

BGP Route Reflect

This chapter provides conceptual and configuration information to enable Border Gateway Protocol Route Reflect (BGP RR) on Cisco NCS 4000 Series routers.

Table 1: Feature History

Feature Name	Release Information	Feature Description
BGP VPNv6 Route Reflect support	Cisco IOS XR Release 6.5.35	NCS 4000 devices now support Route Reflector functionality for VPNv6 prefixes, reflecting routes without participating in data forwarding. Acting as a P-node, it uses IPv4 MPLS label switching to carry VPNv6 traffic, enabling scalable, hardware-compatible deployments with reduced control-plane state for large service provider cores. A new CLI keyword is added within the address-family configuration for BGP: • **vpnv6 unicast**
BGP scale	Cisco IOS XR Release 6.5.31	BGP labeled unicast (BGP LU) supports 500 scale sessions with 8000 prefixes.

- BGP Route Reflectors, on page 2
- Table Policy, on page 5
- BGP Keychains, on page 5
- Configure a Route Reflector for BGP, on page 6
- Applying Table Policy, on page 8
- Configuring BGP Route Reflect Filtering by Table Policy, on page 9
- Verifying BGP, on page 11
- BGP Labeled Unicast, on page 12

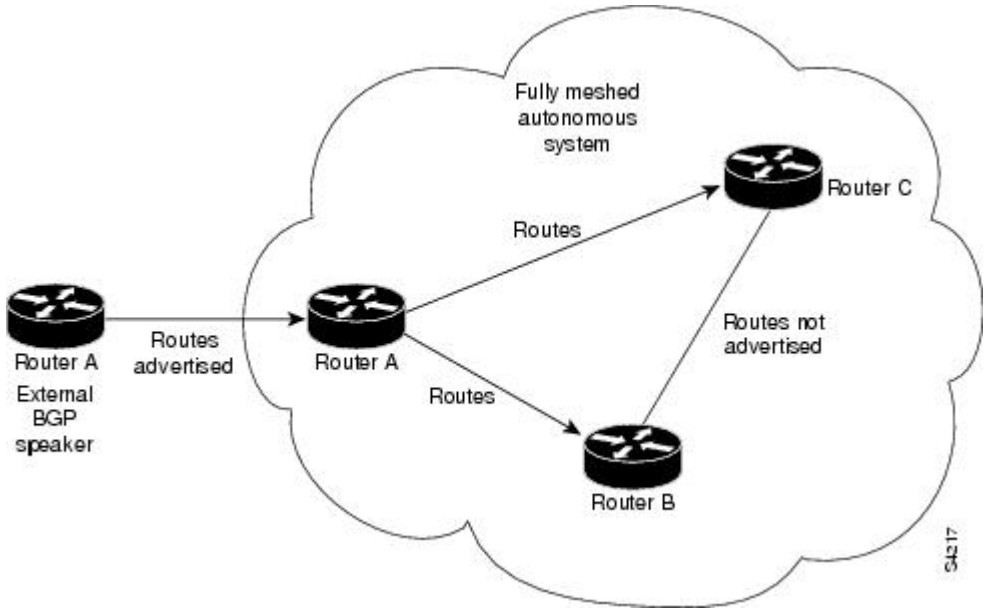
• BGP LU and PIC Configuration, on page 14

BGP Route Reflectors

BGP requires that all iBGP speakers be fully meshed. However, this requirement does not scale well when there are many iBGP speakers. Instead of configuring a confederation, you can reduce the iBGP mesh by using a route reflector configuration.

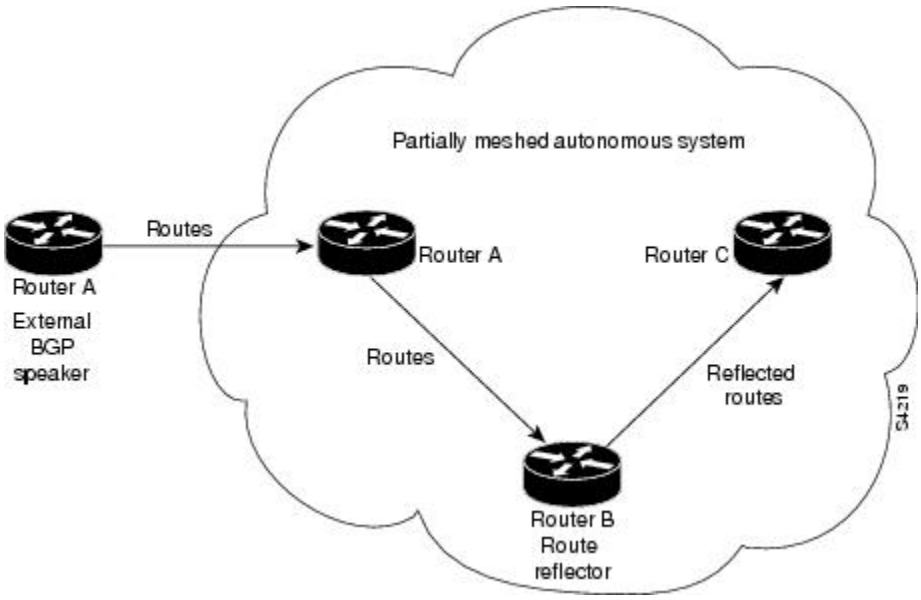
Figure below illustrates a simple iBGP configuration with three iBGP speakers (routers A, B, and C). Without route reflectors, when Router A receives a route from an external neighbor, it must advertise it to both routers B and C. Routers B and C do not readvertise the iBGP learned route to other iBGP speakers because the routers do not pass on routes learned from internal neighbors to other internal neighbors, thus preventing a routing information loop.

Figure 1: Three Fully Meshed iBGP Speakers



With route reflectors, all iBGP speakers need not be fully meshed because there is a method to pass learned routes to neighbors. In this model, an iBGP peer is configured to be a route reflector responsible for passing iBGP learned routes to a set of iBGP neighbors. In figure below, Router B is configured as a route reflector. When the route reflector receives routes advertised from Router A, it advertises them to Router C, and vice versa. This scheme eliminates the need for the iBGP session between routers A and C.

Figure 2: Simple BGP Model with a Route Reflector



The internal peers of the route reflector are divided into two groups: client peers and all other routers in the autonomous system (nonclient peers). A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster*. The nonclient peers must be fully meshed with each other, but the client peers need not be fully meshed. The clients in the cluster do not communicate with iBGP speakers outside their cluster.

Figure 3: More Complex BGP Route Reflector Model

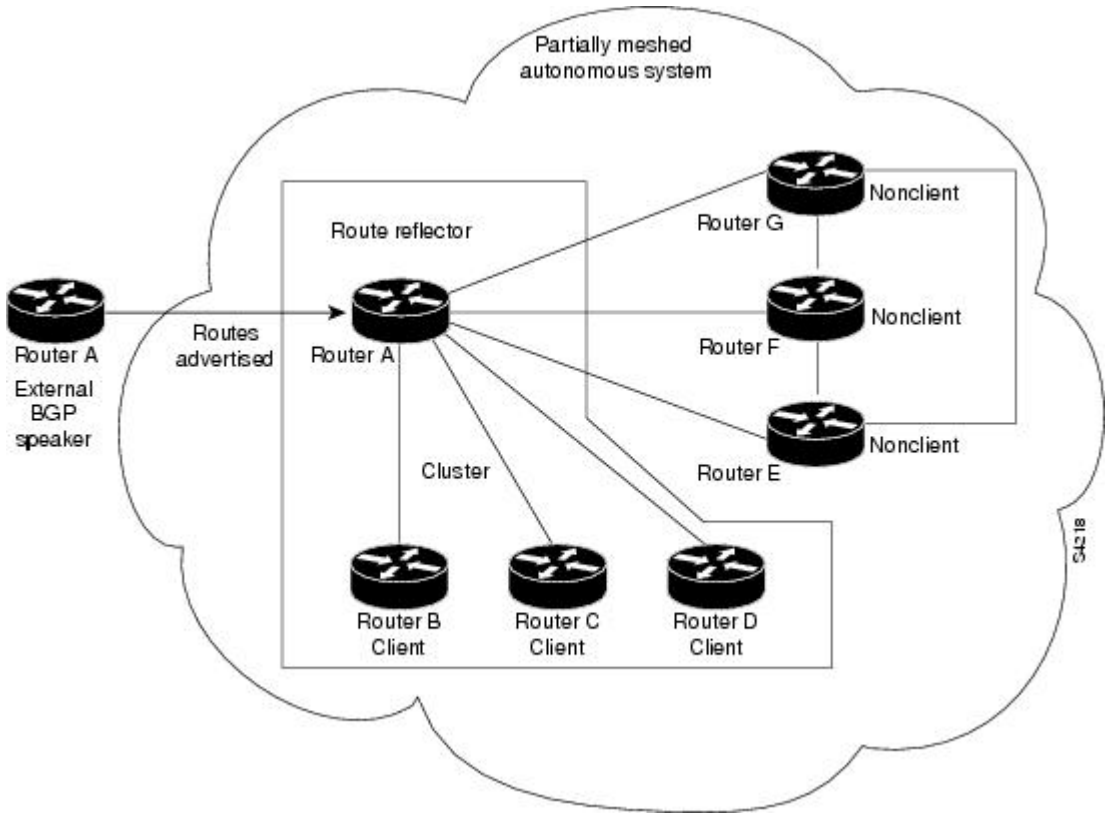


Figure above illustrates a more complex route reflector scheme. Router A is the route reflector in a cluster with routers B, C, and D. Routers E, F, and G are fully meshed, nonclient routers.

When the route reflector receives an advertised route, depending on the neighbor, it takes the following actions:

- A route from an external BGP speaker is advertised to all clients and nonclient peers.
- A route from a nonclient peer is advertised to all clients.
- A route from a client is advertised to all clients and nonclient peers. Hence, the clients need not be fully meshed.

Along with route reflector-aware BGP speakers, it is possible to have BGP speakers that do not understand the concept of route reflectors. They can be members of either client or nonclient groups, allowing an easy and gradual migration from the old BGP model to the route reflector model. Initially, you could create a single cluster with a route reflector and a few clients. All other iBGP speakers could be nonclient peers to the route reflector and then more clusters could be created gradually.

An autonomous system can have multiple route reflectors. A route reflector treats other route reflectors just like other iBGP speakers. A route reflector can be configured to have other route reflectors in a client group or nonclient group. In a simple configuration, the backbone could be divided into many clusters. Each route reflector would be configured with other route reflectors as nonclient peers (thus, all route reflectors are fully meshed). The clients are configured to maintain iBGP sessions with only the route reflector in their cluster.

Usually, a cluster of clients has a single route reflector. In that case, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure, a cluster might have more

than one route reflector. In this case, all route reflectors in the cluster must be configured with the cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster. All route reflectors serving a cluster should be fully meshed and all of them should have identical sets of client and nonclient peers.

By default, the clients of a route reflector are not required to be fully meshed and the routes from a client are reflected to other clients. However, if the clients are fully meshed, the route reflector need not reflect routes to clients.

As the iBGP learned routes are reflected, routing information may loop. The route reflector model has the following mechanisms to avoid routing loops:

- Originator ID is an optional, nontransitive BGP attribute. It is a 4-byte attributed created by a route reflector. The attribute carries the router ID of the originator of the route in the local autonomous system. Therefore, if a misconfiguration causes routing information to come back to the originator, the information is ignored.
- Cluster-list is an optional, nontransitive BGP attribute. It is a sequence of cluster IDs that the route has passed. When a route reflector reflects a route from its clients to nonclient peers, and vice versa, it appends the local cluster ID to the cluster-list. If the cluster-list is empty, a new cluster-list is created. Using this attribute, a route reflector can identify if routing information is looped back to the same cluster due to misconfiguration. If the local cluster ID is found in the cluster-list, the advertisement is ignored.

Table Policy

The table policy feature in BGP allows you to configure traffic index values on routes as they are installed in the global routing table. This feature is enabled using the **table-policy** command and supports the BGP policy accounting feature.

BGP policy accounting uses traffic indices that are set on BGP routes to track various counters.

Table policy also provides the ability to drop routes from the RIB based on match criteria. This feature can be useful in certain applications and should be used with caution as it can easily result in an unwanted traffic drop where BGP advertises routes to neighbors that BGP does not install in its global routing table and forwarding table.

BGP Keychains

BGP keychains enable keychain authentication between two BGP peers. The BGP endpoints must both comply with draft-bonica-tcp-auth-05.txt and a keychain on one endpoint and a password on the other endpoint does not work.

BGP is able to use the keychain to implement hitless key rollover for authentication. Key rollover specification is time based, and in the event of clock skew between the peers, the rollover process is impacted. The configurable tolerance specification allows for the accept window to be extended (before and after) by that margin. This accept window facilitates a hitless key rollover for applications (for example, routing and management protocols).

The key rollover does not impact the BGP session, unless there is a keychain configuration mismatch at the endpoints resulting in no common keys for the session traffic (send or accept).

Configure a Route Reflector for BGP

All the neighbors configured with the **route-reflector-client** command are members of the client group, and the remaining iBGP peers are members of the nonclient group for the local route reflector.

The NCS 4000 devices do not support VPNv6 for data plane forwarding. From Release 6.5.35, they support VPNv6 Route Reflector functionality, enabling them to reflect VPNv6 prefixes in the control plane without participating in data forwarding.

The BGP VPNv6 route-reflector enables in-band IPv6 management connectivity for remote Application Aware Controllers (AACs). This allows IPv6 management traffic to traverse a provider's core transport network, even when it primarily operates with IPv4/MPLS and includes devices like the NCS 4000, which do not support native IPv6 routing.

Use this task to configure a route reflector for BGP:

Procedure

Step 1 Run the **configure router bgp** *as-number* command to specify the autonomous system number and enter the BGP configuration mode, allowing you to configure the BGP routing process.

Example:

```
RP/0/RP0/CPU0(config)# router bgp 100
```

Step 2 Run the **address-family { vpnv6 } unicast** command to globally enable the VPNv6 address family within a router's BGP instance.

Example:

```
RP/0/RP0/CPU0(config)# router bgp 100
RP/0/RP0/CPU0(config-bgp)# address-family vpnv6 unicast
RP/0/RP0/CPU0(config-bgp-af)#
```

Step 3 Run the **neighbor** *ip-address* command to place the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

Example:

```
  RP/0/(config-bgp)# neighbor 172.168.40.24
```

(From Release 6.5.35) This example shows how to designate the specified neighbor as a route reflector client within the VPNv6 address-family configuration. This configures the router as a BGP route reflector and the neighbor as its client, enabling route reflection for that neighbor.

Example:

```
RP/0/RP0/CPU0:N540(config-bgp)#neighbor 4.4.4.4
RP/0/RP0/CPU0:N540(config-bgp-nbr)#address-family vpnv6 unicast
RP/0/RP0/CPU0:N540(config-bgp-nbr-af)#route-reflector-client
RP/0/RP0/CPU0:N540(config-bgp-nbr-af)#
```

Step 4 Run the **remote-as** *as-number* command to create a neighbor and assigns a remote autonomous system number to it.

Example:

```
RP/0/(config-bgp-nbr)# remote-as 2003
```

Step 5 Run the **keychain** *name* command to configures keychain-based authentication. Keychains provide secure authentication by supporting different MAC authentication algorithms and provide graceful key rollover.

Example:

```
RP/0/(config-bgp-nbr)# keychain kych_a
```

Step 6 Run the **update-source** *interface-type interface-id* command to allow sessions to use the primary IP address from a specific interface as the local address when forming a session with a neighbor. The interface-type interface-id arguments specify the type and ID number of the interface, such as TenGigEthernet or Loopback.

Example:

```
RP/0/(config-bgp-nbr)# update-source Loopback 1
```

Step 7 Run the **address-family { ipv4 | vpnv4 } labeled-unicast** to specify IPv4 or vpnv4 address family unicast and enter address family configuration submode.

Example:

```
RP/0/(config-nbr)# address-family ipv4 labeled-unicast
```

Step 8 Run the **route-reflector-client** command to configure the router as a BGP route reflector and configures the neighbor as its client.

Example:

```
RP/0/(config-bgp-nbr-af)# route-reflector-client
```

Step 9 Run the **commit** command and save the changes.

Step 10 Run the **show bgp vpnv6 unicast** command to verify the BGP VPNv6 routes.

Example:

```
RP/0/RP0/CPU0:R5_sito9#show bgp vpnv6 unicast
BGP router identifier 6.6.6.6, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 147
BGP NSR Initial initsync version 4 (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
Route Distinguisher: 65000:123 (default for vrf MGMT)
*>i55::1/128 5.5.5.5 0 100 0 ?
*> 66::1/128 :: 0 32768 ?
```

```
Processed 2 prefixes, 2 paths
RP/0/RP0/CPU0:R5_sito9#
```

55::1/128 and 66::1/128 are BGP VPNv6 routes, where 55::1/128 is internal route (locally originated) and 66::1/128 is iBGP VPNv6 route received from internal BGP neighbor.

Examples:

This example shows how to use an address family to configure internal BGP peer 6.6.6.6 as a route reflector client :

```
RP/0/# configure
RP/0/(config)# router bgp 100
RP/0/(config-bgp)# neighbor 6.6.6.6
RP/0/(config-bgp-nbr)# remote-as 100
RP/0/(config-bgp-nbr)# keychain kych_a
RP/0/(config-bgp-nbr)# update-source Loopback 1
RP/0/(config-bgp-nbr)# address-family ipv4 labeled-unicast
RP/0/(config-bgp-nbr-af)# route-reflector-client
```

From Release 6.5.35:

This example shows how to enable VPNv6 address family globally under a BGP instance:

```
RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# router bgp 100
RP/0/RP0:hostname(config-bgp)# address-family vpnv6 unicast
```

This example shows how to activate VPNv6 unicast for neighbor 4.4.4.4 and place the router in neighbor address family configuration mode for the VPNv6 unicast address family:

```
RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# router bgp 1
RP/0/RP0:hostname(config-bgp)# address-family vpnv6 unicast
RP/0/RP0:hostname(config-bgp-af)# neighbor 4.4.4.4
RP/0/RP0:hostname(config-bgp-nbr)# remote-as 100
RP/0/RP0:hostname(config-bgp-nbr)# update-source Loopback0
RP/0/RP0:hostname(config-bgp-nbr)# address-family vpnv6 unicast
RP/0/RP0:hostname(config-bgp-nbr-af)# route-reflector-client
RP/0/RP0:hostname(config-bgp-nbr-af)# address-family ipv4 labeled-unicast
RP/0/RP0:hostname(config-bgp-nbr-af)# route-reflector-client
```

Applying Table Policy

Perform this task to apply a routing policy to routes being installed into the routing table.

Procedure

Step 1 **configure**

Step 2 **router bgp** *as-number*

Example:

```
RP/0/(config)# router bgp 100
```

Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.

Step 3 **address-family { ipv4 | vpnv4 } unicast**

Example:

```
RP/0/(config-bgp)# address-family ipv4 unicast
```

Specifies the IPv4 or vpnv4 address family and enters address family configuration submode.

Step 4 **table-policy** *policy-name*

Example:

```
RP/0/(config-bgp-af)# table-policy drop-all
```

Applies the specified policy to routes being installed into the routing table.

Step 5 **commit**

Example:

The following example shows how to apply the drop-all policy to IPv4 unicast routes being installed in the routing table :

```
RP/0/# configure
RP/0/(config)# router bgp 100
RP/0/(config-bgp)# address-family ipv4 unicast
RP/0/(config-bgp-af)# table-policy drop-all
```

Configuring BGP Route Reflect Filtering by Table Policy

Perform this task to configure BGP route reflect filtering by table policy.

Procedure

Step 1 **configure**

Step 2 **router bgp** *as-number*

Example:

```
RP/0/(config)# router bgp 100
```

Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.

Step 3 **address-family { ipv4 | vpnv4 } unicast**

Example:

```
RP/0/(config-bgp)# address-family ipv4 unicast
```

Specifies the IPv4 or vpnv4 address family and enters address family configuration submode.

Step 4 **table-policy** *policy-name*

Example:

```
RP/0/(config-bgp-af)# table-policy drop-all
```

Applies the specified policy to routes being installed into the routing table.

Step 5 **exit**

Example:

```
RP/0/(config-bgp-af)# exit
```

Exits the current configuration mode.

Step 6 **neighbor** *ip-address*

Example:

```
  RP/0/(config-bgp)# neighbor 172.168.40.24
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

Step 7 **remote-as** *as-number*

Example:

```
RP/0/(config-bgp-nbr)# remote-as 2003
```

Creates a neighbor and assigns a remote autonomous system number to it.

Step 8 **keychain** *name*

Example:

```
RP/0/(config-bgp-nbr)# keychain kych_a
```

Configures keychain-based authentication. Keychains provide secure authentication by supporting different MAC authentication algorithms and provide graceful key rollover.

Step 9 **update-source** *interface-type interface-id*

Example:

```
RP/0/(config-bgp-nbr)# update-source Loopback 1
```

Allows sessions to use the primary IP address from a specific interface as the local address when forming a session with a neighbor. The interface-type interface-id arguments specify the type and ID number of the interface, such as TenGigEthernet or Loopback.

Step 10 **address-family { ipv4 | vpnv4 } labeled-unicast**

Example:

```
RP/0/(config-nbr)# address-family ipv4 labeled-unicast
```

Specifies IPv4 or vpnv4 address family unicast and enters address family configuration submode.

Step 11 **route-reflector-client**

Example:

```
RP/0/(config-bgp-nbr-af)# route-reflector-client
```

Configures the router as a BGP route reflector and configures the neighbor as its client.

Step 12 **commit**

Example:

The following example shows how to use an address family to configure internal BGP peer 100.4.1.1 as a route reflect filter by table policy client:

```
RP/0/# configure
RP/0/(config)# router bgp 100
RP/0/(config-bgp)# address-family ipv4 unicast
RP/0/(config-bgp-af)# table-policy drop-all
RP/0/(config-bgp-af)# exit
RP/0/(config-bgp)# neighbor 100.4.1.1
RP/0/(config-bgp-nbr)# remote-as 100
RP/0/(config-bgp-nbr)# keychain kych_b
RP/0/(config-bgp-nbr)# update-source Loopback 1
RP/0/(config-bgp-nbr)# address-family ipv4 labeled-unicast
RP/0/(config-bgp-nbr-af)# route-reflector-client
```

Verifying BGP

Perform this task to verify BGP configuration.

Procedure

Step 1 **show bgp summary**

Example:

```
RP/0/# show bgp summary
```

Displays the status of all BGP connections.

Step 2 **show bgp ipv4 labeled-unicast summary**

Example:

```
RP/0/# show bgp ipv4 labeled-unicast summary
```

Step 3 **show bgp neighbors**

Example:

```
RP/0/# show bgp neighbors
```

Displays the information about BGP connections to neighbors.

Step 4 **show bgp paths detail**

Example:

```
RP/0/# show bgp paths detail
```

Displays all the BGP paths in the database.

Step 5 **show bgp route-policy** *route-policy-name*

Example:

```
RP/0/# show bgp route-policy p1
```

Displays the BGP information about networks that match an outbound route policy.

Step 6 **show bgp policy**

Example:

```
RP/0/# show bgp policy
```

Displays the information about BGP advertisements under a proposed policy.

Step 7 **show bgp advertised neighbor** *ip-address* **summary**

Example:

```
RP/0/# show bgp advertised neighbor 10.0.101.4 summary
```

Displays the advertisements for neighbors or a single neighbor.

BGP Labeled Unicast

BGP labeled unicast (LU) enables MPLS transport across IGP boundaries. By advertising loopbacks and label bindings across IGP boundaries, we can communicate to other routers in remote areas that are not part of our local IGP. BGP LU advertisements only impact edge routers and border routers.

Let us consider a network with three different areas: one core and two aggregation areas on the side. Each area runs its own IGP, with no redistribution between them on the Area Border Router (ABR). Use of BGP is needed in order to provide an end-to-end MPLS LSP. BGP advertises the loopbacks of the PE routers with a label across the whole domain, and provides an end-to-end LSP. BGP is deployed between the PEs and ABRs with BGP Labeled Unicast.

The NCS4K-4H-OPW-QC2 line card supports BGP LU.

Advantages of BGP LU

Following are the advantages of BGP LU:

- With BGP LU, routes and labels are carried together and this increases the scalability.
- Enables filtering of next-hop loops, thereby reducing the labels advertised by LDP/ RSVP.
- Reduction of OSPF/ ISIS and LDP/RSVp databases.
- Enables establishing an end-to-end label path across domains.

Limitations of BGP LU

Following are the limitations of BGP LU:

- When PW uses BGP LU for signaling, preferred path is not supported.
- BGP LU is supported on the NCS4K-2H10T-OP-KS line card only in the OTN mode. Use the **hw-module profile otn 200g-slot-otn-only** command to enable BGP LU and this command places the line card in the OTN mode.
- When MPLS activate is configured between two directly connected BGP LU nodes, then a static route must be used to create end to end PW.

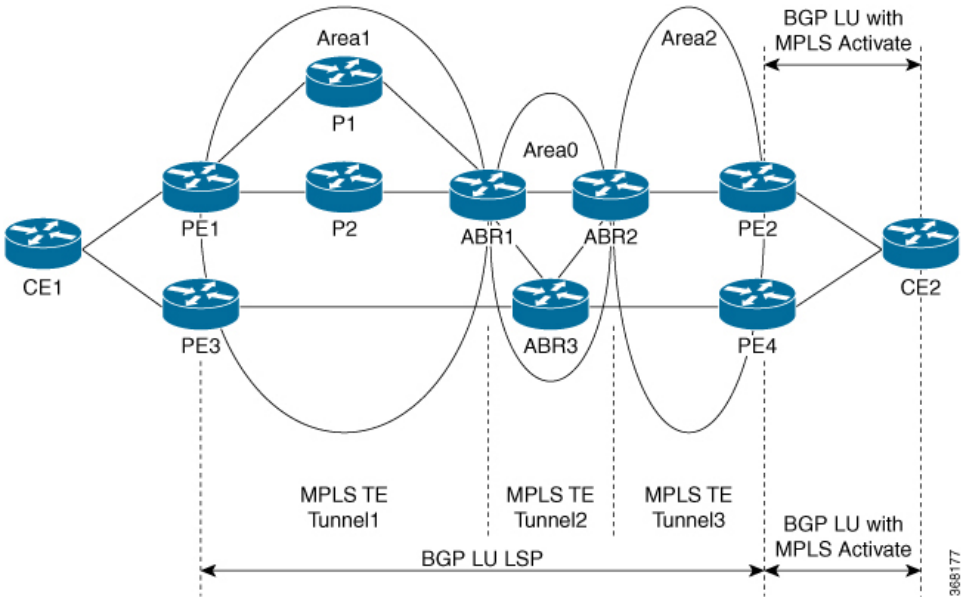
BGP LU Scale Details

In Release 6.5.31, BGP LU supports 500 scale sessions with 8000 prefixes.

Implementing BGP LU

Some of the use cases for BGL LU are discussed here. With reference to the below figure, consider BGP LU runs on the PE and the ABR routers (and not on the P routers). The IGP protocol used can be OSPF or ISIS.

Figure 4: Implementing BGP LU



- IP over BGP LU LSP over LDP - the IP packets are encapsulated with two labels (BGP label and the LDP label) from PE1 and sent to P2. The packet reaches with BGP label in ABR1. In ABR1, the BGP label is swapped and the packet reaches ABR2, only with the swapped BGP label. In ABR2, BGP label again gets swapped to reach PE2. PE2 acts like a PHP where the BGP label is popped before sending the packet to CE2.
- IP over BGP LU LSP over MPLS TE - the packet path is the same as discussed above. Here the IGP area has MPLS TE tunnels as transport.
- IP over BGP LU with TE tunnels with link/node protection FRR path - the packet path is the same , but in case of link failure (PE1 to P) or node failure (P), TE FRR on the PE1 takes the back up path (which is, PE2-P1-ABR1). In this case, the packet has three labels (BG label, TE label, Mergepoint label) to reach ABR1.
- VPWS ober BGP LU with TE tunnels - here, the VPWS service uses the BGP LU labelled path as transport to carry the pseudowires. The VC label is also added to the label stack. The back-up path includes four labels (VC-label, BGP label, TE label, MP label).

BGP LU and PIC Configuration

Perform this task to install a backup path into the forwarding table and provide prefix independent convergence (PIC) in case of a PE-CE link failure.

Procedure

Step 1 **configure**

Enters global configuration mode.

Step 2 **router bgp** *as-number*

Example:

```
RP/0/CPU0:router(config)# router bgp 100
```

Specifies the autonomous number and enters the BGP configuration mode.

Step 3 **address-family ipv4**

Example:

```
RP/0/CPU0:router(config-bgp)# address-family ipv4
```

Enters the address-family configuration mode.

Step 4 **neighbor** *ip address* **remote-as** *as_number* **address-family ipv4 label-unicast**

Example:

```
RP/0/CPU0:router(config-bgp-af)# neighbor 20.20.20.20
  remote-as 1
  update-source Loopback1
  address-family ipv4 label-unicast
   route-policy pass-all in
   route-policy pass-all out
```

Enables connecting to BGP LU neighbors.

Step 5 **additional-paths selection route-policy** *policy-name*

Example:

```
RP/0/CPU0:router(config-bgp-af)# additional-paths selection route-policy p1
```

Configures additional paths selection mode for a prefix. This calculates the backup paths and enables PIC.

Step 6 **commit**

Saves the configuration changes made.