSE
Connect retries remaining: 0, connect retry interval: 0 secs

State flags: none
Feature flags: Win Scale, Nagle, IP FIB TBLID OVERRIDE
Request flags: Win Scale

Datagrams (in bytes): MSS 1240, peer MSS 1240, min MSS 1240, max MSS 1240

Window scales: rcv 0, snd 0, request rcv 0, request snd 0
Timestamp option: recent 0, recent age 0, last ACK sent 0
Sack blocks {start, end}: none
Sack holes {start, end, dups, rxmit}: none
Socket options: SO_REUSEADDR, SO_REUSEPORT, SO_NBIO
Socket states: SS_ISCONNECTED, SS_PRIV, SS_BLOCKCLOSE, SS_BLOCKSND
Socket receive buffer states: SB_DEL_WAKEUP
Socket send buffer states: SB_DEL_WAKEUP
Socket receive buffer: Low/High watermark 1/32768
Socket send buffer   : Low/High watermark 2048/24576, Notify threshold 0
Socket misc info     : Rcv data size (sb_cc) 0, so_qlen 0,
                       so_q0len 0, so_qlimit 0, so_error 0
                       so_auto_rearm 1

PDU information:
 #PDU's in buffer: 0

FIB Lookup Cache:
  Lookup table: default ipv4 unicast (Table ID: 0xe0000001)
  Lookup done at Tue Dec 12 09:16:24 2023 (next lookup due on next protocol message on or after 78
sec)

  Lookup result:
    Matching table: default ipv4 unicast (Table ID: 0xe0000001)
    Outgoing interface: Bundle-Ether1 (IFH: 0xf000024)
    PD ctx:  size: 0 data: {}
    Num Labels: 0  Label Stack: {}
    Next HopID: 0
    VXLAN Encap String size: 0 data:
    VXLAN Next Hop IP size: 0 IP:

Num of peers with authentication info: 0
```

Step 9 Use the **show tcp statistics pcb** command to verify the number of IP-only packets per neighbor:

Example:

```
Router# show tcp statistics pcb 0x00007f733000d618 location 0/rP1/CPU0

Wed Mar 27 06:46:52.566 UTC

================================================================
 Statistics for PCB 0x7f1ca0008550, vrfid 0x60000000
Send:   0 bytes received from application
        0 segment instructions received from partner
        0 xipc pulses received from application
        0 packets sent to network (v4/v6 IO)
        3547 packets sent to network (NetIO)
        0 packets failed getting queued to network (v4/v6 IO)
        0 packets failed getting queued to network (NetIO)
        3217 ip-only-preferred packets sent to network
        0 write operations by application
        0 times armed, 0 times unarmed, 0 times auto-armed
        Last written at: Wed Mar 27 06:46:51 2024
```

```
Rcvd:   3584 packets received from network
        1791 packets queued to application
        1 packets failed queuing to application
        0 packets dropped due to minttl check
        0 send-window shrink attempts by peer ignored
        0 read operations by application
        0 times armed, 0 times unarmed, 0 times auto-armed
        Last read at: Wed Mar 27 06:46:51 2024
```

Verify the BGP control-plane IP-only steering configuration

Confirm that the router is configured to steer BGP control-plane traffic over an IP-only path, separating BGP control-plane traffic from MPLS transport.

Use this task to verify that your router's running configuration supports BGP control-plane IP-only steering using IS-IS and VRF-based routing..

Before you begin

Before you verify the BGP control-plane IP-only steering configuration, ensure the following:

- You have administrative or enable-level access to the router.
- Prefix lists, distribute lists, and route policies for IP-only steering have been configured as intended.
- BGP is enabled and properly configured, including neighbor groups and update sources.

Follow these steps to verify BGP control-plane IP-only steering configuration:

Procedure

Step 1 Use the **show rib tables** command to verify the status and details of the RIB tables on the router

Example:

```
Router# show rib tables
Wed Mar 27 06:39:58.319 UTC

Codes: N - Prefix Limit Notified, F - Forward Referenced
       D - Table Deleted, C - Table Reached Convergence

VRF/Table              SAFI  Table ID     PrfxLmt   PrfxCnt TblVersion  N F D C
default/default        uni   0xe0000000  10000000        21         43  N N N Y
ip_only/default        uni   0xe0000001  10000000        10         42  N N N Y
default-ip-only/defau  uni   0xe0000002  10000000         0          0  N N N Y
**iid/default          uni   0xe00007d9  10000000         0          0  N N N Y
default/default        multi 0xe0100000  10000000         0          0  N N N Y
```

Step 2 Use **show isis rib tables** command to verify the IS-IS routing tables present on the router.

Example:

```
Router# show isis rib tables
Wed Mar 27 06:40:58.587 UTC

IS-IS 100 Routing Tables
```

```
  ISIS routes       VRF/Table                SAFI   Table ID     State

IPv4 Unicast:
  default           default/default          uni    0xe0000000   enabled
  ip-only           ip_only/default          uni    0xe0000001   enabled
  multicast-intact  default/default          uni    0xe0100000   enabled

IPv6 Unicast:
  default           default/default          uni    0xe0800000   enabled
  ip-only           ip_only/default          uni    0xe0800001   enabled
  srv6              default/default          uni    0xe0800000   enabled
```

Step 3 Use the **show running-config** to display the running configuration.

Example:

```
Router# show running-config
vrf ip_only
 fallback-vrf default
 address-family ipv4 unicast
 !
 address-family ipv6 unicast
 !
!
router rib
      table ip-only activate vrf ip_only
!
router bgp 140
 neighbor-group ip_only
  remote-as 100
  update-source Loopback99
  tcp ip-only-preferred
!
ipv4 prefix-list v4-host-only
  10 permit 0.0.0.0/0 eq 32
!
router isis 1
  address-family ipv4 unicast
 distribute-list ip-only prefix-list v4-host-only in
!
route-policy rpl-isis-ip-only
if not destination in (192.0.2.1 192.0.2.2 192.0.2.3) then
  drop
else
  pass
end-policy
!
router isis 1
  address-family ipv4 unicast
    distribute-list ip-only route-policy isis-ip-only in
!
!
```

Review the output to ensure the following configuration elements are present:

- A VRF named `ip_only` with appropriate address families.
- The RIB is configured to activate the `ip_only` VRF.
- The BGP neighbor group includes the `tcp ip-only-preferred` setting.
- Prefix-list and distribute-list settings for IS-IS, if required.

- Any custom route-policies for IP-only steering.

You have confirmed that BGP control-plane traffic is set to use an IP-only path, based on the elements present in the running configuration.