Chapter 3

Configuring BGP-MPLS Applications

This chapter contains the following sections:

NOTE: Before you read this chapter, we recommend you be thoroughly familiar with both BGP and MPLS. For detailed information about those protocols, see *Chapter 1, Configuring BGP Routing* and *Chapter 2, Configuring MPLS*.

Overview

The BGP multiprotocol extensions (MP-BGP) enable BGP to support IPv4 services such as BGP multicast and BGP/MPLS virtual private networks (VPNs). BGP/MPLS VPNs are sometimes known as RFC 2547bis VPNs. Some of the applications for which you might use BGP/MPLS VPNs are to transport packets across an IP backbone, enable overlapping VPNs, operate inter-AS VPNs, enable multicast across VPNs, and provide carrier-of-carriers VPNs.

Address Families

The BGP multiprotocol extensions specify that BGP can exchange information within different types of *address families*. The JUNOSe BGP implementation defines the following different types of address families:

- Unicast IPv4—If you do not explicitly specify the address family, the router is configured to exchange unicast IPv4 addresses by default. You can also configure the router to exchange unicast IPv4 routes in a specified VRF.
- Multicast IPv4—If you specify the multicast IPv4 address family, you can use BGP to exchange routing information about how to reach a multicast source instead of a unicast destination. For information about BGP multicasting commands, see *Chapter 1, Configuring BGP Routing*. For a general description of multicasting, see *JUNOSe Multicast Routing Configuration Guide, Chapter 5, Configuring IPv4 Multicast*.
- VPN IPv4—If you specify the VPN-IPv4 (also known as VPNv4) address family, you can configure the router to provide IPv4 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- Unicast IPv6—If you specify the IPv6 unicast address family, you can configure the router to exchange unicast IPv6 routes or unicast IPv6 routes in a specified VRF. For a description of IPv6, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 2, Configuring IPv6*.
- Multicast IPv6—If you specify the multicast IPv6 address family, you can use BGP to exchange routing information about how to reach an IPv6 multicast source instead of an IPv6 unicast destination. For a general description of multicasting, see *JUNOSe Multicast Routing Configuration Guide, Chapter 5, Configuring IPv4 Multicast*.
- VPN IPv6—If you specify the VPN-IPv6 address family, you can configure the router to provide IPv6 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.

- L2VPN—If you specify the L2VPN address family, you can configure the PE router (L2VPNs) or VE router (VPLS) to exchange layer 2 network layer reachability information (NLRI) for all L2VPN (VPWS) or VPLS instances. Optionally, you can use the **signaling** keyword with the **address-family** command for the L2VPN address family to specify BGP signaling of L2VPN reachability information. Currently, you can omit the **signaling** keyword with no adverse effects. For a description of L2VPNs (VPWS), see ***Chapter 11, Configuring L2VPNs***. For a description of VPLS, see ***Chapter 8, Configuring VPLS***.

- Route-target—If you specify the route-target address family, you can configure the router to exchange route-target membership information to limit the number of routes redistributed among members. For a description of route-target filtering, see *Constraining Route Distribution with Route-Target Filtering* on page 393.

- VPLS—If you specify the VPLS address family, you can configure the router to exchange layer 2 NLRI for a specified VPLS instance. For a description of VPLS, see ***Chapter 8, Configuring VPLS***.

- VPWS—If you specify the VPWS address family, you can configure the PE router to exchange layer 2 NLRI for a specified L2VPN (VPWS) instance. For a description of L2VPNs (VPWS), see ***Chapter 11, Configuring L2VPNs***.

For information about specifying an address family, see *Configuring BGP VPN Services* on page 401.

Equal-Cost Multipath Support

Equal-cost multipath (ECMP) is a traffic load-balancing feature that enables traffic to the same destination to be distributed over multiple paths that have the same cost. BGP ECMP support for BGP/MPLS VPNs enables MPLS VPN routes to be included in the list of available equal-cost paths. You can specify that up to 16 equal-cost paths be considered.

The set of ECMP legs in a network can contain MPLS indirect next hops, either as a leg itself or pointed to by a leg. If the path to any of the MPLS indirect next hops fails, then the routing protocol begins recalculating the set of viable routes as soon as it is notified of the failure. When the recalculation has finished, the protocol then updates the routing table with the new routes.

From the time the path fails until the routing table is updated, the traffic flowing over the ECMP leg that has the failed MPLS indirect next hop is lost.

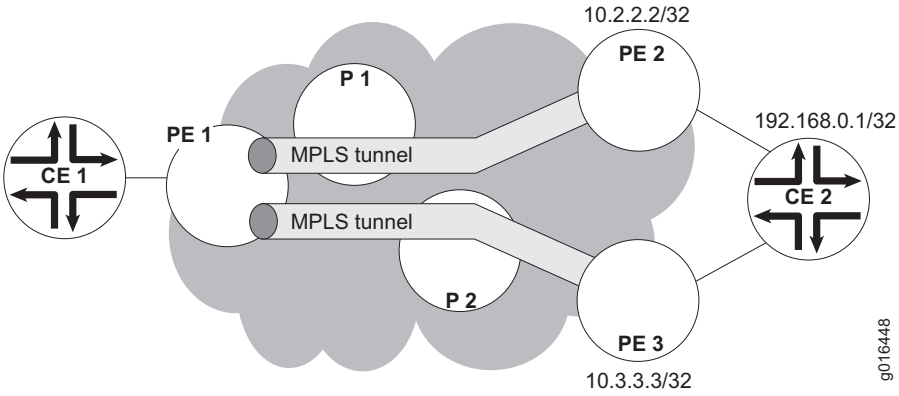
To reduce the amount of lost traffic, the failed path is quickly pruned from the ECMP set as soon as the protocol is notified of the connectivity failure. Traffic for the destination is then forwarded over the remaining equal-cost paths to the destination. When the recalculated set of routes is installed in the routing table, traffic for the destination is forwarded by means of the new route.

ECMP sets can have an MPLS indirect next hop as one of the legs in the following scenarios:

- In a BGP-MPLS VPN where a given VPN prefix is learned from multiple PE routers.
- When multiple RSVP-TE tunnels are created over different paths to the same destination.
- In a network that connects IPv6 islands across an IPv4 core, where a given IPv6 prefix is learned from multiple egress PEs running IPv6.

Consider the simple ECMP scenario for a BGP/MPLS VPN shown in Figure 67.

Figure 67: ECMP BGP/MPLS VPN Scenario



With respect to PE 1, this network has an ECMP set of two equal-cost legs for the VPN prefix of CE 2, 192.168.0.1/32:

- PE 1 -> P 1 -> PE 2 -> CE 2
- PE 1 -> P 2 -> PE 3 -> CE 2

The details of these routes are displayed by the following command:

```
host1:pe1:pe1-ce1#show ip route 192.168.0.1 detail
192.168.0.1/32 Type: Bgp Distance: 200 Metric: 0 Tag: 0 Class: 0
  MPLS next-hop: 741, ECMP next-hop, leg count 2
    MPLS next-hop: 389, label 17, VPN traffic, resolved by MPLS next-hop 376
      MPLS next-hop: 376, resolved by MPLS next-hop 385, peer 10.3.3.3
        MPLS next-hop: 385, label 24 on GigabitEthernet1/1/0.2
(ip19000002.mpls.ip [V:pe1]), nbr 10.3.2.2
    MPLS next-hop: 740, label 18, VPN traffic, resolved by MPLS next-hop 729
      MPLS next-hop: 729, resolved by MPLS next-hop 737, peer 10.2.2.2
        MPLS next-hop: 737, label 27 on GigabitEthernet1/1/0.1
(ip19000001.mpls.ip [V:pe1]), nbr 10.3.1.2
```

If the connection to PE 2 fails, BGP marks the MPLS next hop 729 as a failed indirect next hop as soon as BGP is notified of the loss of connectivity. However, some traffic continues to be forwarded to CE 2 through PE 2; this traffic is lost. BGP quickly prunes the failed route from the FIB, stopping this traffic loss, and then recalculates the routes to CE 2. During this period, traffic for CE 2 is forwarded only through PE 3. When the new routes are installed in the FIB, traffic is forwarded to CE 2 by means of the newly installed route.

BGP/MPLS VPN Components

If you have specified the VPN-IPv4 address family, you can configure virtual private networks across an IP backbone. BGP carries routing information for the network and MPLS labels, whereas MPLS transports the data traffic. Figure 68 shows a typical scenario.

The service provider backbone comprises two types of routers:

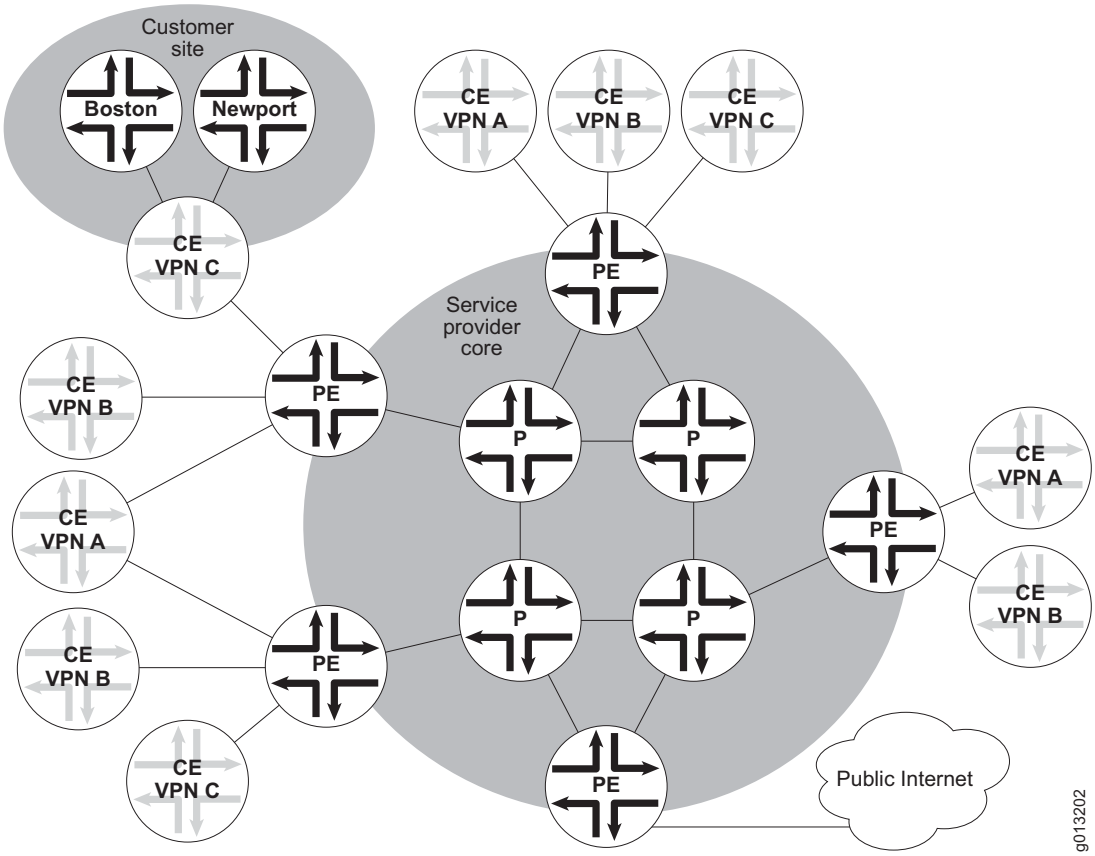
- Provider edge routers (PE routers)
- Provider core routers (P routers)

PE routers are situated at the edge of the service provider core and connect directly to customer sites. These routers must run BGP-4, including the BGP/MPLS VPN extensions. They must also be able to originate and terminate MPLS LSPs. (See *Chapter 2, Configuring MPLS*, for more information.)

P routers connect directly to PE routers or other P routers and do not connect directly to customer sites. These routers must be able to switch MPLS LSPs—that is, they function as MPLS label-switching routers (LSRs) and might function as label edge routers (LERs). Running BGP-4 on the P routers is not necessary to be able to exchange routing information for VPNs. You might run BGP-4 on the core routers for other reasons, such as exchanging routing information for the public Internet or implementing route reflectors. The P routes do not need to contain any information about customer sites.

PE routers communicate with customer sites through a direct connection to a customer edge (CE) device that sits at the edge of the customer site. The CE device can be a single host, a switch, or, most typically, a router. When the CE device is a router, it is a routing peer of all directly connected PE routers, but it is not a routing peer of CE routers at any other site. The link between the CE router and the PE router can employ any type of encapsulation. Using MPLS is not necessary. In Figure 68, each PE router connects to multiple CE routers and at least one P router. Although only one customer site is shown, each CE router lies within a customer site.

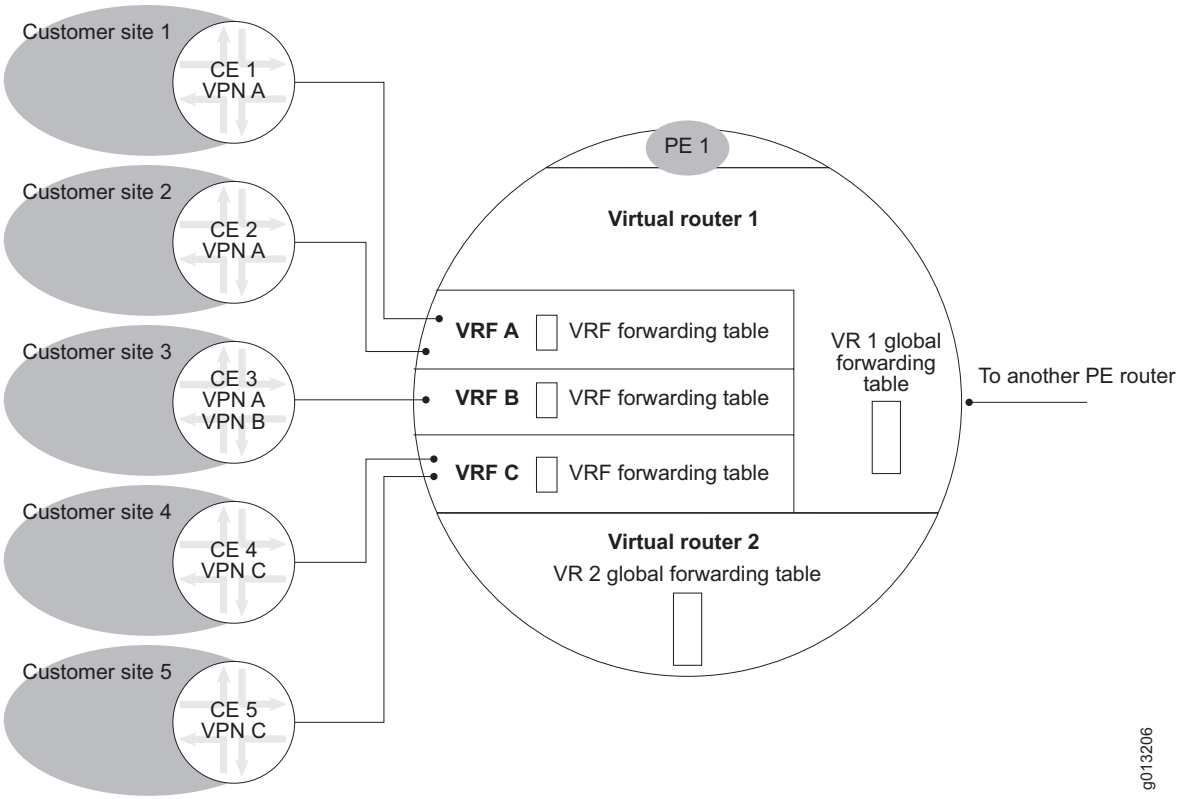
Figure 68: BGP/MPLS VPN Scenario



A customer site is a network that can communicate with other networks in the same VPN. A customer site can belong to more than one VPN. Two sites can exchange IP packets with each other only if they have at least one VPN in common.

Each customer site that is connected to a particular PE router is also associated with a VPN routing and forwarding instance (VRF). As shown in Figure 69, each VRF has its own forwarding table distinct from that of other VRFs and from the virtual router's global forwarding table.

Figure 69: BGP/MPLS VPN Components



A given VRF's forwarding table includes only routes to sites that have at least one VPN in common with the site that is associated with the VRF. For example, in Figure 69, the forwarding table in VRF B stores routes only to sites that are members of at least one of the VPNs to which Customer Site 3 belongs.

VRFs exist within the context of a virtual router (VR). A given virtual router can have zero or more VRFs, in addition to its global routing table (which is not associated with any VPN, CE router, or customer site). A router can support up to 1000 forwarding tables; that is, up to a combined total of 1000 VRs and VRFs.

You assign one or more interfaces or subinterfaces to a given VRF. If multiple customer sites are members of the same set of VPNs, they can share a VRF—that is, you do not need to create a specific VRF for each customer site. In Figure 69, Customer Sites 1 and 2 share VRF A; both sites belong to the same set of VPNs. The router looks up a packet's destination in the VRF associated with the interface on which the packet is received. The VRFs are populated by BGP while it learns routes from the VPN. If a customer site is a member of multiple VPNs, the routes learned from all those VPNs populate the VRF associated with the site.

VPN-IPv4 Addresses

Because each VPN has its own private address space, the same IP address might be used in several VPNs. To provide for more than one route to a given IPv4 address (each route unique to a single VPN), BGP/MPLS VPNs use route distinguishers (RDs) followed by an IPv4 address to create unique VPN-IPv4 addresses. A route can have only one RD.

The RD contains no routing information; it simply enables you to create unique VPN-IPv4 address prefixes. You can specify the RD in either of the following ways:

- An autonomous system (AS) number followed by a 32-bit assigned number. If the AS number is from the public address space, it must have been assigned to the service provider by the Internet Assigned Numbers Authority (IANA). The service provider can choose the assigned number. We recommend you do not use numbers from the private AS number space.

- An IP address followed by a 16-bit assigned number. If the IP address is from the public IP address space, it must have been assigned to the service provider by IANA. The assigned number may be chosen by the service provider. Use of numbers from the private IP address space is strongly discouraged.

You can create unique VPN-IPv4 addresses by assigning a unique RD to each VRF in your network. However, the optimal strategy depends on the configuration of your network. For example, if each VRF always belongs to only one VPN, you might use a single RD for all VRFs that belong to a particular VPN.

Route Targets

A route-target extended community, or route target, is a type of BGP extended community that you use to define VPN membership. The route target appears in a field in the update messages associated with VPN-IPv4.

You create route-target import lists and route-target export lists for each VRF. The route targets that you place in a route target export list are attached to every route advertised to other PE routers. When a PE router receives a route from another PE router, it compares the route targets attached to each route against the route-target import list defined for each of its VRFs. If any route target attached to a route matches the import list for a VRF, then the route is imported to that VRF. If no route target matches the import list, then the route is rejected for that VRF.

Depending on your network configuration, the import and export lists may be identical. Typically, you do the following:

- Allocate one route-target extended-community value per VPN.

- Configure the import list and the export list to include the same information: the set of VPNs comprising the sites associated with the VRF.

For more complicated scenarios—for example, hub-and-spoke VPNs—the route-target import list and the route-target export list might not be identical.

A route-target import list is applied before any inbound routing policy (route map) is applied. If an inbound route map contains a **set extcommunity** clause, the clause replaces all extended communities in the received route. BGP applies the default route-target export list associated with the VRF if the route does not have any route-target extended-community attributes after the inbound policy has been applied. On the other hand, the default export list is not applied if either a valid route-target export list is received or the inbound route map sets one or more route targets.

Distribution of Routes and Labels with BGP

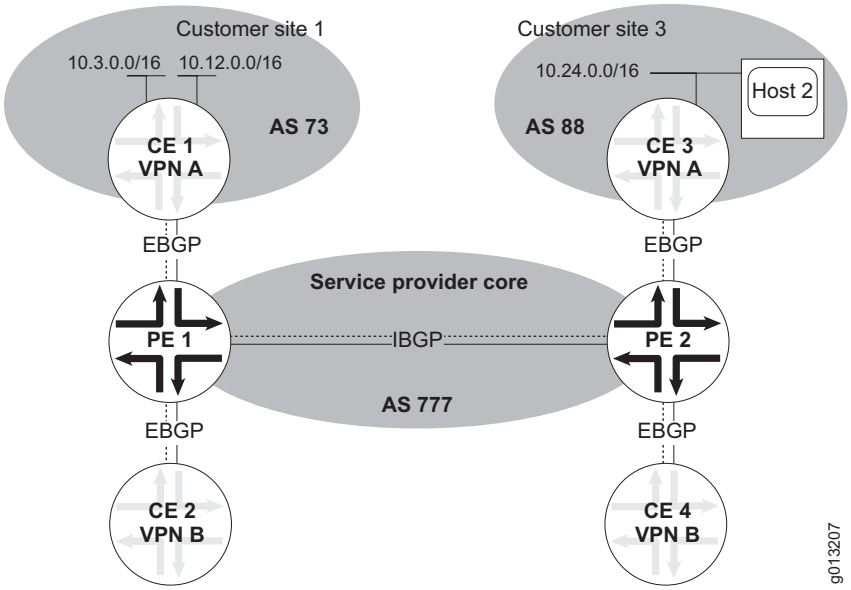
The extensions to BGP include enhancements to update messages that enable them to carry the route distinguishers, route-target extended-community information, and MPLS labels required for BGP/MPLS VPNs.

Consider the simple example shown in Figure 70. The customer edge devices are connected with their associated provider edge routers by external BGP sessions (CE 1–PE 1 and CE 3–PE 2). PE 1 and PE 2 are BGP peers by an internal BGP session across the service provider core in AS 777.

In this example, the PE routers run EBGP to the CE routers to do the following:

- Learn the prefixes of the networks in the local customer site.
- Advertise routes to networks and remote customer sites.

Figure 70: Route and Label Distribution



Rather than running EBGP between the PE routers and the CE routers, you can do either of the following:

- Run an IGP (such as IS-IS, OSPF, or RIP) between the CE router and the PE router.
- Configure static routes on the CE and PE routers (on the CE router this would typically be a default route).

In this example the two customer sites use different AS numbers, which simplifies configuration. Alternatively, the same AS numbers can be used.

Customer site 1 has two networks that need to be reachable from customer site 3—10.3.0.0/16 and 10.12.0.0/16—and uses BGP to announce these prefixes to PE 1. CE 1 uses a standard BGP update message as shown in Figure 71 to carry this and additional information. CE 1 is withdrawing prefix 10.1.0.0/16. CE 1 specifies its own address as the next hop; 10.4.1.1 is from the private address space of VPN A.

PE 1 passes the advertisement along the backbone through an IBGP session, but uses MP-BGP rather than standard BGP-4. Consequently, PE 1 uses an extended BGP update message, which is different in format from the standard message, as shown in Figure 71.

The extended update uses different attributes for some of the advertised information. For example it carries the advertised prefixes in the MP-Reach-NLRI attribute instead of the NLRI attribute. Similarly, it uses the MP-Unreach-NLRI attribute for withdrawn routes rather than the withdrawn-routes attribute.

PE 1 advertises the customer site addresses by prepending information to the addresses as advertised by CE 1, thus creating *labeled VPN-IPv4 prefixes*. The prepended information consists of a route distinguisher and an MPLS label.

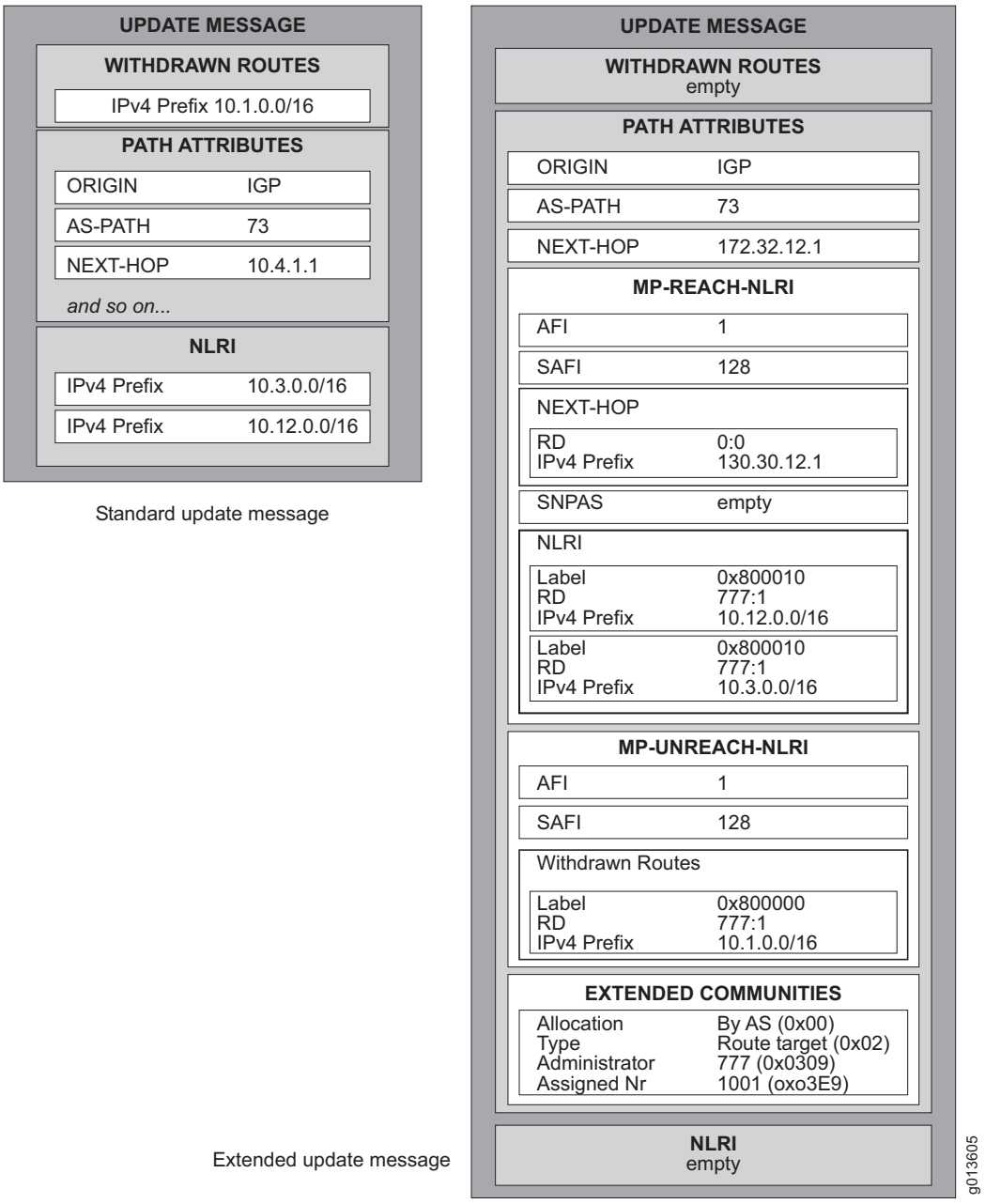
Because the CE router uses IPv4 addresses from the VPN's private address space, these addresses can be duplicated in other VPNs to which PE 1 is attached. PE 1 associates a route distinguisher with each IPv4 address to create a globally unique address. In this example, the RD consists of the AS that PE 1 belongs to and a number that PE 1 assigns. The RD is prepended immediately before the IPv4 address.

PE routers assign MPLS labels to each VRF. In this example, the label for the VRF associated with customer site 1 is 16. The MPLS label is prepended immediately before the route distinguisher.

NOTE: The explicit null label is prepended only to routes that are being withdrawn in the MP-REACH-NLRI attribute.

Some non–E-series implementations allocate a separate label for each prefix. By default, the E-series router generates one label for all BGP routes advertised by the VRF, thus reducing the number of stacked labels to be managed. The **ip mpls forwarding-mode label-switched** command enables you to have the router generate a label for each different FEC pointed to by a BGP route in a given VRF. However, some routes always receive a per-VRF label; see *Creating Labels per FEC* on page 421 for more information.

Figure 71: Standard and Extended BGP Update Messages



Using the **next-hop-self** option on PE 1 causes PE 1 to set the next-hop attribute to its own address, 172.32.12.1. Doing so is necessary because the next hop provided by CE 1 is from VPN A's private address space and has no meaning in the service provider core. In addition, PE 2 must have PE 1's address so that it can establish an LSP back to PE 1. The next-hop address must also be carried in the MP-Reach-NLRI attribute, according to MP-BGP.

The extended update also has the extended-communities attribute, which identifies the VPN to which the routes are advertised. In this example, the route target is 777:1001, identifying VPN A.

Platform Considerations

For information about modules that support BGP/MPLS VPNs on the ERX-7xx models, ERX-14xx models, and the ERX-310 router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support BGP/MPLS VPNs.

For information about modules that support BGP/MPLS VPNs on E120 routers and E320 routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support BGP/MPLS VPNs.

References

For more information about BGP/MPLS VPNs, consult the following resources:

- BGP/MPLS IP VPNs—draft-ietf-l3vpn-rfc2547bis-03.txt (April 2005 expiration)
- BGP-MPLS VPN extension for IPv6 VPN—draft-ietf-l3vpn-bgp-ipv6-03.txt (December 2004 expiration)
- Connecting IPv6 Islands across IPv4 Clouds with BGP—draft-ietf-ngtrans-bgp-tunnel-04.txt (July 2002 expiration)
- *JUNOSe Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.
- RFC 2545—Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing (March 1999)
- RFC 2547—BGP/MPLS VPNs (March 1999)
- RFC 2858—Multiprotocol Extensions for BGP-4 (June 2000)
- RFC 3107—Carrying Label Information in BGP-4 (May 2001)
- RFC 4684—Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs) (2006)

For more information about BGP and MPLS, see the *References* sections in *Chapter 1, Configuring BGP Routing* and in *Chapter 2, Configuring MPLS*.

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Web site at http://www.ietf.org for the latest drafts.

Transporting Packets Across an IP Backbone with MPLS

As described in the previous section, PE 1 and PE 2 exchange routing information, including MPLS labels for their customer sites, by means of a BGP session established between them across the service provider core.

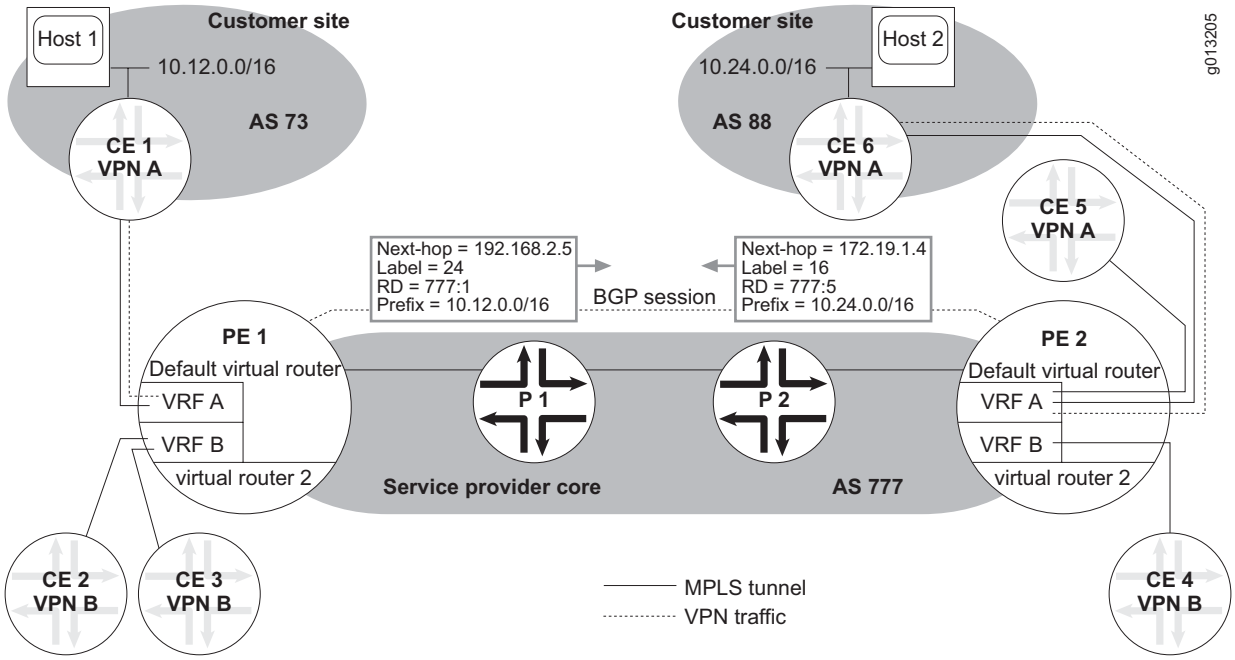
NOTE: To better understand MPLS before you read this section, see *Chapter 2, Configuring MPLS*.

Labels are employed in both the BGP control plane and the MPLS data plane. In the control plane, BGP advertises a route with an in label; this in label is also the label needed when MPLS traffic is received. BGP receives routes with an associated out label; the out label is the label sent with MPLS traffic.

Consider the network shown in Figure 72. If you display the in label on PE 1, you see that MP-BGP advertises a labeled VPN-IPv4 prefix of 10.12.0.0/16 with an in label of 24 (and an RD of 777:1, as shown in the illustration).

```
host1:pe1#show ip bgp vpn all field in-label
Prefix         In-label
10.12.0.0/16     24
10.24.0.0/16     none
```

Figure 72: BGP/MPLS VPN Route Exchange



If you display the in label on PE 2, you see that MP-BGP advertises a labeled VPN-IPv4 prefix of 10.24.0.0/16 with an in label of 16 (and an RD of 777:5, as shown in the illustration).

```
host2:pe2#show ip bgp vpn all field in-label
Prefix         In-label
10.12.0.0/16     none
10.24.0.0/16     16
```

On PE 1, you see that MP-BGP receives a labeled VPN-IPv4 prefix of 10.24.0.0/16 with an out label of 16. MP-BGP on PE 2 advertised this label with the prefix. In the data plane, MPLS traffic is sent by PE 1 to PE 2 with this label.

```
host1:pe1#show ip bgp vpn all field out-label
Prefix         Out-label
10.12.0.0/16     none
10.24.0.0/16     16
```

On PE 2, you see that MP-BGP receives a labeled VPN-IPv4 prefix of 10.12.0.0/16 with an out label of 24. MP-BGP on PE 1 advertised this label with the prefix. In the data plane, MPLS traffic is sent by PE 2 to PE 1 with this label.

```
host2:pe2#show ip bgp vpn all field out-label
Prefix         Out-label
10.12.0.0/16     24
10.24.0.0/16     none
```

The data packets are transported within a VPN across the service provider core by MPLS. This transport process requires two layers of MPLS labels, stacked one upon the other.

The inner labels are assigned by each PE router for each VRF. When an MPLS packet arrives at the egress PE router, that egress PE router uses the inner label to determine which VRF the packet is destined for. In the default, per-VRF label allocation mode (described in *Creating Labels per FEC* on page 421), the egress PE router does an IP lookup in the IP forwarding table of that VRF using the IP destination address in the IP packet that is encapsulated in the MPLS packet. The egress PE router then forwards the IP packet (without the MPLS header) to the appropriate customer site. The inner labels themselves are communicated between PE routers in the MP-BGP extended update messages as described in the previous section.

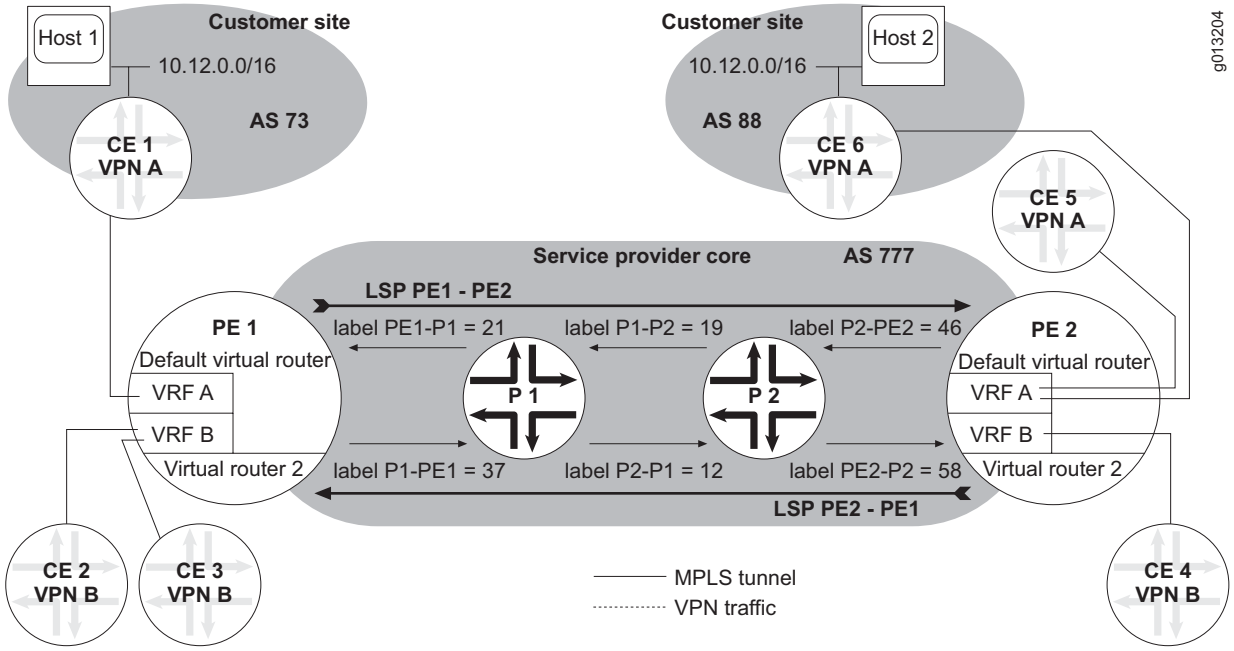
MPLS uses the outer labels to forward data packets from the ingress PE router through a succession of P routers across the core. This succession of P routers constitutes a label-switched path (LSP), also referred to as an MPLS tunnel. The labels are assigned to links in the path.

At each P router, MPLS pops the outer label from a data packet. The label is an index into the P router's forwarding table, from which it determines both the next hop along the LSP and another label. The router pushes the label on to the label stack and forwards the packet to the next P router. The combination of popping one label and pushing another is known as a label swap. At the egress PE router, MPLS pops the outer label, then the inner label. The inner label determines the CE router to which the packet is sent. The P routers never examine the inner MPLS label or the destination IP address encapsulated in the MPLS packet.

In many cases, the PE routers are fully meshed by means of LSPs. You can use tunnel profiles to simplify the LSP configuration process. See ***Chapter 2, Configuring MPLS***, for procedures to configure an LSP.

Each LSP is unidirectional for data traffic, so you must establish LSPs in both directions for two-way data transport. Figure 73 shows that two LSPs have been created between PE 1 and PE 2. PE 1 and PE 2 have an MP-BGP session as shown previously in Figure 72.

Figure 73: LSP Creation for BGP/MPLS VPN



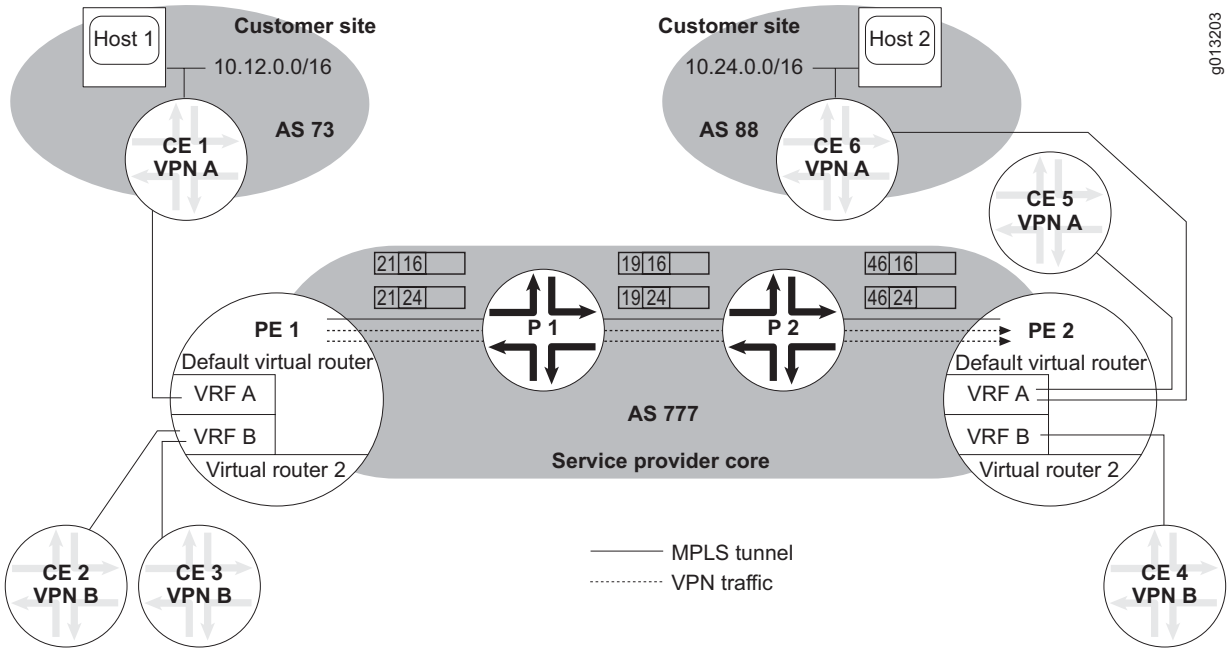
The PE 1–PE 2 LSP carries traffic only from PE 1 to PE 2, using label 21 for the PE 1 to P 1 link, label 19 for the P 1 to P 2 link, and label 46 for the P 2 to PE 2 link. PE 1 can forward data packets along the LSP to PE 2 and its customer sites.

Similarly, the PE 2–PE 1 LSP carries traffic only from PE 2 to PE 1, using label 58 for the PE 2 to P 2 link, label 12 for the P 2 to P 1 link, and label 37 for the P 1 to PE 1 link. PE 2 can forward data packets along the LSP to PE 1 and its customer sites.

Example: Data Transport

The process of data transport is shown in Figure 74. PE 1 has already received announcements from PE 2; an LSP has been established between PE 1 and PE 2.

Figure 74: Traffic Across the MPLS Backbone of a BGP/MPLS VPN



Host 1 constructs an IP packet with the address of Host 2 as the final destination, and sends the packet to router CE 1. CE 1 encapsulates the packet appropriately and forwards it to PE 1.

PE 1 receives the packet from CE 1. Based on the interface the packet came in on, PE 1 determines that it must use the forwarding table for VRF A to route the packet. PE 1 looks up the destination address of Host 2 in the forwarding table of VRF A and finds the following instructions:

- Push label 16; that is, prepend it to the data packet. This innermost label identifies the VRF on PE 2, where the final destination and interface lookup takes place. Label 16 was previously allocated by PE 2 and communicated to PE 1 by MP-BGP. VRF A shows this label as part of the NLRI for destination address Host 2.

- Push label 21 and forward the MPLS-encapsulated data packet to router P 1. Label 21 is prepended to label 16; the labels are *stacked*. Label 21 becomes the outermost label and is assigned to the first segment—PE 1–P 1—in the label-switched path from PE 1 to PE 2. The LSP was previously configured.

P 1 receives the data packet from PE 1 and pops label 21. P 1 looks up label 21 in its forwarding table and determines it must push label 19 on the stack, and forwards the data packet to P 2.

P 2 receives the data packet from P 1 and pops label 19. P 2 looks up label 19 in its forwarding table and determines it must push label 46 on the stack, and forwards the data packet to PE 2.

PE 2 receives the data packet from P 2, and looks up label 46. PE 2 determines it is the egress router of the LSP and must pop label 46. Then it proceeds to look up the next label, label 16, and determines that the packet goes to VRF A. Then the IP address is looked up in VRF A to determine the destination and outgoing interface for the packet. PE 2 forwards the packet to CE 6.

CE 6 receives the IP packet from PE 2 and looks up the destination address Host 2. Subsequent forwarding to Host 2 occurs by means of the IGP in the customer site.

The network structure shown in Figure 74 consists of two VPNs, A and B. VPN A comprises CE 1, CE 5, and CE 6. VPN B comprises CE 2, CE 3, and CE 4. CE 1 has data traffic destined for both CE 5 and CE 6. Because both of these destination sites are within the same VPN, PE 1 uses the same forwarding table, in VRF A, to do the lookups and MPLS encapsulation. The innermost label determines the destination VRF and is the same for all packets in that VPN, even if they are destined for different CE routers. CE 2 and CE 3 have traffic destined for CE 4. Because these all are in VPN B, PE 1 uses a different forwarding table, in VRF B, for looking up destinations for traffic originating with these sites. However, both VPNs use the same LSP, because both VPNs use the same ingress (PE 1) to and the same egress (PE 2) from the service provider core. Remember that the illustrated LSP carries data traffic only from PE 1 to PE 2. Traffic from PE 2 to PE 1 requires a different LSP.

Configuring IPv6 VPNs

The JUNOSe software supports IPv6 VPNs tunneled over an MPLS IPv4 backbone. A service provider can offer IPv4 VPN services, IPv6 VPN services, or both. MPLS over IPv6 is not currently supported. MPLS base tunnels to IPv6 destinations as tunnel endpoints are not supported, so you cannot establish an MPLS IPv6 backbone.

NOTE: You must configure an IPv6 interface in the parent VR for IPv6 VPNS to work.

BGP can negotiate VPNv6 capability without having to negotiate the IPv6 capability. BGP next-hop encoding varies depending on whether the backbone is IPv4 or IPv6. In the JUNOSe software implementation for IPv6 VPNs, the BGP next hops in the MP-BGP update message follow the convention for BGP next-hop encoding for IPv4 backbone. If an E-series router receives a BGP next hop that follows the encoding for an MPLS-enabled IPv6 backbone, that BGP next hop is treated as unreachable because currently no MPLS base tunnel to the native IPv6 tunnel endpoint address can exist.

The PE routers have both IPv4 and IPv6 capabilities. They maintain IPv6 VRFs for their IPv6 sites and encapsulate IPv6 traffic in MPLS frames that are then sent into the MPLS core network.

Link-local scope addresses cannot be used for reachability across IPv6 VPN sites and can never be advertised by means of MP-BGP to remote PE routers. Global scope addresses are expected to be used within and across IPv6 VPN Sites.

All features previously supported for BGP/MPLS IPv4 VPNs, such as policy-based routing, redistribution to and from other protocols, aggregation, route-flap dampening, and so on are also supported for BGP/MPLS IPv6 VPNs.

address-family

- Use to configure the router or VRF to exchange IPv4 or IPv6 addresses by creating the specified address family.
- IPv4 and IPv6 addresses can be exchanged in unicast, multicast, or VPN mode.
- The default setting is to exchange IPv4 addresses in unicast mode from the default router.
- Creating an address family for a VRF automatically disables both synchronization and automatic summarization for that VRF.
- This command takes effect immediately.
- Examples

```
host1:vr1(config-router)#address-family ipv4 multicast
host1:vr1(config-router)#address-family ipv4 unicast
host1:vr1(config-router)#address-family ipv4 unicast vrf vr2
host1:vr1(config-router)#address-family vpvn4 unicast
host1:vr1(config-router)#address-family vpnv6 unicast ecmplabel
host1:vr1(config-router)#address-family ipv6 multicast
```

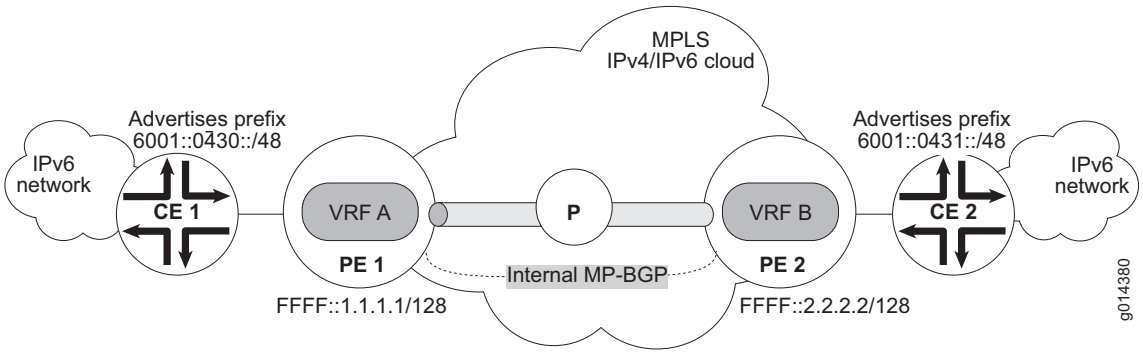
- Use the **no** version to disable the exchange of a type of prefix.

Intra-AS IPv6 VPNs

In Figure 75, a service provider is offering IPv6 VPN service over an MPLS-enabled IPv4 backbone. The base MPLS tunnels are established in the IPv4 core network with either of the MPLS signaling protocols (LDP or RSVP). The ingress PE router pushes the LSP tunnel label directly onto the label stack of the labeled IPv6 VPN packet. The topmost label imposed corresponds to the LSP that runs from the ingress PE router to the egress PE router. The BGP next-hop field identifies the egress PE router, and therefore the topmost label to be pushed on the stack. The bottom label is the label bound to the IPv6 VPN prefix by means of BGP.

The CE devices can attach to the VRFs on the PE routers using both an IPv6 link and an IPv4 link. In Figure 75, the CE devices attach to the VRFs over an IPv4 link, and use MP-BGP to connect to the VRFs on the PE routers. This arrangement enables the PE routers to learn IPv6 routes from MP-BGP running over TCPv4 from the CE devices. You can also configure IPv6 static routes in the VRFs on the PE routers to reach the networks through the CE IPv6 link. Alternatively, you can configure the static routes with any routing protocol that supports IPv6, such as OSPFv3.

Figure 75: IPv6 VPN Services over IPv4 MPLS



The PE routers use an MP-BGP session over TCPv4 to advertise the IPv6 routes from the CE devices to the remote PE routers. The IPv6 routes are advertised as labeled VPNv6 routes with a BGP next hop set to the base tunnel endpoint destination address. The next hop is formatted as an IPv4-mapped IPv6 address.

For IPv6 VPN services over an IPv4 backbone, the BGP next hop in the MP_REACH_NLRI attribute contains a VPN-IPV6 address with the RD set to zero and with the 16-byte IPv6 address encoded as an IPv4-mapped IPv6 address that contains the IPv4 address of the advertising PE router. This IPv4 address must be routable in the service provider's backbone.

BGP Control Plane Behavior

The VPN service in Figure 75 includes both CE 1 (VRF A) and CE 2 (VRF B). The MPLS base tunnels are established to tunnel endpoints PE 1 and PE 2 at their loopback interfaces. The loopback address for PE 1 is FFFF::1.1.1.1/128; for PE 2, it is FFFF::2.2.2.2/128.

The BGP next hop that is advertised in the MP-BGP update includes the following:

- A VPN-IPV6 address with the RD set to zero
- The 16-byte IPv6 address encoded as an IPv4-mapped IPv6 address that contains the IPv4 loopback address of the advertising PE router

The IPv4 IGP, such as OSPF, advertises the reachability of the loopback interfaces on the PE routers. LDP binds label L2 to 1.1.1.1/32 on the P router.

CE–PE Behavior

CE 1 is connected to VRF A in PE 1 through an IPv4 interface. Similarly, CE 2 is connected to VRF B in PE 2 through an IPv4 interface. You can alternatively run OSPF to the CE devices over IPv6 links and redistribute the OSPF IPv6 routes into BGP.

The MP-BGP sessions between the CE devices and the VRFs in the PE routers are established over TCPv4. The AFI value is 2, indicating IPv6; the SAFI value is 1, indicating unicast. CE 1 advertises IPv6 network 6001:0430::/48 to its MP-BGP peer in VRF A. CE 2 advertises 6001:0431::/48 to its MP-BGP peer in VRF B. When it receives the advertised prefix in VRF A, BGP adds 6001:0430::/48 to its BGP VPNv6 RIB with the stacked label L1, which MPLS allocated for this prefix. The default IPv6 VRF label is L1.

PE–PE Behavior

PE 1 advertises the VPNv6 prefixes in the MP_REACH_NLRI attribute of the update messages sent to its MP-IBGP peer, PE 2. The AFI and SAFI values are negotiated for VPNv6. The AFI value is 2 for IPv6, and the SAFI value is 128 for MPLS-labeled VPN-IPv6.

When PE 2 receives the VPNv6 prefix 6001:0430::/48 with label L1, it imports the prefix into VRF B because VRF B's import route target matches the route target received in the MP-BGP update. For all labeled VPNv6 prefixes installed in VRF B that come from the same endpoint on PE 1 (loopback FFFF::1.1.1.1/128), a single dynamic IPv6 interface stacked on top of an MPLS tunnel head is created in VRF B regardless of the number of different stacked labels associated with each VPNv6 prefix. The prefix is then installed in VRF B's routing table as pointing to this dynamic IPv6 interface.

If PE 1 is not running either JUNOSe or JUNOS software, each VPNv6 prefix usually has a different stacked label value sent in the MP-BGP update. If an implementation allocates one VPN interface per received stacked label, this behavior might potentially become a scaling issue if many dynamic IPv6 interfaces are allocated to resolve each VPNv6 prefix in VRF B.

MPLS Data Plane Behavior

When PE 2 receives a data packet from CE 2 destined for the 6001:0430::/48 network, the router detects a native IPv6 packet on its link to CE 2. PE 2 does a lookup in its VRF B IPv6 routing table, prepends labels L2 and L1 to the IPv6 header, and then forwards this packet on its core-facing IPv6 dynamic interface. When the P router receives this packet, it performs a lookup on L2 and label switches the packet toward PE 1. The P router either replaces L2 with another label or pops that label if PE 1 requested PHP.

When PE 1 receives the packet on its core-facing interface, it pops all the labels, and performs a lookup in the IPv6 table of VRF A (which is associated with L1) using the destination address in the IPv6 header. After that, PE 1 forwards the IPv6 packet out to CE 1 on the IPv6 link.

Providing IPv4 VPN Services Across Multiple Autonomous Systems

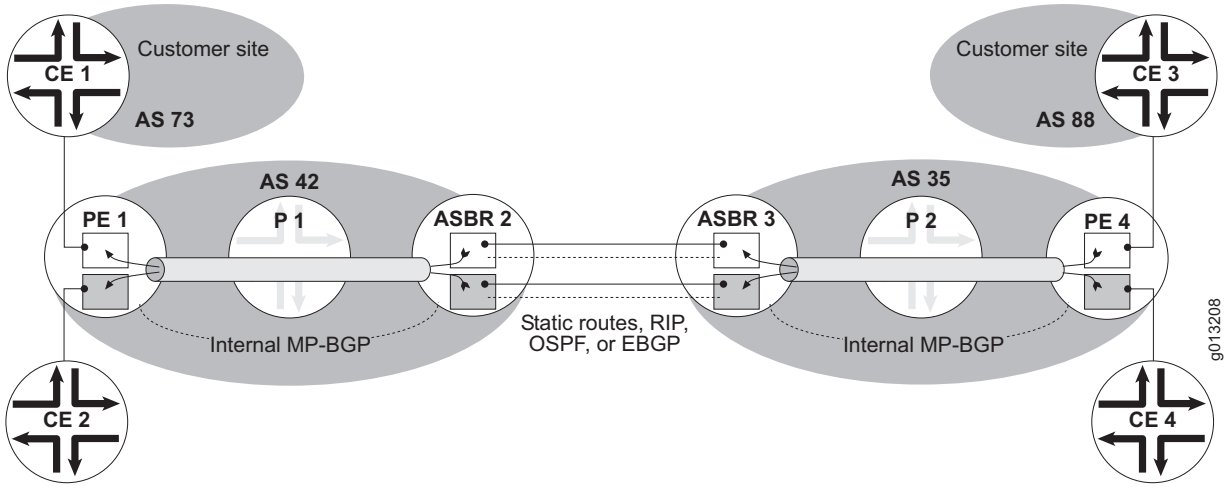
Inter-AS services, sometimes known as interprovider services, support VPNs that cross AS boundaries. VPNs might need to cross AS boundaries because of a customer deployment that involves geographically separated ASs. The VPN sites can be provided by the same service provider or by different service providers as part of a joint VPN service offering. Inter-AS services are also useful to service providers that use confederations of sub-ASs to reduce the IBGP mesh inside the AS.

You can support these inter-AS services in three different ways, known as inter-AS option A, option B, and option C. Option C is preferred to option B; option B is preferred to option A. For inter-AS options B and C, you must explicitly configure MPLS on all the inter-AS links.

Inter-AS Option A

Figure 76 illustrates the first method, where you create a VRF for each VPN on each AS boundary router.

Figure 76: Inter-AS Topology with VRFs on Each AS Boundary Router



Within each AS, routes are announced by internal MP-BGP and the data packets are forwarded across an MPLS tunnel. You create a logical connection such as an ATM VC between each pair of VRFs (on separate AS boundary routers); these logical connections can share the same physical connection. The following factors limit the scalability of this method:

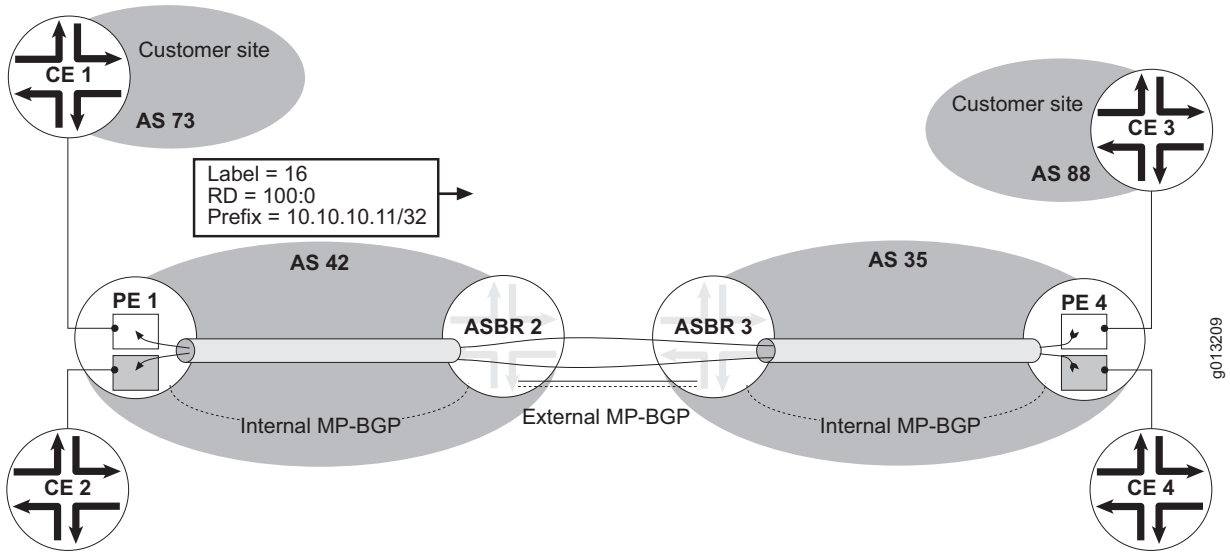
- All inter-AS VPN routes (potentially a very large number) must be stored in the BGP RIBs and IP routing tables on the AS boundary routers.
- You must configure VRFs on each AS boundary router.

MPLS tunnels are unidirectional; Figure 76 shows only the tunnels established to carry traffic from ASBR 2 to PE 1 and from PE 4 to ASBR 3. Note that ASBR 2 and ASBR 3 are both also PE routers. In that sense, ASBR 2 treats ASBR 3 as a CE router, and ASBR 3 treats ASBR 2 as a CE router.

Inter-AS Option B

The second method is known as inter-AS option B or 2547bis option B, after IETF draft RFC BGP/MPLS IP VPNs—draft-ietf-l3vpn-rfc2547bis-03.txt (April 2005 expiration). This method uses BGP to signal VPN labels between the AS boundary routers (Figure 77). The base MPLS tunnels are local to each AS. Stacked tunnels run from end to end between PE routers on the different ASs. This method provides greater scalability, because only the BGP RIBs store all the inter-AS VPN routes.

Figure 77: Inter-AS Topology with End-to-End Stacked MPLS Tunnels



PE 1 assigns labels for routes to the customer sites, and distributes both the label assignments and the VPN-IPv4 routes throughout AS 42 in extended BGP update messages by means of internal MP-BGP. ASBR 2 then distributes the routes to ASBR 3 with external MP-BGP; ASBR 2 specifies itself as the next-hop address and assigns a new label to the route so that ASBR 3 can properly direct traffic. ASBR 3 propagates the routes by internal MP-BGP throughout AS 35, including to PE 4.

Example You can use the **show ip bgp vpn all field in-label** and **show ip bgp vpn all field out-label** commands in the context of each VPN element to display the in label and out label associated with the route at that point. Suppose that CE 1 advertises a route to prefix 10.10.10.11/32 to its external BGP peer PE 1 (10.2.2.2) in VRF A. PE 1 associates the label 16 with this route; an extended update message sent to internal MP-BGP peer ASBR 2 carries this information as a labeled VPN-IPv4 prefix (label 16, RD 100:0, IPv4 prefix 10.10.10.11/32).

```
host1:pe1#show ip bgp vpn all field in-label
Prefix             In-label
10.10.10.11/32     16
```

On PE 1, no out label is associated with the IPv4 prefix 10.10.10.11/32.

```
host1:pe1#show ip bgp vpn all field out-label
Prefix             Out-label
10.10.10.11/32     none
```

ASBR 2 receives the labeled VPN-IPv4 prefix and generates a new label, 44, to associate with this VPN-IPv4 prefix instead of label 16 when it sends the prefix to ASBR 3.

```
host1:asbr2#show ip bgp vpn all field out-label
Prefix             Out-label
10.10.10.11/32     16

host1:asbr2#show ip bgp vpn all field in-label
Prefix             In-label
10.10.10.11/32     44
```

ASBR 2 receives MPLS frames with label 44 (the in label) from ASBR 3 and sends MPLS frames with label 16 (the out label) to PE 1.

The inter-AS next hop shows label 44 as the label advertised to inter-AS peer ASBR 3. Label 44 was generated for the indirect next hop PE router/label pair, 10.2.2.2 (PE 1) and 16. Indirect next hop 1.1.1.1 is for the MP-IBGP peering between PE 1 (loopback address 1.1.1.1) and ASBR 2. Indirect next hop 10.5.5.5 is ASBR 3.

```
host1:asbr2#show ip bgp vpn all next-hops
Indirect next-hop 1.1.1.1
  Resolution in IP route table of VR
       IP indirect next-hop index 10
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.20 (10.20.20.1)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 29
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 23
  Reference count is 1

Indirect next-hop 10.5.5.5
   Resolution in IP route table of VR
       IP indirect next-hop index 5
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/0.21 (10.5.5.5)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 14
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/0.21.mpls
  Reference count is 3

host1:asbr2#show mpls next-hop 23
MPLS next-hop: 23, label 33 on ATM6/1.20, nbr 10.20.20.1
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards

host1:asbr2#show mpls next-hop 29
MPLS next-hop: 29, resolved by MPLS next-hop 23, peer 1.1.1.1
  MPLS next-hop: 23, label 33 on ATM6/1.20, nbr 10.20.20.1
  Statistics collection is disabled
```

```
host1:asbr2#show mpls forwarding brief
....
44       bgp      swap to 16, push 34 on ATM6/1.20, nbr 10.20.20.1

host1:asbr2#show mpls forwarding label 44
In label: 44
  Label space: platform label space
  Owner: bgp
  Spoof check: router ASBR2
  Action:
    MPLS next-hop: 30, label 43, resolved by MPLS next-hop 29
      MPLS next-hop: 29, resolved by MPLS next-hop 23, peer 1.1.1.1
        MPLS next-hop: 23, label 34 on ATM6/1.20, nbr 10.20.20.1
  Statistics:
    0 in pkts
    0 in Octets
    0 in errors
    0 in discard pkts
```

ASBR 3 in turn generates a new label, 50, to advertise with the VPN-IPv4 prefix to its internal MP-BGP peer inside its autonomous system, AS 35. Indirect next hop 4.4.4.4 is for the MP-IBGP peering between PE 4 (loopback address 4.4.4.4) and ASBR 3. Indirect next hop 10.5.5.50 is ASBR 2.

```
host1:asbr3#show ip bgp vpn all field out-label
Prefix             Out-label
10.10.10.11/32     44

host1:asbr3#show ip bgp vpn all field in-label
Prefix             In-label
10.10.10.11/32     50

host1:asbr3#show ip bgp vpn all next-hops
Indirect next-hop 4.4.4.4
  Resolution in IP route table of VR
       IP indirect next-hop index 11
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.33 (33.33.33.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 28
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 22
  Reference count is 1

Indirect next-hop 10.5.5.50
  Resolution in IP route table of VR
       IP indirect next-hop index 4
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.21 (10.5.5.50)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 11
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.21.mpls
  Reference count is 3

host1:asbr3#show mpls forwarding brief
...
50       bgp      swap to 44,  on ATM6/1.21
```

In turn, ASBR 3 receives MPLS frames with label 50 (the in label) from PE 4 and sends MPLS frames with label 44 (the out label) to ASBR 2.

PE 4 receives the VPN-IPv4 prefix with label 50:

```
host1:pe4#show ip bgp vpn all field out-label
Prefix             Out-label
10.10.10.11/32     50
```

On PE 4, no in label is associated with the IPv4 prefix 10.10.10.11/32.

```
host1:pe4#show ip bgp vpn all field in-label
Prefix             In-label
10.10.10.11/32     none
```

The labels that are generated to be sent to the inter-AS BGP peers are generated for each next-hop PE router/received label tuple. Scaling is improved when all routes advertised from a given VRF have the same label; this is the default E-series router behavior. You can disable this behavior by issuing the **ip mpls forwarding-mode label-switched** command for the VRF.

Inter-AS Option C

The third method of configuring inter-AS services and inter-AS VPNs is known as inter-AS option C or 2547bis option C. This method is described in BGP/MPLS IP VPNs—draft-ietf-l3vpn-rfc2547bis-03.txt (April 2005 expiration).

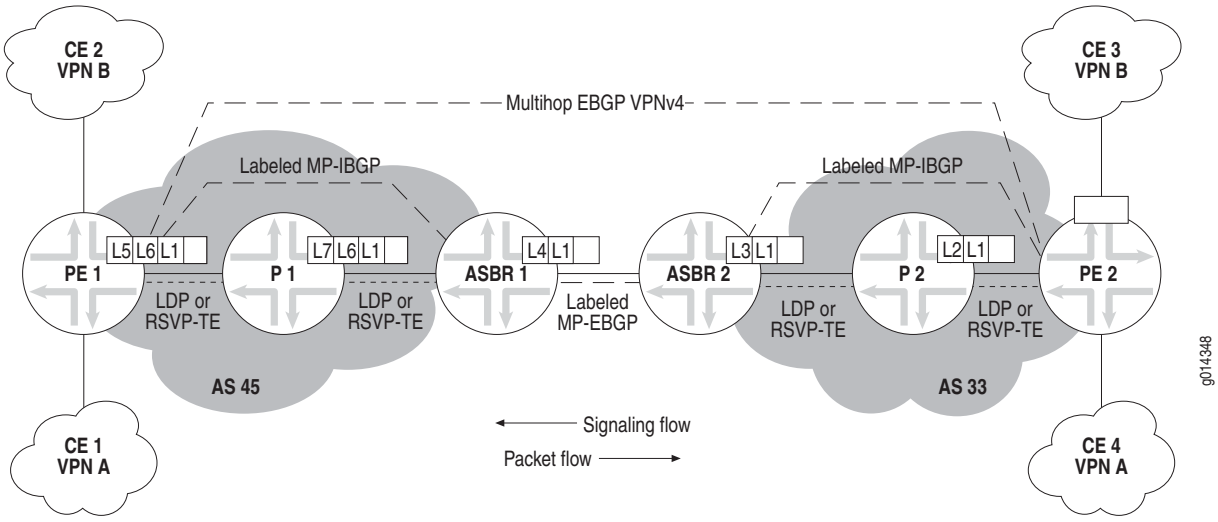
Inter-AS option C, similarly to the carrier-of-carriers configuration, requires a label-switched path from a packet's ingress PE router to its egress PE router. Option C introduces multihop EBGP redistribution of labeled VPN-IPv4 routes between source and destination autonomous systems. Labeled IPv4 routes are redistributed by EBGP between neighboring autonomous systems. Inter-AS option C uses BGP as the label distribution protocol.

In an inter-AS option C network, ASBRs do not maintain or distribute VPN-IPv4 routes. Each ASBR maintains labeled IPv4 /32 routes to the PE routers within its AS. The ASBR distributes these routes to other autonomous systems with EBGP. If transit autonomous systems are included in the topology, their ASBRs must also distribute the labeled /32 routes with EBGP. This configuration creates a label-switched path from the ingress PE router to the egress PE router. This configuration enables the PE routers in different autonomous systems to establish multihop EBGP connections to each other, and to exchange VPN-IPv4 routes over those connections.

Two different configuration scenarios are possible with option C, one employing a two-label stack and the other a three-label stack.

Figure 78 illustrates the three-label stack scenario. PHP is not used in this example.

Figure 78: Topology for Three-label Stack Configuration for Inter-AS Option C



In this topology, you can use either LDP or RSVP-TE to establish an LSP between each ASBR router and the PE router in an autonomous system. A labeled MP-IBGP session exists between the ASBR and the PE router in each autonomous system. A labeled MP-EBGP session exists between the two ASBR routers. The ASBR routers advertise the loopback IP addresses of their PE routers and associates the prefixes with labels.

When PE 1 learns the PE 2 loopback address and PE 2 learns the PE 1 loopback address, these PE routers can establish a multihop MP-EBGP session in order to exchange VPN-IPv4 routes. Because VPN-IPv4 routes are only exchanged between end PE routers, no other router on the path from PE 1 to PE 2 needs to keep or install VPN routes in its RIB or FIB.

1. P 2 learns label L2 for the route to the loopback address on PE 2 by means of LDP or RSVP-TE from PE 2.

2. ASBR 2 learns label L3 for the route to the loopback address on PE 2 by means of LDP or RSVP-TE from P 2.

   Each ASBR builds its own MPLS forwarding table with the received and advertised routes and labels. ASBR 2 uses its own IP address as the next hop.

3. ASBR 2 uses an MP-EBGP labeled unicast session to advertise label L4 for the route to the loopback address on PE 2 to neighboring ASBR 1.

4. ASBR 1 receives this route and the associated label L4.

5. ASBR 1 assigns label L6 to the route to the loopback address on PE 2 and changes the next-hop address to its own address.

6. ASBR 1 then uses an MP-IBGP session to advertise that address to PE 1. PE 1 therefore has an update with the label information and a next hop to ASBR 1.

7. P 1 learns label L7 for the route to the loopback address on ASBR 1 by means of LDP or RSVP-TE from ASBR 1.

8. PE 1 learns label L5 for the route to the loopback address on ASBR 1 by means of LDP or RSVP-TE from P 1.

9. PE 1 learns label L1 for the VPN-IPv4 route from the multihop EBGP session with PE 2.

Because the routes to the PE routers are unknown to all P routers other than the ASBRs, the ingress PE must push a three-label stack on packets received from the VPN end users. This is illustrated in Figure 78 as follows:

1. The first (innermost or bottom) label, L1, is assigned by the egress PE router, PE 2. This label is obtained from the multihop MP-EBGP session. It corresponds to the packet's destination address in a particular VRF at the remote PE router.

2. The middle label, L6, is assigned by ASBR 1. This label is obtained from the MP-IBGP labeled unicast session from the ASBR. It corresponds to the /32 route to the egress PE router, PE 2.

3. The top (outermost) label, L5, is assigned by the ingress PE router's IGP next hop. P 1. This label is obtained from an LDP or RSVP-TE session with the next hop. It corresponds to the /32 route to ASBR 1.

While the packet travels across the VPN from ingress router PE 1, labels are swapped as follows:

1. P 1 swaps outermost label L5 for L7 to get to its next hop, ASBR 1.

2. ASBR 1 pops outermost label L7 and swaps the middle label L6 for L4 to get to ASBR 2.

3. ASBR 2 swaps outer label L4 for L3 to get to its next hop, P 2.

4. P 2 swaps outer label L3 for L2 to get to its next hop, PE 2.

5. PE 2 pops outer label L2 and inner label L1 and then processes the IP data packet.

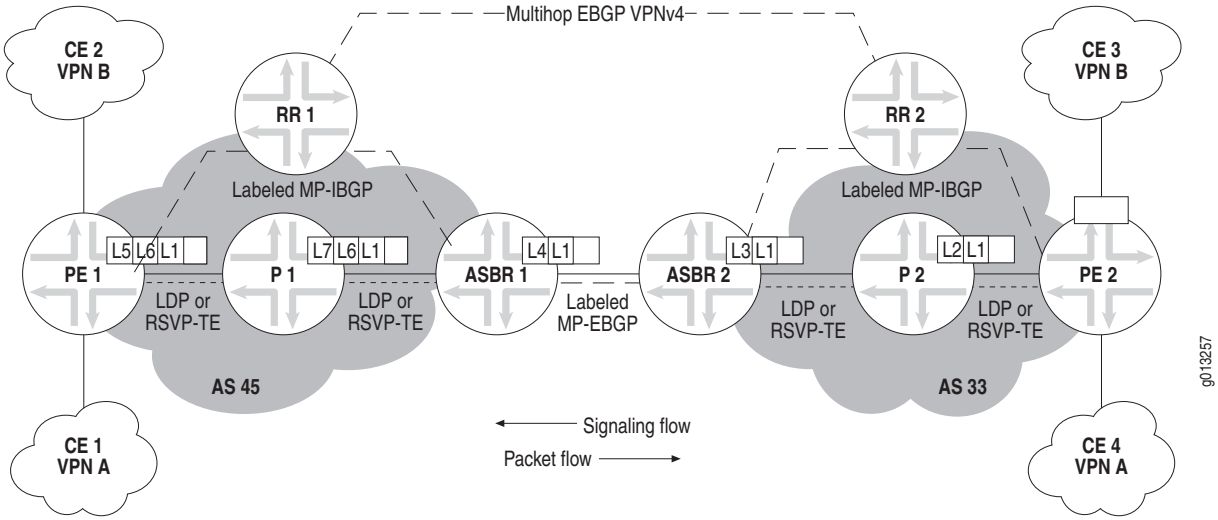
In contrast to the three-label stack scenario described previously, in a two-label stack scenario, BGP labeled unicast is not used inside the autonomous system. Instead, only LDP is used as the label distribution protocol. A PE router in one AS has a direct LSP to a PE in another AS, achieved by using LDP labels within the AS and BGP labels across the AS boundary.

For a two-label stack scenario to work, you must issue the **mpls ldp redistribute bgp** command on the ASBRs. This command enables the BGP prefixes to be advertised by LDP inside the autonomous systems. For more information on this command, see *Chapter 2, Configuring MPLS*.

Inter-AS Option C with Route Reflectors

When the BGP/MPLS VPN peer is a route reflector (Figure 79 on page 387), issue the **neighbor next-hop-unchanged** command to prevent the RR from rewriting the BGP next-hop attribute when the RR advertises routes to external neighbors. Issuing this command causes the VPN RR that is multihop peering with another RR in the AS to send the next hop unchanged for the VPN routes that it advertises.

Figure 79: Topology for Inter-AS Option C with Route Reflectors



neighbor next-hop-unchanged

- Use to prevent BGP from modifying the next hop sent to the BGP peer.
- Outbound route maps take precedence over this command, enabling prefixes that match the route map to be modified, regardless of this command.
- Takes effect immediately.
- Example

  host1:vr1(config-router-af)#**neighbor next-hop-unchanged 10.24.15.32**

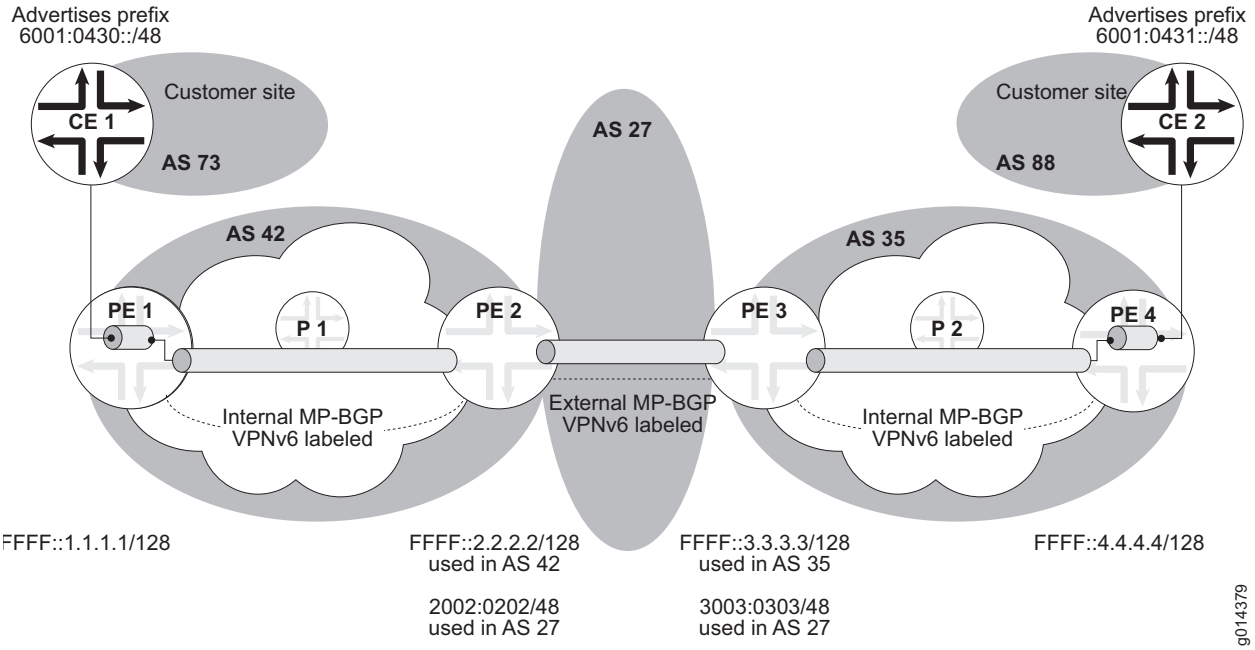
- Issuing this command automatically removes the **neighbor next-hop-self** configuration (enabled or disabled) on the peer or peer group. Issuing the **no** or **default** version of this command has no effect on the **neighbor next-hop-self** configuration.
- Use the **no** version to reenable BGP to modify the next hop.

Providing IPv6 VPN Services Across Multiple Autonomous Systems

The JUNOSe software supports inter-AS services for IPv6 VPNs in addition to IPv4 VPNs. See *Providing IPv4 VPN Services Across Multiple Autonomous Systems* on page 380 for more information about inter-AS services and IPv4 VPNs.

The JUNOSe software currently supports only 2547bis option B for IPv6 VPNs. This method—(described in BGP/MPLS IP VPNs—draft-ietf-l3vpn-rfc2547bis-03.txt (April 2005 expiration)—uses BGP to signal VPN labels between the AS boundary routers (Figure 80). The base MPLS tunnels are local to each AS. Stacked tunnels run from end to end between PE routers on the different ASs. This method enhances scalability, because only the BGP RIBs store all the inter-AS VPN routes.

Figure 80: Inter-AS IPv6 VPN Services



In Figure 80, the base tunnels between the PE routers are established in the IPv4 core networks with LDP or RSVP. The PE routers advertise IPv6 prefixes from the CE devices within their respective ASs as VPNv6 prefixes with MP-IBGP. For example, PE 1 advertises the CE 1 prefix 6001:0430::/48 over to PE 2 in its MP_REACH_NLRI attribute. The next-hop attribute in the update message is the PE 1 loopback address—the IPv4-mapped IPv6 address, FFFF::1.1.1.1/128.

PE 2 advertises 6001:0430::/48 by means of MP-EBGP to PE 3. The prefix is sent as a VPNv6-labeled prefix (2002:0202/48), with the default BGP next hop being the IPv4-mapped IPv6 address of the IPv4 interface going to PE 3.

For inter-AS services, in contrast to intra-AS services, JUNOSe software supports both IPv4 backbone and IPv6 backbone types of BGP next-hop encodings. The default BGP next-hop encoding used for IPv6 VPN inter-AS services is the one specified for the IPv4 backbone where IPv4-mapped IPv6 addresses are used. Alternatively, you might also configure the IPv6 backbone type of BGP next-hop encoding by configuring route maps that use native IPv6 addresses for the BGP next hop.

Using Route Targets to Configure VPN Topologies

You can use VRF import and export route targets to configure a variety of VPN topologies, such as full-mesh VPNs, hub-and-spoke VPNs, and overlapping VPNs.

Full-Mesh VPNs

In a full-mesh VPN, each site in the VPN can communicate with every other site in that same VPN. For example, in Figure 81, each site in VPN A can communicate with all other VPN A sites but not with the sites in VPN B.

Figure 81: Site Connectivity in a Full-Mesh VPN

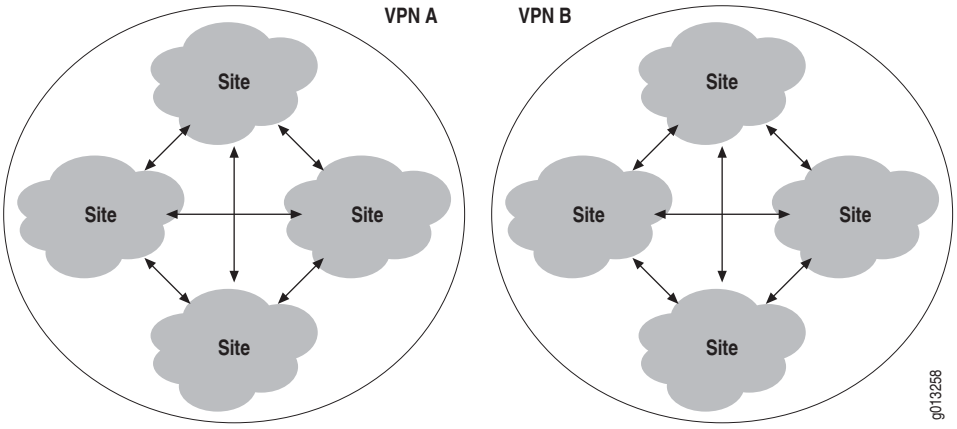
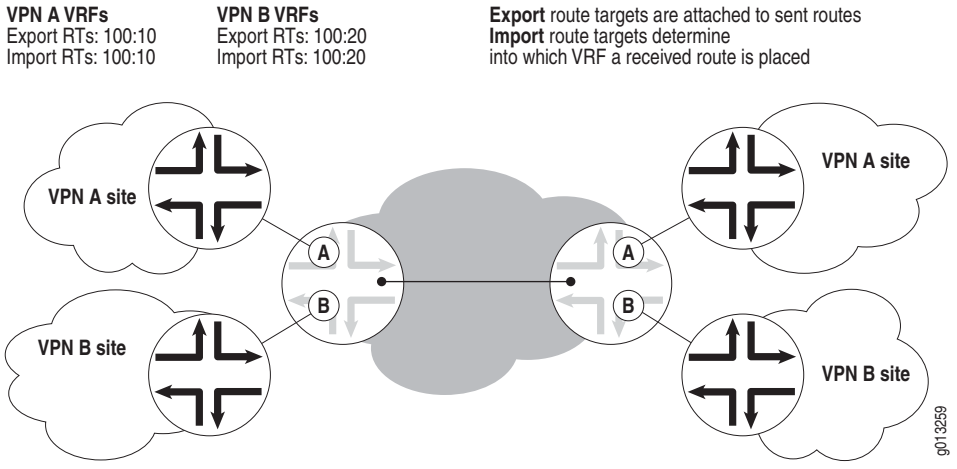


Figure 82 illustrates how you can configure the VRF import and export route targets to build a full-mesh VPN. Each VRF in VPN A has the same route target, 100:10, in their import list and export list. Each VPN A VRF accepts only received routes that have this route target attached. Because this route target is attached to each route advertised by VPN A VRFs, every site in VPN A accepts routes only from other sites in VPN A. The same principle applies to VPN B.

Figure 82: Route Target Configuration for a Full-Mesh VPN



Hub-and-Spoke VPNs

In a hub-and-spoke VPN, the spoke sites in the VPN can communicate only with the hub sites; they cannot communicate with other spoke sites, as shown in Figure 83.

Figure 83: Site Connectivity in a Hub-and-Spoke VPN

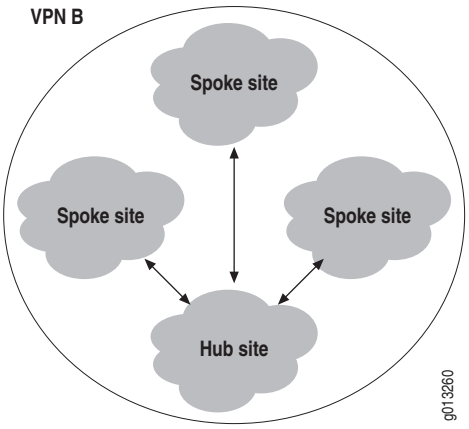
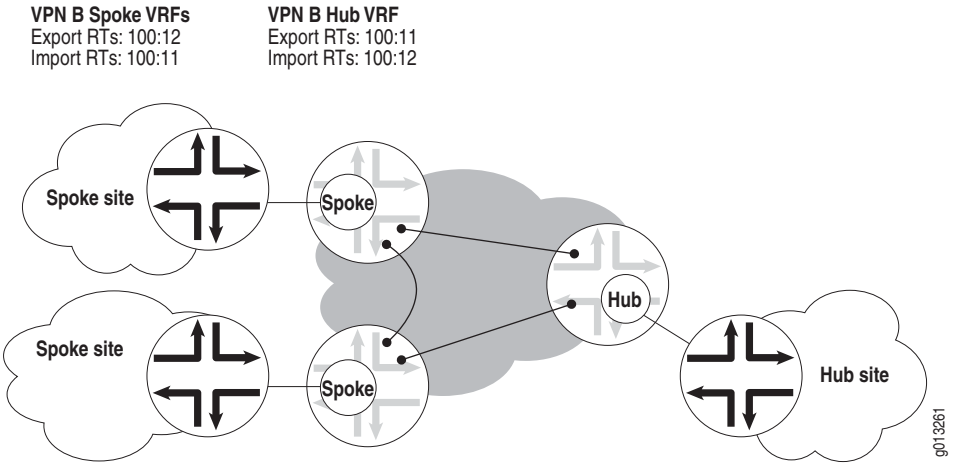


Figure 84 shows how to configure the VRF import and export route targets to build a hub-and-spoke VPN. Each spoke VRF has the same export route target, 100:12. The hub VRF has its import route target set to 100:12, so it accepts only routes from the spoke VRFs. Each spoke VRF has the same import route target, 100:11. Every route advertised by any spoke has an attached route target of 100:12. Because that route target does not match the import route target of any spoke, the spokes cannot accept any routes from another spoke. However, the hub VRF has an export route target of 100:11, so routes advertised by the hub do match the import target of each spoke and are accepted by all of the spokes.

Figure 84: Route Target Configuration for a Hub-and-Spoke VPN



Overlapping VPNs

In an overlapping VPN, a site is a member of more than one VPN. For example, in Figure 85, the middle site is a member of both VPN A and VPN B. In other words, that site can communicate with all other VPN A sites and all other VPN B sites. An overlapping VPN is often used to provide centralized services. The central site might contain DNS servers or WWW servers or management stations that need to be reachable from multiple VPNs. Overlapping IPv4 and IPv6 VPNs are supported by the same route-target mechanism.

Figure 85: Site Connectivity in an Overlapping VPN

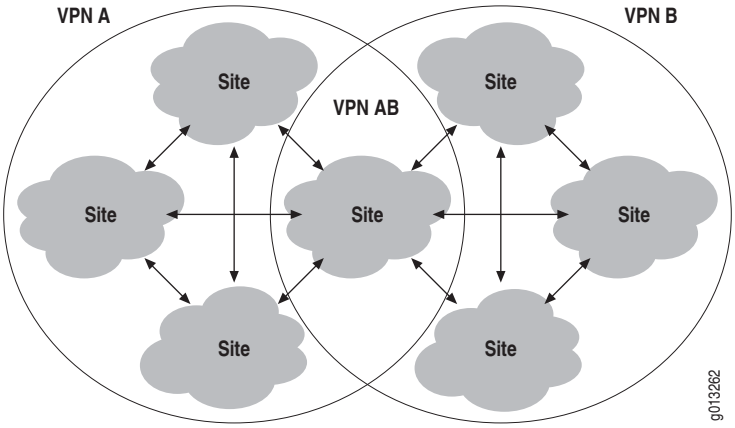
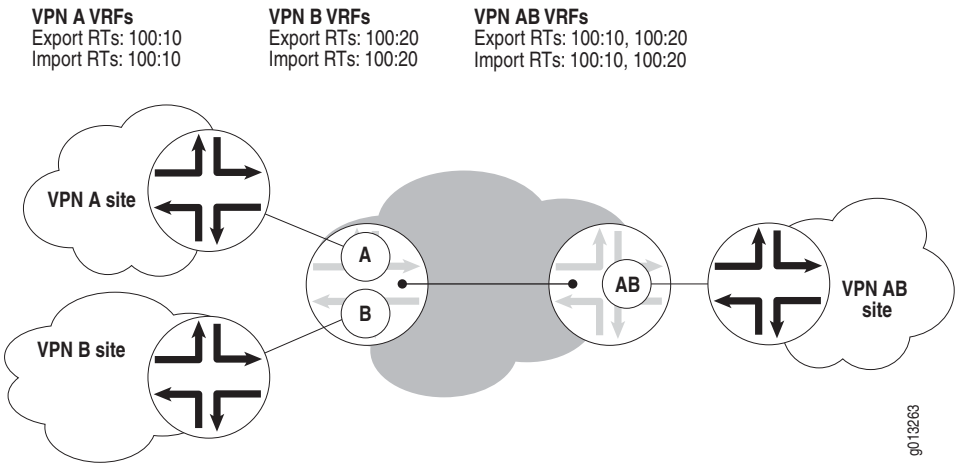


Figure 86 shows how to configure the VRF import and export route targets to build an overlapping VPN. In this example, the export and import route targets are different for VPN A and VPN B. Therefore, VPN A does not accept routes from VPN B and VPN B does not accept routes from VPN A.

The import route target list for the overlapping VPN AB includes both 100:10 and 100:20. VPN AB can therefore accept routes advertised by any site in either VPN A or VPN B. Because the VPN AB export route target list also includes both 100:10 and 100:20, every route advertised by VPN AB can be accepted by any site in either VPN A or VPN B.

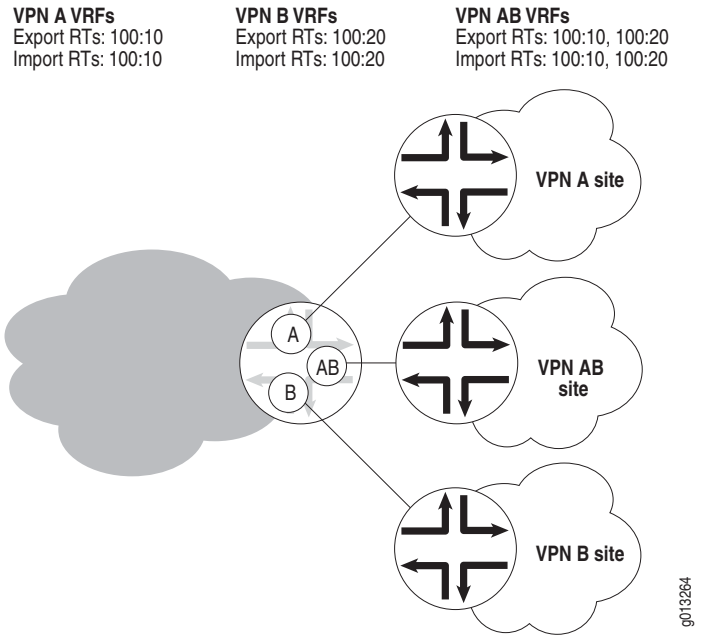
Figure 86: Route Target Configuration for an Overlapping VPN



A interesting special case of an overlapping VPN is when two VRFs on the same PE router belong to the same VPN as shown in Figure 87. The configuration of the VRF import and export route targets is the same as for the example in Figure 86.

If the export route target of one VRF (for example, the VPN AB VRF) matches the import route target of another VRF (for example, the VPN A VRF), then BGP routes are exported from one VRF to the other VRF; in this case from the VPN AB VRF to the VPN A VRF. Consequently, traffic that arrives in one VRF is forwarded out another VRF without going through the MPLS core network.

Figure 87: Overlapping VPNs on a Single PE



From a given CE router you can ping the local address of any VRF that has a VPN overlapping another VPN to which the CE router belongs.

To achieve this internally, the router obtains the source address as follows:

- If the next-hop interface is in the same VRF and the interface is numbered, the router uses the source address of the interface.
- If the next-hop interface is in the same VRF and the interface is unnumbered, the router uses either the source address of the interface it is pointing to or the router ID of the VRF.
- If the next-hop interface is in a different VRF, the router uses the source address of the VRF. If the router does not have a router ID value, the packet is discarded.

NOTE: The source address of the transmit interface is not used as the source address of the packet.

Constraining Route Distribution with Route-Target Filtering

In typical BGP configurations, you can use cooperative route filtering to reduce the amount of processing required for inbound BGP updates and the amount of BGP control traffic generated by BGP updates. Cooperative route filtering works by having the remote peer install a BGP speaker's inbound route filter as its own outbound route filter. This filtering causes the remote peer to advertise only those routes that the local peer can accept.

For BGP/MPLS VPNs, route-target filtering is a better approach. Route-target filtering controls the distribution of BGP routes based on the VPNS (indicated by the route-target extended communities) to which peer routers belong. PE routers use the MP_REACH_NLRI and MP_UNREACH_NLRI attributes in BGP updates to exchange information about each router's route-target membership.

The PE router subsequently advertises VPN NLRI—the routing information carried in MP-BGP update messages—only to peers that are members of a route target that is associated with the VPN route. The VPN routes flow in the opposite direction to the route-target membership information.

Route-target filtering works across multiple ASs and with asymmetric VPN topologies, such as a hub-and-spoke. Route-target filtering can reduce the size of the BGP routing table in PE routers, as well as the amount of VPN NLRI exchange traffic between routes in the VPN. Route-target filtering also reduces router memory requirements by reducing the amount of routing information stored and propagated. For example, route reflectors scale according to the total number of VPN routes present in their network. With route-target filtering, you can reduce the scaling requirements of the reflectors by restricting the number of VPN routes they must process to only those VPN routes actually used by the route reflector clients.

Applications such as BGP/MPLS VPNs, L2VPNs, and VPLS all use route targets as part of their route reachability information, and can therefore employ route-target filtering and potentially accrue the benefits of reduced traffic and smaller routing tables.

Exchanging Route-Target Membership Information

BGP peers exchange route-target membership information in the following sequence:

1. When the BGP peers negotiate the BGP multiprotocol extensions capability during the establishment of a BGP session, they indicate support for the route-target address family by including the (AFI, SAFI) value pair for the route-target membership NLRI (RT-MEM-NLRI) attribute. This pair has an AFI value of 1 and a SAFI value of 132.

2. If the capability is successfully negotiated, BGP speaker Router A expresses its interest in a VPN route target by advertising to its peers the RT-MEM-NLRI attribute that contains the particular route target. This attribute is represented as a prefix in the following format:

   AS number:*route-target extended community*/*prefix length*

   - *AS number*—Number of the originating AS
   - *route-target extended community*—Two-part number identifying the route target extended community. Consists of *number1*:*number2*, where:
     - *number1*—Autonomous system (AS) number or an IP address
     - *number2*—Unique integer; 32 bits if *number1* is an AS number; 16 bits if *number1* is an IP address
   - *prefix length*—Length of the prefix. A prefix less than 32 or greater than 96 is invalid. However, the prefix for the Default-RT-MEM-NLRI attribute is an exception to this rule. For the Default-RT-MEM-NLRI attribute, 0 is a valid prefix length.

   For example, 100:100:53/36 is a valid RT-MEM-NLRI.

3. Remote peers of Router A use the route-target membership advertised by Router A to filter their VPN routes that are outbound to Router A. A peer advertises a VPN route to Router A only when one of the following conditions is true

   - Router A advertised a default route-target membership.
   - Router A advertised membership in any of the route targets associated with the VPN route.

4. Router A then receives and processes the RT-MEM-NLRI attributes sent by its peers to determine which VPN routes it advertises to the peers.

BGP speakers advertise and withdraw the RT-MEM-NLRI attribute in MP-BGP update messages. BGP speakers ignore RT-MEM-NLRI attributes received from peers that have not successfully negotiated this capability with the speaker.

If dynamic negotiation for the route-refresh capability is enabled, BGP negotiates the route-refresh capability for the RT-MEM-AFI-SAFI address family when a peer is activated in that family. As a consequence, you can use the **clear ip bgp soft** command to refresh the RT-MEM-NLRI routes in the BGP speaker's Adj-RIBs-Out table.

The usefulness of BGP VPN route-target filtering depends on the sparseness of route target membership among the VPN sites. In configurations where VPNs are members of many route target communities—that is, route target membership is dense—the amount of VPN NLRI exchange traffic is about the same regardless of whether route-target filtering is configured.

Receiving and Sending RT-MEM-NLRI Routing Updates

RT-MEM-NLRI routing updates are processed in the following sequence:

1. During the initial RT-MEM-NLRI route exchange that takes place when a session with a peer is being brought up, BGP sends an End-of-RIB marker for RT-MEM-AFI-SAFI that signals it has finished advertising route-target membership information.

2. Remote peers interpret the End-of-RIB marker for RT-MEM-AFI-SAFI to mean that the BGP speaker has advertised all of its route target-memberships. If the BGP speaker does not receive an End-of-RIB marker for RT-MEM-AFI-SAFI from a remote BGP peer in the context of the route-target address family, by default the local BGP speaker waits for 60 seconds before timing out.

3. The BGP speaker then starts advertising its VPN routes. The routes are passed through the outbound route-target membership filters for that peer.

4. When a BGP speaker receives a RT-MEM-NLRI update message, it re-evaluates the advertisement status of VPN routes that match the corresponding route target in the peer's Adj-RIBS-Out table. This can result in an incremental update that advertises or withdraws some routes for the VPN.

You can use the **bgp wait-on-end-of-rib** command to specify how long BGP waits for the End-ofRIB marker from route-target peers.

When the route-refresh capability has been negotiated for the route-target address family. BGP handles route-refresh messages for the RT-MEM-AFI-SAFI by resending all RT-MEM-NLRI routes to the remote peer

You can use the **neighbor maximum-prefix** command to specify the maximum number of prefixes that the speaker can receive from a BGP peer.

Route-target filtering generally follows the standard BGP rules for route advertisement to determine when to advertise RT-MEM-NLRI prefixes that have been received from BGP peers. Table 33 lists the destinations that the prefixes are advertised to based on their source. In this table, client-to-client reflection is enabled and the source and destination peers are not the same.

Table 33: Route-Target Filtering Advertisement Rules for Routes Received from Peers

Routes Received From	Advertise to IBGP Route Reflector Client?	Advertise to IBGP Route Reflector Nonclient?	Advertise to EBGP Peer?	Advertise to EBGP Confederation Peer?
IBGP route reflector client	Yes	Yes	Yes	Yes
IBGP route reflector nonclient	Yes	No	Yes	Yes
EBGP peer	Yes	Yes	Yes	Yes
EBGP confederation peer	Yes	Yes	Yes	Yes

Advertising to IBGP clients varies from the standard advertisement rules in terms of path attribute modifications. When locally originated RT-MEM-NLRI routes are advertised to IBGP route reflector clients, BGP does the following:

- Sets the originator ID as the router ID of the advertising router.
- Sets the next hop as the local address of the session.

This behavior is useful when the route reflector does not advertise the Default-RT-MEM-NLRI route.

When locally originated RT-MEM-NLRI routes are advertised to IBGP route reflector nonclients, the route from the client is advertised to the nonclient peer when the best path route is advertised by a nonclient but an alternative route from a client exists. This behavior signals the client's interest in the route target routes that were not selected as the best path.

You cannot filter RT-MEM-NLRI routes with inbound policies or outbound policies, because policy items cannot currently match a RT-MEM-NLRI prefix (origin AS number:route target). However, you can filter route-target filtering routes with policies that include items that match on other BGP attributes, such as the extended community attached to the route-target filtering route.

bgp wait-on-end-of-rib

- Use to configure how long BGP waits to receive End-of-RIB markers from route-target address family peers.
- The wait interval applies to all route-target address family peers.
- This command takes effect immediately.
- Example

  host1(config-router)#**address-family route-target signaling**
  host1(config-router-af)#**bgp wait-on-end-of-rib 360**

- Use the **no** version to restore the default wait interval, 60 seconds.

neighbor maximum-prefix

- Use to control how many prefixes can be received from a neighbor.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- By default, BGP checks the maximum prefix limit only against accepted routes. You can specify the **strict** keyword to force BGP to check the maximum prefix against all received routes. The accepted and received routes will likely differ when you have configured inbound soft reconfiguration and route filters for incoming traffic.
- This command takes effect immediately. To prevent a peer from continually flapping, when it goes to state idle because the maximum number of prefixes has been reached, the peer stays in state idle until you use the **clear ip bgp** command to issue a hard clear.
- Example

  host1(config-router)#**address-family route-target signaling**
  host1(config-router-af)#**neighbor maximum-prefix 10.1.2.3 100**

- Use the **no** version to remove the maximum number of prefixes.

Conditions for Advertising RT-MEM-NLRI Routes

The following conditions must be met for routes in the route-target address family to be advertised to a BGP peer:

1. The BGP peers have successfully negotiated the route-target address family.
2. The import route-target list for the IPv4 VRF is not empty or is transitioning to empty.

In a VRF, a RT-MEM-NLRI attribute represented by (origin AS number:route target) is advertised for every route target added to the VRF's route target import list when the preceding conditions have been met.

A withdrawal for the RT-MEM-NLRI attribute is generated when the route target is removed from this VRF's import list.

Advertising a Default Route

You can configure BGP to send a default route to indicate that the speaker accepts routes for any VPNs associated with any route target. For example, this might be desirable for a route reflector advertising to one of its PE router clients, or when a VPN provider is migrating the network to route-target filtering but one or more PEs in the provider's network do not support this feature.

When you configure the default route, the RT-MEM-NLRI attribute contains 0:0:0/0 as the Default-RT-MEM-NLRI. This 4-byte prefix contains only the local (origin) AS number field, set to zero.

By default, BGP does not generate or advertise the Default-RT-MEM-NLRI route. You can use the **default-information originate** command to generate the Default-RT-MEM-NLRI route and send it to all peers. You can use the **neighbor default-originate** command generate the route and send it to a particular peer group. The configuration must be the same for all members of the peer group.

A BGP speaker sends the Default-RT-MEM-NLRI route only to the peers with which it has negotiated the route-target filtering capability. Any other peers are considered to be unaware of this capability and have no use for that route.

default-information originate

- Use in the route-target address family to cause a BGP speaker (the local router) to send the Default-RT-MEM-NLRI route (0:0:0/0) to all peers for use as a default route.
- Use the **route-map** keyword to specify outbound route maps to apply to the default route. The route map can modify the attributes of the default route.
- This command takes effect immediately. However, if the contents of the route map specified with this command change, the new route map may or may not take effect immediately. If the **disable-dynamic-redistribute** command has been configured, you must issue the **clear ip bgp redistribution** command to apply the changed route map.
- Outbound policy configured for the neighbor (using the **neighbor route-map out** command) is applied to default routes that are advertised because of the **default-information originate** command.
- Policy specified by a route map with the **default-information originate** command is applied at the same time as the policy for redistributed routes, before any outbound policy for peers.
- Example

  host1(config-router)#**router address-family route-target**
  host1(config-router-af)#**default-information originate**

- Use the **no** version to restore the default, preventing the redistribution of default routes.

neighbor default-originate

- Use in the route-target address family to cause a BGP speaker (the local router) to send the Default-RT-MEM-NLRI route (0:0:0/0) to a peer group for use as a default route.
- Use the **route-map** keyword to specify outbound route maps to apply to the default route. The route map can modify the attributes of the default route.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Outbound policy configured for the neighbor (using the **neighbor route-map out** command) is not applied to default routes that are advertised because of the **neighbor default-originate** command.
- This command takes effect immediately.

- Example

  host1(config)#**router bgp 100**
  host1(config-router)#**router address-family route-target**
  host1(config-router-af)#**neighbor default-originate**

- Use the **no** version to prevent the default route from being advertised by BGP. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.

Route Selection When Route-Target Filtering Is Enabled

When route-target filtering is enabled for a peer, BGP applies outbound filters to initially prevent the speaker from advertising any VPN routes to the peer.

If the BGP speaker subsequently receives a Default-MEM-NLRI route from a peer, BGP applies outbound filters for the peer to prevent route-target filtering from suppressing any VPN routes sent to the peer.

BGP follows the standard route selection process to find the route-target filtering best path for RT-MEM-NLRI routes received from other autonomous systems. The selection is based on the AS path and other MP-NLRI path-attributes attached to the route.

The route-target membership information, which includes the route target and the originator AS number, enables BGP speakers to use the standard path selection rules to remove duplicate, less-preferred paths from the total set of paths to route-target membership peers.

For RT-MEM-NLRI routes that originated within the local AS and are received from an IBGP peer, BGP considers the route-target filtering best path to be the set of all available IBGP paths for the RT-MEM-NLRI prefix. BGP then sets outbound route filters so that VPN routes that match the route target are sent to all IBGP peers that advertised the RT-MEM-NLRI route. This behavior does not affect how the BGP speaker in turn advertises the RT-MEM-NLRI routes.

When BGP selects a RT-MEM-NLRI route from a peer as the best path for the RT-MEM-NLRI prefix, BGP modifies the outbound filters to enable the speaker to advertise to that peer all VPN routes that correspond to that route target. These filters affect the subsequent calculation of the peer's Adj-RIBs-Out entries.

EBGP confederation peers are treated as IBGP peers when the BGP speaker is selecting the route-target filtering best path. When the BGP speaker advertises routes, then the EBGP confederation peers are treated normally, as EBGP peers.

You can control the maximum number of received EBGP best paths that are considered for path selection. The **external-paths** command limits external route target membership, thus controlling the number of EBGP peers that receive the route target VPN routes referenced by the RT-MEM-NLRI route. BGP ignores routes received from the peer after the limit specified with the **external paths** command is reached.

Configuring Route-Target Filtering

To configure route-target filtering:

1. Enable the BGP routing process in the specified AS.

   The AS number identifies the PE router to other BGP routers.

   host1(config)#**router bgp 738**

2. Configure the peers for the BGP speaker. Use **neighbor** commands to specify the PE router peers to which BGP advertises routes and to configure any additional BGP attributes.

   host1(config-router)#**neighbor 10.2.2.2 remote-as 45**
   host1(config-router)#**neighbor 10.2.2.2 update-source loopback 0**
   host1(config-router)#**neighbor 10.2.2.2 next-hop-self**

3. Create the route-target address family to configure the router to use BGP signaling to exchange the RT-MEM-NLRI attribute with peer routers.

   Optionally, you can use the **signaling** keyword with the **address-family** command when you configure the route-target address family to specify BGP signaling of reachability information. Currently, you can omit the **signaling** keyword with no adverse effects.

   host1(config-router)#**address-family route-target signaling**

4. Activate the neighbors that routes of the route-target address family are exchanged with for this BGP session. The neighbors must first be created in the default IPv4 unicast address family.

   host1(config-router-af)#**neighbor 10.2.2.2 activate**
   host1(config-router-af)#**neighbor 10.2.2.2 next-hop-self**

5. (Optional) Configure BGP to send a Default-MEM-NLRI route for all peers in the address family or for a specific peer or peer group in the address family.

   host1(config-router-af)#**default-information originate**
   or
   host1(config-router-af)#**neighbor 10.2.2.2 default-originate**

6. Set the maximum number of received external BGP paths that can be accepted for route-target signaling.

   host1(config-router-af)#**external-paths 2**

7. Configure any additional address family parameters desired for the session.

external-paths

- Use to set the maximum number of received external BGP best paths allowed for route-target signaling.
- Specify a value in the range 1–255; the default value is 1.
- This command takes effect immediately; it does not bounce the session.

- This command applies to only the route-target address family.
- Example 1

  host1(config-router)#**external-paths 45**

- Example 2

  host1:vr1(config-router-af)#**external-paths 45**

- Use the **no** version to restore the default value, 1.

Multicast Services over VPNs

For information on VPN multicast services, see *Creating Multicast VPNs* in *JUNOSe Multicast Routing Configuration Guide, Chapter 7, Configuring PIM for IPv4 Multicast*.

Configuring BGP VPN Services

To configure a router to provide BGP VPN services, you must perform some tasks once per PE router and some tasks for each VRF on the PE router.

VRF Configuration Tasks

To configure a VRF to provide BGP VPN services:

1. Create the VRF.

   host1(config)#**virtual-router vr1**
   host1:vr1(config)#**ip vrf vrfA**

2. Assign a route distinguisher to the VRF.

   host1:vr1(config-vrf)#**rd 100:100**
3. Set the route-target import and route-target export lists for the VRF.

   host1:vr1(config-vrf)#**route-target import 100:1**
   host1:vr1(config-vrf)#**route-target export 100:1**

4. (Optional) Set import and export maps for the VRF.

   host1:vr1(config-vrf)#**import map Another-route-map**
   host1:vr1(config-vrf)#**export map A-route-map**
   host1:vr1(config-vrf)#**exit**

5. Assign interfaces for PE-to-CE links to the VRF from outside or inside the VRF context:

   host1:vr1(config)#**interface gigabitEthernet 1/0**
   host1:vr1(config-if)#**ip vrf forwarding vrfA**
   host1:vr1:vrfA(config-if)#**ip address 10.16.2.77 255.255.255.0**
   host1:vr1:vrfA(config-if)#**exit**

   or

   host1:vr1(config)#**virtual-router :vrfA**
   host1:vr1:vrfA(config)#**interface gigabitEthernet 1/0**

NOTE: You can also use the **ip vrf forwarding** command to specify secondary route lookup at the parent (global) level, in the event the original lookup does not yield any results.

6. Use either of the following methods to establish how the VRF learns routes to customer sites:

   - Create static routes to the customer site in the VRF by one of the following methods:

     host1(config)#**virtual-router vr1**
     host1:vr1(config)#**ip vrf vpnA**
     host1:vr1(config-vrf)#**ip route vrf vrfA 10.3.0.0 255.255.0.0 10.1.1.1**
     host1:vr1(config-vrf)#**ip route vrf vrfA 10.12.0.0 255.255.0.0 10.1.1.1**

     or

     host1(config)#**virtual-router vr1:vrfA**
     host1:vr1:vrfA(config)#**ip route 10.3.0.0 255.255.0.0 10.1.1.1**
     host1:vr1:vrfA(config)#**ip route 10.12.0.0 255.255.0.0 10.1.1.1**

   - Configure an IGP on the VRF to learn routes from the CE router.

     See *Configuring IGPs on the VRF* on page 419 for examples.

   - Configure a PE-to-CE EBGP session.

     See *Configuring PE-to-CE BGP Sessions* on page 426 for information about configuring EBGP.

7. (Optional) Configure the router to generate a label for each different FEC pointed to by a BGP route in the VPN.

   host1:vr1(config-vrf)#**ip mpls forwarding-mode label-switched**

8. (Optional) For carrier-of-carriers VPNs, configure carrier-of-carriers mode in the provider carrier's PE router that connects to the customer carrier's network.

   host1:vr1:VrfA(config)#**mpls topology-driven-lsp**

   See *Carrier-of-Carriers IPv4 VPNs* on page 451 for information about configuring carrier-of-carriers VPNs.

PE Router Configuration Tasks

To configure a PE router to provide BGP VPN services:

1. Configure PE-to-PE LSPs.

   See *Chapter 2, Configuring MPLS*, for information about configuring LSPs.

2. Enable BGP routing.

   host1:vr1(config)#**router bgp 100**

3. (Optional) Disable automatic route-target filtering.

   host1:vr1(config-router)#**no bgp default route-target filter**

4. Configure PE-to-PE BGP sessions.

   a. Create the PE-to-PE session.

      host1:vr1(config)#**router bgp 100**
      host1:vr1(config-router)#**neighbor 192.168.1.158 remote-as 100**

   b. Create the VPN-IPv4 address family.

      host1:vr1(config-router)#**address-family vpnv4**

   c. Activate the PE-to-PE session in the VPN-IPv4 address family.

      host1:vr1(config-router-af)#**neighbor 192.168.1.158 activate**
      host1:vr1(config-router-af)#**exit-address-family**

   d. (Optional) Enable the BGP speaker to check the reachability of indirect next hops when selecting the best VPN-IPv4 route to a prefix.

      host1:pe1(config-router-af)#**check-vpn-next-hops**

5. Configure PE-to-CE BGP sessions.

   a. Enable and configure BGP:

      host1:vr1(config)#**router bgp 100**

      See *Chapter 1, Configuring BGP Routing*, for more information about configuring BGP.

   b. Specify the IPv4 unicast address family for each VRF:

      host1:vr1(config-router)#**address-family ipv4 unicast vrf vrfA**

c. Configure the method of route advertisement by doing one of the following:

- Use **neighbor** commands to specify peers to which BGP advertises the routes:

  host1:vr1(config-router)#**neighbor 10.12.13.0 remote-as 200**

- Use **network** commands or the **redistribute static** command to make BGP advertise static routes to customers.

  host1:vr1(config-router)#**network 10.3.0.0 mask 255.255.0.0**
  host1:vr1(config-router)#**redistribute static**

- Use **redistribute** commands to make BGP advertise IGP routes to customers.

  host1:vr1(config-router)#**redistribute ospf**

6. (Optional) Configure an AS override.

   See *Using a Single AS Number for All CE Sites* on page 428 for examples.

7. (Optional) Force the BGP speaker to accept routes that have the speaker's AS number in its AS path.

   host1:vr1(config-router)#**bgp enforce-first-as**

Creating a VRF

Access the desired virtual router context; then create the VRF(s) for that VR.

host1(config)#**virtual-router vr1**
host1:vr1(config)#**ip vrf vrfA**

ip vrf

- Use to create a VRF or access VRF Configuration mode to configure a VRF.
- You must specify a route distinguisher after you create a VRF. Otherwise, the VRF will not operate.
- Example

  host1:vr1(config)#**ip vrf vrfA**

- Use the **no** version to remove a VRF.
- Use the **wait-for-completion** keyword with the **no** version if you require a synchronous, deterministic deletion of a VRF, such as when executing Telnet or console commands by means of an external script. If you do not issue the **wait-for-completion** keyword in these circumstances, an **ip vrf** command issued as soon as the prompt appears might fail because the router is still deleting the VRF. You can specify a period during which the CLI waits before it returns a prompt. If you do not specify a wait time, then the CLI does not return a prompt until the operation completes. You can press Ctrl + c to break out of the wait period early.

Specifying a Route Distinguisher

The route distinguisher enables you to establish unique VPN-IPv4 addresses to accommodate the possibility that more than one VPN might use the same IP address from their private address spaces.

rd

- Use to specify a route distinguisher to a VRF.
- You can specify either an AS number or an IP address as the first part of the route distinguisher. Specify some unique integer as the second part.
- You must specify a route distinguisher for a VRF. Otherwise, the VRF will not operate.
- After you have configured the route distinguisher, you can change it only by removing and recreating the VRF.
- Example

  host1:vr1(config-vrf)#**rd 100:100**

- There is no **no** version.

Defining Route Targets for VRFs

BGP uses an extended-community attribute, the *route target*, to filter appropriate VPN routes into the correct VRFs. You configure the *export list* on the VRF to specify export route targets. When BGP advertises a route from this VRF's forwarding table, it associates the list of export route targets with the route and includes this attribute in the update message that advertises the route.

You also configure a route-target *import list* on each VRF to specify import route targets. When a PE router receives a route, BGP compares the route target list associated with the route (and carried in the update message) with the import list associated with each VRF configured in the PE router.

For VPN-IPv4 routes received from another PE router, if *any* route target in the export list matches a route target in a VRF's import list, then the route is installed in that VRF's forwarding table.

For the most common configuration, do the following:

1. Allocate one route-target extended-community value per VPN.
2. Define the route-target import list and a route-target export list to include only the route-target extended-community values for the VPN(s) to which the VRF belongs:

   host1:vr1(config-vrf)#**route-target export 777:100**
   host1:vr1(config-vrf)#**route-target import 777:100**

   If the import and export lists are identical, you can use the **both** keyword to define the lists simultaneously:

   host1:vr1(config-vrf)#**route-target both 777:105**

A route-target export list can be modified on the sending PE router by an export map or outbound routing policy. It can be modified on the receiving PE router by an import map or inbound routing policy.

route-target

- Use to create—or add to—lists of VPN extended communities for a VRF that determine whether a route is imported into a VRF.
- An export list defines a route-target extended community; routes having any route target in their export list that matches a route target in a VRF's import list are installed in the VRF's forwarding table.
- An import list defines a route-target extended community; only routes that have at least one matching route target in their associated export list can be installed into the VRF's forwarding table.
- If the import and export lists are identical, use the **both** keyword to define both lists simultaneously.
- You can add only one route target to a list at a time.
- Example

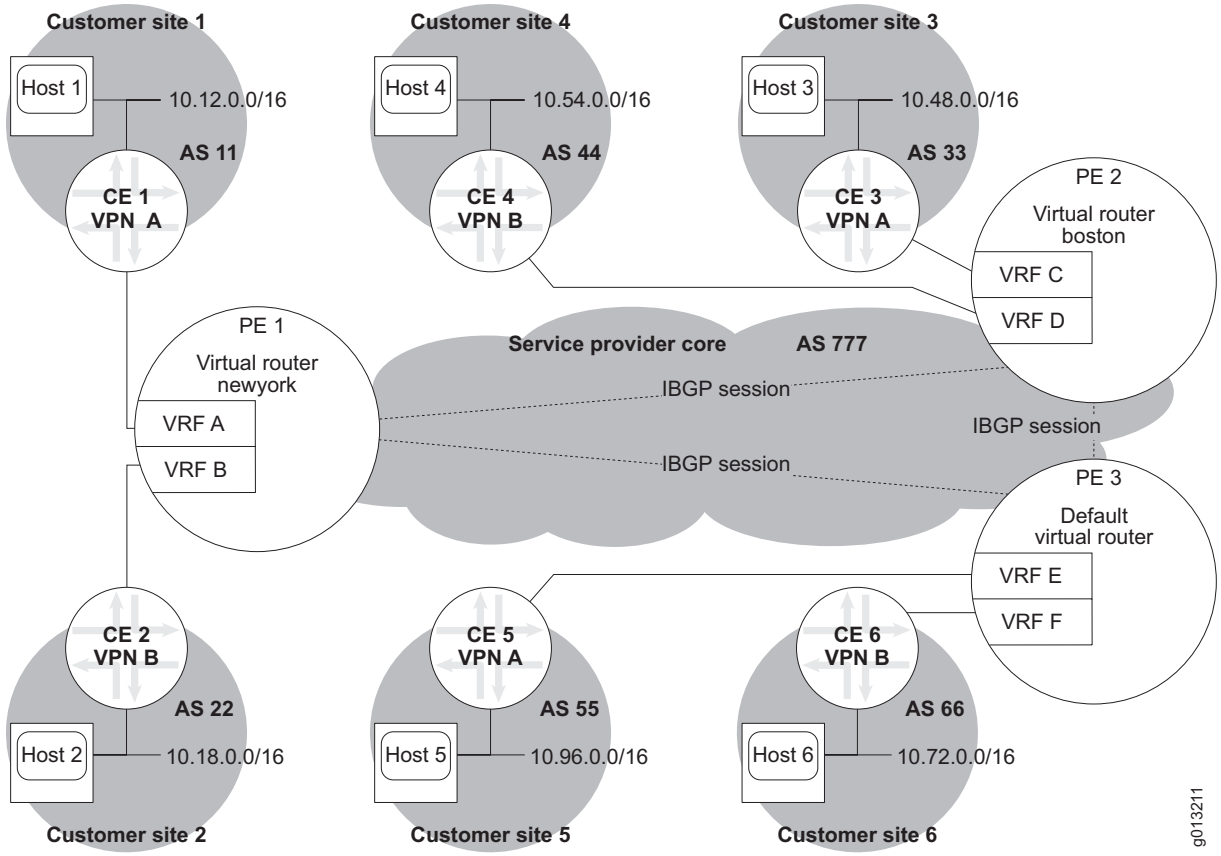
  host1:vr1(config-vrf)#**route-target export 100:1**
  host1:vr1(config-vrf)#**route-target import 100:1**

- Use the **no** version to remove a route target from the import list, the export list, or both lists.

Example: Fully Meshed VPNs

In a fully meshed VPN, each site in the VPN can reach every other site in the VPN. Figure 88 illustrates a situation with two fully meshed VPNs, VPN A and VPN B. VPN A includes Customer Sites 1, 3, and 5 through VRFs A, C, and E. VPN B includes Customer Sites 2, 4, and 6 through VRFs B, D, and F.

Figure 88: Fully Meshed VPNs



BGP sessions exist between PE 1 and PE 2, PE 2 and PE 3, and PE 3 and PE 1. The MPLS paths through the service provider core are omitted for clarity.

To configure route targets for this fully meshed scenario, you specify the same route target for the import list and export list on all VRFs in VPN A. The VRFs in VPN B use a different route target, but it is the same for the import list and export list for all.

Route-target configuration on PE 1:

```
host1(config)#virtual-router newyork
host1:newyork(config)#ip vrf vrfA
host1:newyork(config-vrf)#route-target both 777:1
host1:newyork(config-vrf)#exit
host1:newyork(config)#ip vrf vrfB
host1:newyork(config-vrf)#route-target both 777:2
```

Route-target configuration on PE 2:

```
host2(config)#virtual-router boston
host2:boston(config)#ip vrf vrfC
host2:boston(config-vrf)#route-target both 777:1
host2:boston(config-vrf)#exit
host2:boston(config)#ip vrf vrfD
host2:boston(config-vrf)#route-target both 777:2
```

Route-target configuration on PE 3:

```
host3(config)#ip vrf vrfE
host3(config-vrf)#route-target both 777:1
host3(config-vrf)#exit
host3(config)#ip vrf vrfF
host3(config-vrf)#route-target both 777:2
```

Example: Hub-and-Spoke VPN

In one type of a hub-and-spoke design, only the hub site can reach every site in the VPN. All other sites—spokes—can reach only the hub site. (More complex hub-and-spoke designs are possible, but require additional configuration and route targets to achieve.) In Figure 89, Customer Site 1 is the hub site for VPN A. As such it can reach both spokes, Customer Sites 2 and 3 through VRF A. Customer Site 2 can reach only the hub, customer 1, through VRF C. Customer Site 3 can reach only the hub, customer 1, through VRF E.

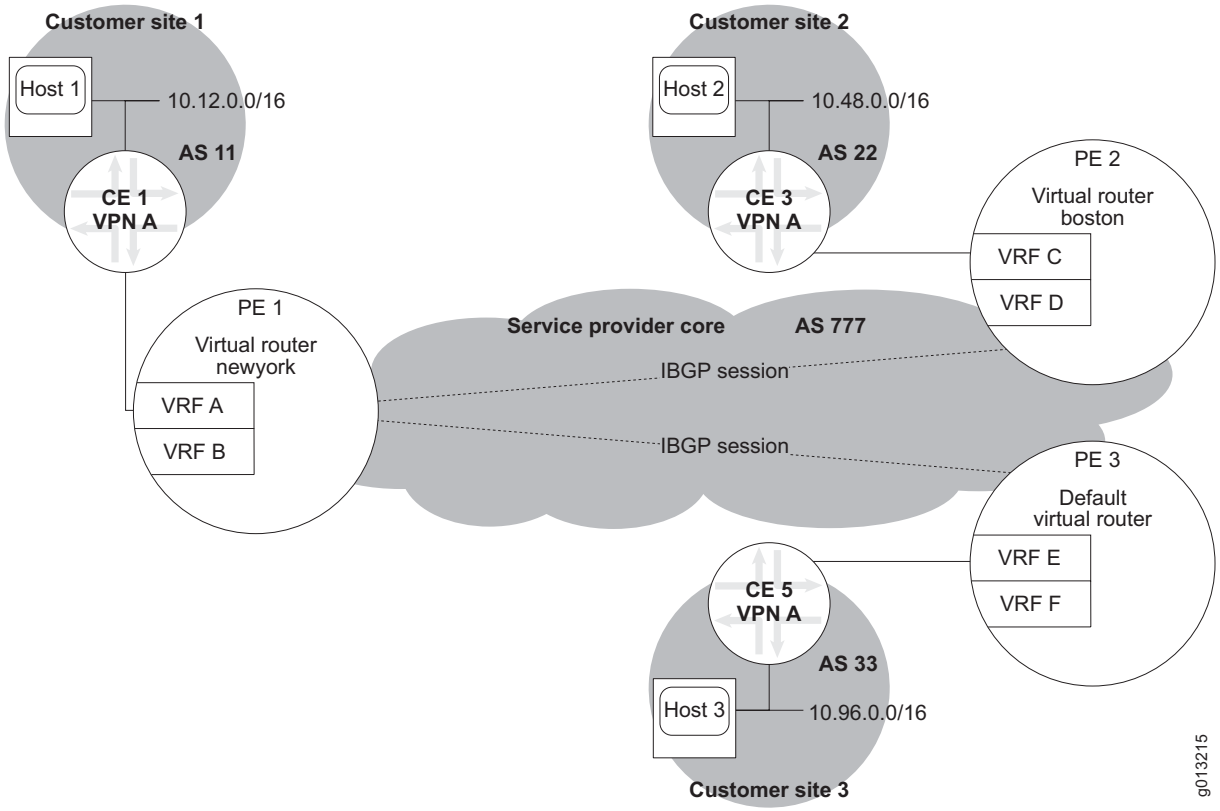
BGP sessions exist between PE 1 and PE 2 and between PE 1 and PE 3. In most situations, BGP itself is fully meshed, but that level of complexity is not necessary for this example. The MPLS paths through the service provider core are omitted for clarity.

To configure route targets for this hub and spoke, you specify different import and export route targets on the hub VRF. On the spoke VRFs, you switch these route targets.

Route-target configuration on PE 1:

```
host1(config)#virtual-router newyork
host1:newyork(config)#ip vrf vrfA
host1:newyork(config-vrf)#route-target export 777:25
host1:newyork(config-vrf)#route-target import 777:50
```

Figure 89: Hub-and-Spoke VPN



Route-target configuration on PE 2:

```
host2(config)#virtual-router boston
host2:boston(config)#ip vrf vrfC
host2:boston(config-vrf)#route-target export 777:50
host2:boston(config-vrf)#route-target import 777:25
```

Route-target configuration on PE 3:

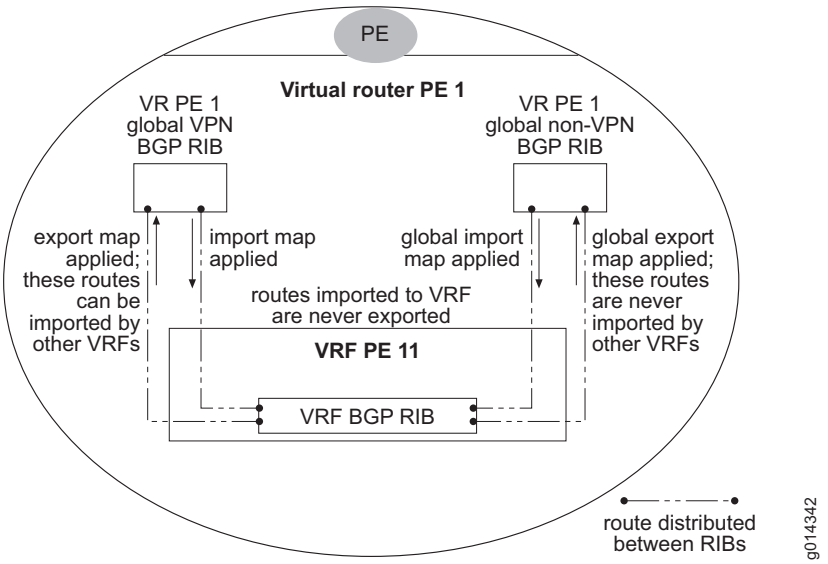
```
host3(config)#ip vrf vrfE
host3(config-vrf)#route-target export 777:50
host3(config-vrf)#route-target import 777:25
```

This configuration ensures that when VRF E on PE 3 receives an update message from PE 1, BGP installs the advertised route only if it has a route target of 25. Routes from PE 2 have a route target of 50, and cannot be installed. Similarly, when VRF C on PE 2 receives an update message from PE 1, BGP installs the advertised route only if it has a route target of 25. Routes from PE 3 have a route target of 50, and cannot be installed. When PE 1 receives updates from either PE 2 or PE 3, the routes have a route target of 50, match VRF A's import list, and are installed in VRF A's forwarding table.

Setting Import and Export Maps for a VRF

The combination of the route-target export list of VRF A and the route-target import list of VRF B determines whether routes from VRF A are distributed to VRF B. You can provide finer-grained control of route distribution by associating any combination of export, import, global export, and global import maps with VRFs. As shown in Figure 90, a route is distributed (leaked) between RIBs and its attributes are changed as specified in the route map when the map returns an accept message. If the map returns a deny message, then the route is not distributed.

Figure 90: Import and Export Maps



Both IPv4 and IPv6 VPNs are supported. You can specify that only IPv4 or only IPv6 routes are imported or exported. By default, the import or export map applies to both kinds of routes. You can configure some maps to apply to IPv4 routes and different maps to apply to IPv6 routes.

When the name or the contents of a route map change, BGP automatically waits for a nonconfigurable hold-down interval of 30 seconds and then re-imports or re-exports the appropriate routes using the modified route map.

Even when suppressed by an aggregate or auto-summary route, the more specific routes are distributed. Aggregation and auto-summarization take place in each VRF independently. For example, a route that is imported into a VRF is only aggregated in that VRF if an aggregate address has been configured in the context of the BGP address family for that VRF.

Routes maintain their type when exported. Private prefixes are exported without being converted into public prefixes. Consequently the prefix of an exported route is the same as the original route. Global export maps are therefore not useful when NAT is enabled.

Characteristics of Import and Global Import Maps

Import maps and global import maps can import both labeled and unlabeled routes. If you want to import only one or the other, you can use a **match mpls-label** command in the global import route map. Furthermore, if BGP imports labeled routes from the global BGP non-VPN RIB into a VRF RIB and then advertises them further upstream as labeled routes, the MPLS cross-connects are correctly created and MPLS forwarding works. The global VPN RIB never contains unlabeled routes, so the issue is moot for import maps.

When a route that was previously imported into the local VRF RIB is modified in the global BGP RIB (VPN or non-VPN) such that it no longer matches the import or global import map, that route is removed from the local VRF RIB.

Imported routes point to the same interface and next hop as the original route. Shared IP interfaces are not created.

Table 34 lists additional characteristics of import and global import maps.

Table 34: Characteristics of Import and Global Import Maps

Characteristic	Import	Global Import
Distributes routes from the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB.	Yes	–
Distributes routes from the global BGP non-VPN RIB local to the VR. This RIB is often referred to as the core non-VPN RIB or core RIB.	–	Yes
Imports all types of routes (received routes, redistributed routes, network routes, aggregate routes, and auto-summary routes).	Yes	Yes
Imports both best and non-best routes. The best route selection (including the decision to use or not use ECMP) is made in the VRF after the routes are imported.	Yes	Yes

Characteristics of Export and Global Export Maps

Export maps and global export maps can export both labeled and unlabeled routes. If you want to export only one or the other, you can use a **match mpls-label** command in the export or global export route map.

Table 35 lists additional characteristics of export and global export maps.

Table 35: Characteristics of Export and Global Export Maps

Characteristic	Export	Global Export
Distributes routes to the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB.	Yes	–
Distributes routes to the global BGP non-VPN RIB local to the VR. This RIB is often referred to as the core non-VPN RIB or core RIB.	–	Yes
Exports all types of routes (received routes, redistributed routes, network routes, aggregate routes, and auto-summary routes).	Yes	–
Exports only locally originated routes (all routes other than those that have been received).	–	Yes
Exports both best and non-best routes. The best route selection is made again in the core after the export.	Yes	Yes

Subsequent Distribution of Routes

Routes that are imported from the global BGP non-VPN RIB (with a global import map) into a VRF RIB are never exported again. Because these routes are not exported to the global VPN RIB, they are not advertised to other PE routers. These imported routes are never exported to the VRF RIBs of overlapping VPNs.

Routes that are exported from a VRF RIB to the global BGP non-VPN RIB with the global export map are never imported back in to any VRF.

Routes that are imported from the global BGP VPN RIB (with an import map) into a VRF RIB are never exported again.

Routes that are exported from a VRF RIB to the global VPN RIB can be imported into the RIB of other VRFs. This behavior might be seen with overlapping VPNs.

Creating a Map

For information about creating a route map to be used as an import or export map, see *Chapter 1, Configuring BGP Routing*. The following example shows how to apply the route map *routemap5* to the VRF *vpnA* configured on the virtual router *boston*.

```
host1(config)#virtual router boston
host1:boston(config)#ip vrf vpnA
host1:boston(config-vrf)#import map routemap5
```

Export Maps

You can use an export map to change the attributes of a route when it is exported from a VRF to the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB. Export maps can optionally filter routes.

When the VRF route matches the export map, the route is exported and the attributes are changed as specified in the export map.

When the VRF route does not match the export map, the **filter** keyword determines what happens. If the **filter** keyword has been issued, then the route is not exported. If the **filter** keyword has not been issued, then the route is exported but the attributes of the route are not modified (because the export map was not matched).

If you do not configure an export map, then all routes are exported from the VRF to the global BGP VPN RIB. However, routes that are imported into the VRF cannot be exported again.

export map

- Use to apply a route map to a VRF to modify or filter routes exported from the VRF to the global BGP VPN RIB in the parent VR.
- You can specify that only IPv4 or only IPv6 routes are exported. By default, both types of routes are exported.
- Example

  ```
  host1:boston(config-vrf)#export map routemap5 filter
  ```

- Use the **no** version to remove the route map from the VRF.

Global Export Maps

You can use a global export map to change the attributes of a route when it is exported from a VRF to the global BGP non-VPN RIB local to the VR.

If the VRF route matches the export map, then the route is exported and the attributes are changed as specified in the export map. If the VRF route does not match the export map, then the route is not exported. If you do not configure a global export map, then no routes are exported from the VRF to the global BGP non-VPN RIB.

Routes that are imported into the VRF cannot be exported again. As a consequence, VPN routes can be injected only into the global IP routing table on the PE router that is directly connected to the CE router that originates the prefix.

See *Global Export of IPv6 VPN Routes into the Global BGP IPv6 RIB* on page 414 for information about global export maps and IPv6 VPNs.

global export map

- Use to apply a route map to a VRF to modify and filter routes exported by the VRF to the global BGP non-VPN RIB in the parent VR.
- You can specify that only IPv4 or only IPv6 routes are exported. By default, both types of routes are exported.
- Example

  host1:boston(config-vrf)#**global export map routemap14**

- Use the **no** version to disable the exporting of routes to the global BGP non-VPN RIB.

Import Maps

You can use an import map to change the attributes of a route when it is imported from the global BGP VPN RIB to a VRF. You can also use an import map to filter routes. If you associate an import map with a VRF, that VRF then accepts only received routes that pass the import map (and match the import route target list).

import map

- Use to apply an import route map to a VRF to modify and filter routes imported to the BGP RIB of the VRF from the global BGP VPN RIB in the parent VR.
- You can specify that only IPv4 or only IPv6 routes are imported. By default, both types of routes are imported.
- Example

  host1:boston(config-vrf)#**import map routemap72**

- Use the **no** version to remove the route map from the VRF.

Global Import Maps

Global import maps enable BGP routes to be imported from the global BGP non-VPN RIB into the BGP RIB of a VRF based on a configured route map. You can use import maps as an automated mechanism that enables a subset of the Internet to be reachable from a VPN. This feature is intended to provide simplified central access to a limited number of centralized services in the provider network. Use this feature to import only a relatively small number (tens) of routes from the global domain into the VPNs, such as a small number of routes to DNS servers, content servers, management stations, and so on.

If instead you import the full Internet routing table into one or more VPNs, too much memory will be consumed because this action stores multiple copies of the full Internet routing table. To prevent an accidental misconfiguration, you must specify the maximum number of routes to be imported into a VRF when you configure global import. If you must provide access to the full Internet from a VPN, use the **fallback global** command.

global import map

- Use to apply a route map to a VRF to modify and filter routes imported to the BGP RIB of the VRF from the global BGP non-VPN RIB in the parent VR.
- You can specify that only IPv4 or only IPv6 routes are imported. By default, both types of routes are imported.
- Use the **max-routes** keyword to specify the maximum number of routes that you want to be imported into the local RIB. BGP generates a log message when the specified number of routes has been imported; no additional routes are imported.

  WARNING 02/11/2005 10:28:35 bgpRoutes (default,10.13.5.21): Maximum number of routes (5000) imported from global RIB into RIB of VRF foo.

- Changes to the maximum number of routes take effect immediately.
- Example

  host1:boston(config-vrf)#**global import map routemap22 max-routes 512**

- Use the **no** version to disable the importing of routes from the global BGP non-VPN RIB to the BGP RIB of the VRF.

Global Export of IPv6 VPN Routes into the Global BGP IPv6 RIB

VPNv6 routes can be exported from the BGP RIB of an IPv6 VRF to the global IPv6 BGP RIB based on policy by means of a route map and the **global export map** command.

For example, if you have a mixed IPv4 and IPv6 VPN configuration, but want only the IPv6 VPN routes to be exported from the IPv6 VRF into the global IPv6 RIB, you can use a route map that matches on IPv6 access-lists (IPv6 prefix-lists). You can have the route map disallow IPv4 VPN routes by matching on IPv4 access lists that filter out IPv4 prefixes.

The following commands illustrate this behavior.

- Configure an IPv6 access list to export IPv6 VPN prefixes to the global IPv6 RIB.

  host1(config)#**ipv6 access-list everything-v6 permit any any**

- Configure an IPv4 access list to disallow the export of IPv4 prefixes to the global IPv4 RIB.

  host1(config)#**access-list nothing-v4 deny ip any any**

- Configure a route map to permit global export of IPv6 VPN routes to the global IPv6 RIB.

  host1(config)#**route-map export-only-v6**
  host1(config-route-map)#**match ip address nothing-v4**
  host1(config-route-map)#**match ipv6 address everything-v6**
  host1(config-route-map)#**set local-preference 444**
  host1(config-route-map)#**exit**
  host1(config)#**ip vrf foo**
  host1(config-route-vrf)#**global export map export-only-v6**

If you need to export both IPv4 and IPv6 VPN routes from the IPv4/IPv6 VRF to the global IPv4 BGP RIB and to the global IPv6 BGP RIB, then configure a route map that permits both IPv4 and IPv6 prefixes.

Assigning an Interface to a VRF

You must assign an interface or subinterface to a VRF so that when the router receives a packet at this interface, it routes the packet using the VRF's forwarding table rather the global forwarding table. You can assign the interface from outside the context of the VRF or inside the context of the VRF.

To assign an interface to a VRF from outside the VRF context:

1. Select the interface.

2. Specify the VRF to associate with the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**
   host1:vr1(config-if)#**ip vrf forwarding vrfA**

3. Assign an IP address to the interface because forwarding the interface from the VR to the VRF removes the existing IP configuration from the interface.

   host1:vr1:vrfA(config-if)#**ip address 10.16.2.77 255.255.255.0**

To assign an interface to a VRF from inside the VRF context:

1. Select the interface.

2. Enter the VRF context.

   host1:vr1(config)#**virtual-router :vrfA**

3. Associate the interface.

   host1:vr1:vrfA(config)#**interface gigabitEthernet 1/0**

   In this case, you do not have to reassign an IP address to the interface because you did not use the **ip vrf forwarding** command.

ip vrf forwarding

- Use to assign a VRF to an interface or subinterface by forwarding the interface from the VR to the VRF. This command also enables you to specify secondary routing table lookup for a VRF, in the event that an initial routing table lookup does not yield results.
- Forwarding the interface removes the IP configuration from the interface. You must reassign an IP address to the interface after you issue this command.
- The **ip vrf forwarding** command changes the prompt to indicate that the CLI is now in Interface Configuration mode within the child VRF. This condition persists only for as long as you are configuring attributes on the given interface within the VRF. Entering a top-level command, such as **interface**, within this VRF context takes the CLI out of the VRF context back to the parent VR context.
- When you issue the **ip vrf forwarding** command from within the Interface Configuration or Subinterface Configuration mode of the parent VR, the IP address and other attributes of the interface are deleted from the interface. You must then reconfigure the IP attributes in the context of the VRF after issuing the command.
- Example

  host1:foo(config-if)#**ip vrf forwarding vrfA**
  host1:foo:vrfA(config-if)#**ip address 10.12.4.5 255.255.255.0**

  or

  host1:foo(config-if)#**ip vrf forwarding vrfA fallback global**

- Use the **no** version to remove the interface assignment or discontinue secondary routing table lookup.

Defining Secondary Routing Table Lookup

You can enable secondary routing table lookup on the virtual router routing table of the parent (global) virtual router. The secondary lookup takes place when the initial route lookup on a VRF is unsuccessful. You can define secondary routing table lookup outside the context of the VRF or inside the context of the VRF.

To configure secondary routing table lookup from outside the VRF context:

1. Select the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**

2. Specify a VRF and that you want it to perform secondary routing table lookup.

   host1:vr1(config-if)#**ip vrf forwarding vrfA fallback global**
   host1:vr1:vrfA(config-if)#**ip address 10.12.4.5 255.255.255.0**

To specify from inside the VRF context that an interface use the fallback global routing table lookup:

1. Select the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**

2. Enter the VRF context.

   host1:vr1(config-if)#**virtual-router :vrfA**

3. Specify that the VRF perform a secondary routing table lookup.

   host1:vr1:vrfA(config-if)#**ip fallback global**

ip fallback global

- Use to specify secondary routing table lookup for an interface in a VRF if an initial routing table lookup is unsuccessful.
- Example

  host1:vr1:vrfA(config-if)#**ip fallback global**

- Use the **no** version to discontinue secondary routing table lookup.

ip vrf forwarding

- Use to assign a VRF to an interface or subinterface by forwarding the interface from the VR to the VRF. This command also enables you to specify secondary routing table lookup for a VRF if an initial routing table lookup is unsuccessful.
- Forwarding the interface removes the IP configuration from the interface. You must reassign an IP address to the interface after you issue this command.
- The **ip vrf forwarding** command changes the prompt to indicate that the CLI is now in Interface Configuration mode within the child VRF. This condition persists only for as long as you are configuring attributes on the given interface within the VRF. Entering a top-level command, such as **interface**, within this VRF context takes the CLI out of the VRF context back to the parent VR context.
- When you issue the **ip vrf forwarding** command from within the Interface Configuration or Subinterface Configuration mode of the parent VR, the IP address and other attributes of the interface are deleted from the interface. You must then reconfigure the IP attributes in the context of the VRF after issuing the command.
- Example

  host1:vr1(config-if)#**ip vrf forwarding vrfA**
  host1:vr1:vrfA(config-if)#**ip address 10.12.4.5 255.255.255.0**

  or

  host1:vr1(config-if)#**ip vrf forwarding vrfA fallback global**
  host1:vr1:vrfA(config-if)#**ip address 10.12.4.5 255.255.255.0**

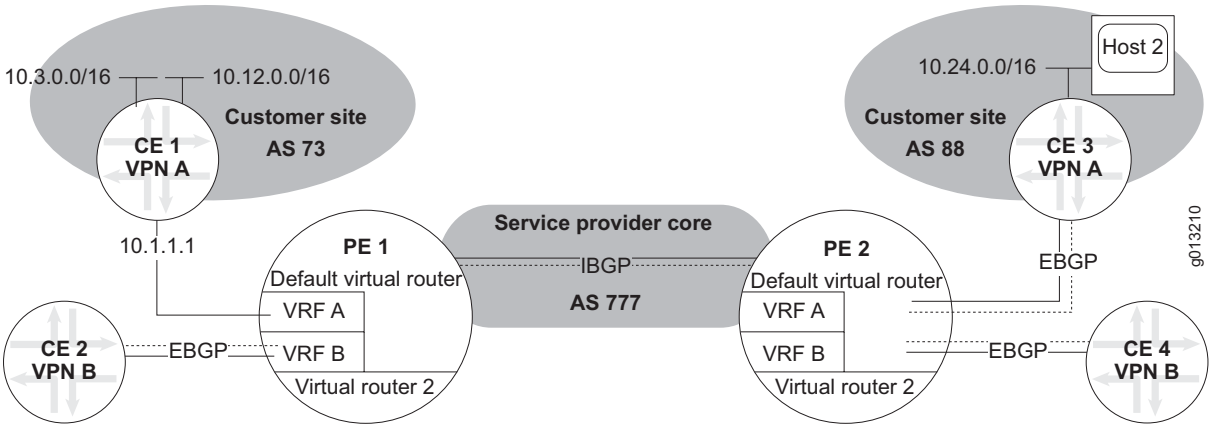
- Use the **no** version to remove the interface assignment or discontinue secondary routing table lookup.

Adding Static Routes to a VRF

Consider the network structure shown in Figure 91. If no routing protocol—BGP or any other IGP—is running between the PE router and the CE router, you must use the **ip route vrf** command to add a static route in the customer's VRF for each prefix in that customer's site.

Each of these static routes must point to the link connecting the PE router to the CE router. Typically, you redistribute these static routes in the VRF's address family in BGP or use **network** commands to make those prefixes reachable from other CE routers in the same VPN.

Figure 91: Configuring Static Routes



In Figure 91, PE 2 has external BGP connections to CE 3 and CE 4. PE 1 has an EBGP connection to CE 2. However, no BGP (or IGP) connection exists between PE 1 and CE 1. The following example shows how to configure static routes on VRF A for both prefixes in CE 1.

```
host1(config)#virtual-router pe1
host1:pe1(config)#ip vrf vpnA
host1:pe1(config-vrf)#ip route vrf vrfA 10.3.0.0 255.255.0.0 10.1.1.1
host1:pe1(config-vrf)#ip route vrf vrfA 10.12.0.0 255.255.0.0 10.1.1.1
```

ip route vrf

- Use to add a static route to a VRF.
- Example

  ```
  host1:pe1(config-router-af)#ip route vrf vrfA 10.0.0.0 255.0.0.0 192.168.1.1
  ```

- Use the **no** version to remove a static route from a VRF.

Configuring IGPs on the VRF

If you do not configure static routes on the VRF for each prefix in the associated customer site, then you must configure an IGP on the VRF so that the VRF can learn routes from customer sites.

Configuring the IGP in the VRF Context

After creating a VRF, you can access it as if it were a virtual router for the purpose of configuring the IGP.

If you are in the context of the virtual router that has the VRF, you access the VRF as follows:

```
host1(config)#virtual-router :vrfa
host1:default:vrfa(config)#
```

If you are *not* in the context of the virtual router that has the VRF, you access the VRF as follows:

```
host1(config)#virtual-router boston:vrfa
host1:boston:vrfa(config)#
```

The following commands illustrate one way to configure OSPF; you can configure RIP and IS-IS similarly:

```
host1(config)#ip vrf vrfa
host1(config-vrf)#rd 100:5
host1(config-vrf)#route-target both 100:5
host1(config-vrf)#exit
host1(config)#virtual-router :vrfa
host1:default:vrfa(config)#router ospf 100
host1:default:vrfa(config-router)#redistribute bgp
```

At this point you proceed with the IGP configuration for the VRF.

Configuring the IGP Outside the VRF Context

The RIP and OSPF protocols also enable you to specify a VRF and configure the protocol without actually entering the VRF context.

For example, for OSPF you might issue the following command and then complete OSPF configuration tasks for VRF A:

```
host1(config)#router ospf 100 vrf vrfa
```

For RIP, you create the RIP process, specify the address family for the VRF, and specify redistribution of BGP routes for VRF A:

```
host1(config)#router rip 100
host1(config-router)#address-family ipv4 vrf vrfa
host1(config-router-af)#redistribute bgp
```

At this point you proceed with RIP configuration for the VRF.

See the appropriate chapter for information about configuring the desired IGP:

- *Chapter 1, Configuring BGP Routing*
- *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 4, Configuring RIP*
- *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 5, Configuring OSPF*
- *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 6, Configuring IS-IS*

virtual-router

- Use to access a VRF to configure it with an IGP to learn routes from a CE router.
- To access the VRF from its VR context (in this example, the default VR):

  host1(config)#**virtual-router :vrfsouthie**
  host1:default:southie(config)#

- To access the VRF from the context of a different VR:

  host1(config)#**virtual-router boston:southie**
  host1:boston:southie(config)#

- You must use the **no ip vrf** command to remove a VRF. Issuing a **no** version of this command (**no virtual-router :***vrfName* or **no virtual-router** *vrName*:*vrfName*) that specifies an existing VRF only displays the error message:

  "Cannot delete a VRF with this command"

Disabling Automatic Route-Target Filtering

When BGP receives a VPN-IPv4 or VPN-IPv6 route from another PE router, BGP stores that route in its local routing table only if at least one VRF imports a route target of that route. If no VRF imports any of the route targets of the route, BGP discards the route; this feature is called automatic route-target filtering. The intention is that BGP keeps track of routes only for directly connected VPNs, and discards all other VPN-IPv4 or VPN-IPv6 routes to conserve memory.

If a new VPN is connected to the router (that is, if the import route-target list of a VRF changes), BGP automatically sends a route-refresh message to obtain the routes that it previously discarded.

You can use the **no bgp default route-target filter** command to disable automatic route-target filtering globally for all VRFs. However, automatic route-target filtering is always disabled on route reflectors that have at least one route-reflector client. You cannot enable automatic route-target filtering for such route reflectors.

bgp default route-target filter

- Use to control automatic route-target filtering.
- Route-target filtering is enabled by default.
- Takes effect immediately. When route target filtering is turned on, this command immediately removes routes to be filtered.

  If route-target filtering is turned off, BGP automatically sends out a route-refresh message over every VPNv4 or VPNv6 unicast session (for which the route-refresh capability was negotiated) to get previously filtered routes. If the route-refresh capability was not negotiated over the session, BGP bounces the session.
- Example

  host1:vrf1(config-router)#**no bgp default route-target filter**
- Use the **no** version to disable automatic route-target filtering.

Creating Labels per FEC

By default, the router minimizes the number of stacked labels to be managed by generating a single label for all BGP routes advertised by a given VRF; this is a per-VRF label. Upon receiving traffic for a per-VRF label, the router performs a label pop and a route lookup to forward the traffic to the next hop.

You can use the **ip mpls forwarding-mode label-switched** command to configure the router to generate a label for each different FEC that a BGP route points to in the VPN; this is a per-FEC label. Issuing this command enables you to avoid a route lookup for traffic destined for CE routers, because in this mode traffic is label switched to the corresponding next hop over that interface; a route lookup is not performed.

The route for which a label is allocated can be an ECMP route; in that case, the label-switched traffic uses ECMP.

For the following types of routes, the router always generates a per-VRF label and forwards traffic after a route lookup (rather than label switching the traffic without a route lookup) regardless of the status of this command:

- Local connected interfaces redistributed into BGP, regardless of the interface type.
- BGP redistributed routes that point to loopback interfaces.

The following commands configure a router where BGP is running in VRF pe11 and static and connected routes are redistributed into the VRF:

```
host1(config)#ip vrf pe11
host1(config-vrf)#ip mpls forwarding-mode label-switched
host1(config-vrf)#ip route vrf pe11 10.3.4.5 255.255.255.255 fastEthernet 0/1
host1(config-vrf)#ip route vrf pe11 10.1.1.1 255.255.255.255 loopback 1
host1(config-vrf)#exit
host1(config)#router bgp 100
host1(config-router)#address-family ipv4 unicast vrf pe11
host1(config-router-af)#exit
host1(config-router)#no auto-summary
```

```
host1(config-router)#no synchronization
host1(config-router)#redistribute static
host1(config-router)#redistribute connected
```

For each connected route that is redistributed into the VRF and advertised across the BGP/MPLS VPN, the router assigns a per-VRF label rather than a per-FEC label.

The static route 10.1.1.1/32 points to loopback interface 1. BGP therefore advertises this static route with a per-VRF label.

ip mpls forwarding-mode label-switched

- Use to generate a label for each different FEC pointed to by a BGP route.
- For some types of routes, issuing this command has no effect on the labels created; they are always per-VRF labels.
- Example

  ```
  host1:vr1(config-vrf)#ip mpls forwarding-mode label-switched
  ```

- Use the **no** version to restore the default, generating a single label for all BGP routes sent from a given VRF.

Configuring PE-to-PE LSPs

See *Chapter 2, Configuring MPLS*, for information about configuring LSPs.

Enabling BGP Routing

You must enable the BGP routing process on the router serving as the PE router.

router bgp

- Use to enable the BGP routing protocol and to specify the local AS—the AS to which this BGP speaker belongs.
- All subsequent BGP configuration commands are placed within the context of this router and AS; you can have only a single BGP instance per virtual router.
- Specify only one BGP AS per virtual router.
- Example

  ```
  host1:vr1(config)#router bgp 100
  ```

- This command takes effect immediately.
- Use the **no** version to remove the BGP process.

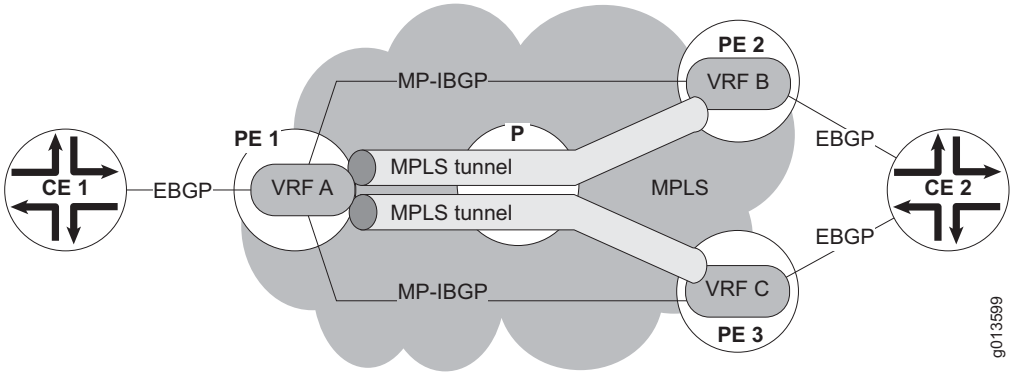
Enabling BGP ECMP for BGP/MPLS VPNs

Enabling ECMP support for BGP/MPLS VPNs allows multiple VPN routes to be included in the list of available equal-cost paths. You can use the **maximum-paths** command with the **ibgp** or **eibgp** keywords to enable ECMP support for BGP/MPLS VPNs.

The **eibgp** keyword specifies that the E-series router consider *both* external BGP (EBGP) and internal BGP (IBGP) paths when determining the number of equal-cost paths to the same destination that BGP can submit to the IP routing table. The **ibgp** keyword specifies that the E-series router consider multiple internal IBGP paths, but not EBGP paths, when determining the number of equal-cost paths.

Example 1 You can create an ECMP environment in which multiple IBGP paths are selected as multipaths and used for load balancing. In the example shown in Figure 92, the E-series router gives equal consideration to IBGP VPN routes learned from multiple remote PE devices when determining load balancing.

Figure 92: BGP/MPLS VPN IBGP Example



The sample BGP/MPLS network connects PE 1, PE 2, and PE 3, which are configured for VPNv4 unicast IBGP peering. CE 1 and CE 2 are configured for EBGP peering with the PE devices. CE 2 is multihomed, connected to both PE 2 and PE 3.

VRF A has two equal-cost paths through the MPLS network to get to CE 2: the IBGP path to PE 2, and the IBGP path to PE 3.

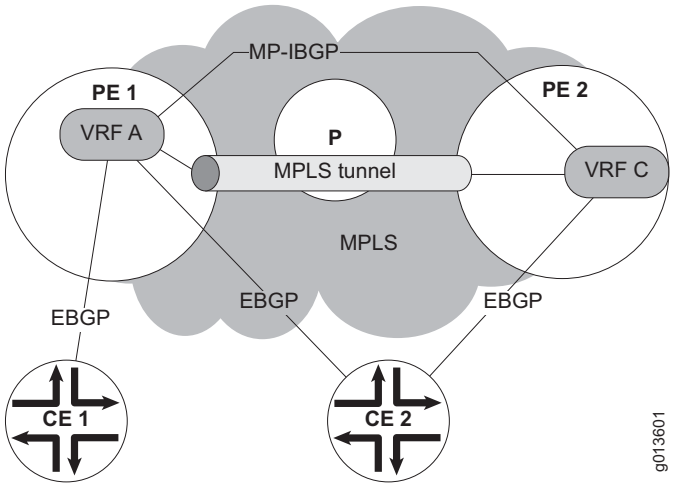
To support BGP/MPLS ECMP, PE 1 is configured with the **maximum-paths ibgp** command under IPv4 unicast VRF A address family. Doing this allows IBGP paths from both PE 2 and PE 3 to be selected as multipaths for use in load balancing.

Traffic from CE 1 to CE 2 that takes an IBGP route from PE 1 to either PE 2 or PE 3 is forwarded as MPLS-encapsulated packets. PE 2 and PE 3 receive the MPLS-encapsulated traffic from PE 1, remove the MPLS encapsulation, and then forward the traffic as IP packets by means of their EBGP route to CE 2.

Example 2 You can create a mixed ECMP environment in which both EBGP and IBGP paths are selected as multipaths and used for load balancing. Doing this enables the E-series router to take into account both EBGP VPN routes learned from a CE router device and IBGP VPN routes learned from a remote PE device when determining load balancing.

In Figure 93, a BGP/MPLS network connects PE 1 and PE 2, which are configured for VPNv4 unicast IBGP peering. CE 1 and CE 2 are configured for EBGP peering with the PE devices. CE 2 is multihomed, connected to both PE 1 and PE 2.

Figure 93: BGP/MPLS VPN EIBGP Example



VRF A has two paths to get to CE 2: the IBGP path through the MPLS network, and the EBGP path by means of regular IP.

To support BGP/MPLS ECMP, PE 1 is configured with the **maximum-paths eibgp** command in the IPv4 unicast VRF A address family. Doing this allows both the EBGP paths from CE 2 and the IBGP paths from PE 2 to be selected as multipaths in the VRF A routing information base (RIB) for use in load balancing.

Traffic taking the various routes from CE 1 to CE 2 is treated as follows:

- Traffic from CE 1 to CE 2 that takes the EBGP route from PE 1 is forwarded as IP packets.
- Traffic from CE 1 to CE 2 that takes the IBGP route from PE 1 is forwarded as MPLS-encapsulated packets. PE 2 receives the MPLS-encapsulated traffic from PE 1, removes the encapsulation, and then forwards the traffic as IP packets by means of the EBGP route to CE 2.

maximum-paths

- Use to enable ECMP support for BGP/MPLS VPNs.
- Specify a value in the range 1–16; the default value is 1. The value indicates the maximum number of equal-cost multipaths for VPN routes.
- This command takes effect immediately; it does not bounce the session.
- For BGP/MPLS support, you must specify the maximum number of equal-cost multipaths in the context of a VRF IPv4 unicast or IPv6 unicast address family.
- This command is not supported for the VPNv4 or VPNv6 address families.
- The **maximum-paths eibgp** command cannot be used if the router is currently configured with the **maximum-paths** or **maximum-paths ibgp** command.

- Example

  host1(config)#**router bgp 100**
  host1(config-router)#**address-family ipv4 vrf vrfA**
  host1(config-router-af)#**maximum-paths eibgp 6**

- Use the **show ip bgp vpnv4 vrf** *vrfName* **summary** or **show bgp ipv6 vpnv6 vrf** *vrfName* **summary** command to verify your ECMP configuration. The output includes a line indicating the equal-cost paths:

  ```
  Maximum number of both EBGP and IBGP equal-cost paths is 16
  ```

- Use the **no** version to restore the default value, 1.

Enabling VPN Address Exchange

To limit the exchange of routes to those from within the VPN-IPv4 address family, and to set other desired BGP parameters:

1. Specify that the router exchanges addresses within a VPN by choosing the VPN-IPv4 address family.
2. Specify individual neighbors or peer groups to exchange routes with from only within the current (VPN-IPv4) address family.
3. Configure BGP parameters for VPN services.

   See *Chapter 1, Configuring BGP Routing*, for information about configuring BGP sessions. The section *Understanding BGP Command Scope* on page 17 has tables that list BGP commands according to their scope. From Address Family Configuration mode, you can issue the commands in Table 8 on page 18 and Table 10 on page 19.
4. Exit Address Family Configuration mode.

address-family

- Use to configure the router to exchange IPv4 addresses in VPN mode.
- The default setting is to exchange IPv4 addresses in unicast mode from the default router.
- This command takes effect immediately.
- Example

  host1:vr1(config-router)#**address-family vpnv4**

- Use the **no** version to disable the exchange of a type of prefix.

exit-address-family

- Use to exit Address Family Configuration mode and access Router Configuration mode.
- Example

  host1:vr1(config-router-af)#**exit-address-family**

- There is no **no** version.

neighbor activate

- Use to specify neighbors to exchange routes with from within the current address family.
- Takes effect immediately.
- If dynamic capability negotiation was not negotiated with the peer, the session is automatically bounced so that the exchanged address families can be renegotiated in the open messages when the session comes back up.
- If dynamic capability negotiation was negotiated with the peer, BGP sends a capability message to the peer to advertise or withdraw the multiprotocol capability for the address family in which this command is issued.
- If a neighbor is activated, BGP also sends the full contents of the BGP routing table of the newly activated address family.
- Example

  host1:vr1(config-router-af)#**neighbor 192.168.1.158 activate**

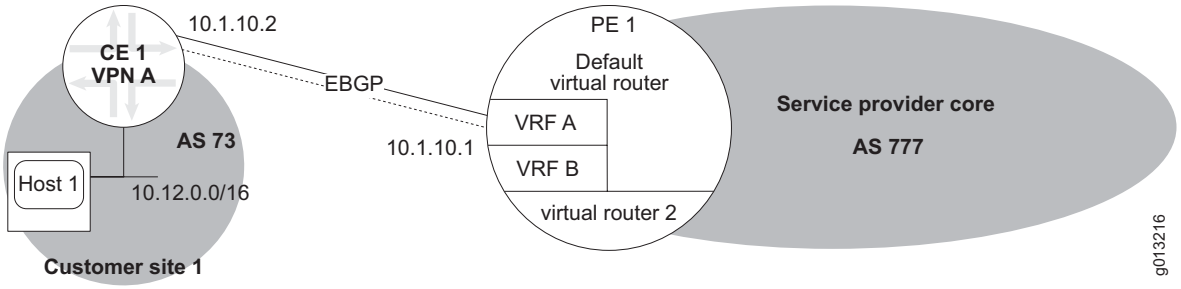
- Use the **no** version to indicate that routes of the current address family should not be exchanged with the peer. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.

Configuring PE-to-CE BGP Sessions

If you have established a BGP session between a PE and a particular CE router, you can configure BGP sessions with all the other customer sites within the VPN so that they can learn the routes to the particular CE router.

Configuring the PE-to-CE external BGP session is a bit different from the usual external BGP session. You must configure the session in the context of the IPV4 unicast address family of the VRF. Consider the topology shown in Figure 94.

Figure 94: PE-to-CE Session



You configure the characteristics of VRF A, the global BGP attributes, the address family for the session, and BGP attributes relevant to the VRF or address family.

host1(config)#**ip vrf vrfa**
host1(config-vrf)#**rd 777:5**
host1(config-vrf)#**route-target both 777:5**
host1(config-vrf)#**exit**
host1(config)#**interface gigabitEthernet 1/0**
host1(config-if)#**ip vrf forwarding vrfA**

```
host1(config-if)#ip address 10.1.10.1 255.255.255.0
host1(config-if)#exit
host1(config)#router bgp 777

    (Not shown: Configuration of other global BGP attributes)

host1(config-router)#address-family ipv4 unicast vrf vrfA
host1(config-router-af)#neighbor 10.1.10.2 remote-as 73

    (Not shown: Configuration of BGP attributes relevant to the VRF or the address
family)
```

See *Chapter 1, Configuring BGP Routing*, for more information about configuring BGP.

Advertising Static Routes to Customers

If you established static routes on a PE router for each prefix in a particular customer site, you can configure BGP on the PE router to advertise these static routes to customer sites within the VPN with **network** commands.

```
host1:vr1(config-router)#network 10.3.0.0
host1:vr1(config-router)#network 10.12.0.0
```

In this example, both networks end on a classful boundary, eliminating the need to configure a network mask.

Alternatively, you can use the **redistribute** command to advertise the static routes as follows:

```
host1:vr1(config-router)#redistribute static
```

See *Chapter 1, Configuring BGP Routing*, for more information about advertising static routes.

Advertising IGP Routes to Customers

If the PE router learns routes from a CE router by means of an IGP, you can configure BGP to advertise these IGP routes to all customer sites within the VPN with **redistribute** commands. For example, if the PE router learns the routes by means of OSPF, you can issue the following command to inject these routes into BGP for advertisement:

```
host1:vr1(config-router)#redistribute ospf
```

See *Chapter 1, Configuring BGP Routing*, for more information about advertising IGP routes.

Disabling the Default Address Family

PE routers can exchange routes in the IPv4 address family, VPNv4 address family, or both. Issuing the **neighbor remote-as** command automatically activates the IPv4 unicast address family, meaning that the PE router exchanges routes in the IPv4 unicast address family with that peer.

Example 1 The following commands illustrate how to configure the exchange of routes in both the IPv4 unicast and the VPNv4 unicast address families for a BGP peer:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

The **neighbor remote-as** command activated the IPv4 unicast address family for the peer. The **address-family** command entered the context of the VPNv4 unicast family and the **neighbor activate** command activated the address family for the peer.

Example 2 The following commands illustrate one way to disable the exchange of routes in the IPv4 unicast address family and enable the exchange of routes in the VPNv4 unicast address family:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family ipv4 unicast
host1:vr1(config-router-af)#no neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

In this case, the **no neighbor activate** command specifically disables the IPv4 unicast address family for that peer alone; no other peers are affected. The VPNv4 unicast address family is activated for the peer as in Example 1.

Example 3 The following commands illustrate another way to disable the exchange of routes in the IPv4 unicast address family and enable the exchange of routes in the VPNv4 unicast address family:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#no bgp default ipv4-unicast
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

In this case, the **no bgp default ipv4-unicast** command prevents the automatic enabling of the IPv4 unicast address family for all peers subsequently configured with the **neighbor remote-as** command. Previously configured peers are not affected. The VPNv4 unicast address family is activated for the peer as in Examples 1 and 2.

Using a Single AS Number for All CE Sites

If you want to use the same AS number for all of your CE sites, you can substitute a PE router's autonomous system number for that of a neighbor by specifying the neighbor's IP address in the **neighbor as-override** command. If you fail to do this, the CE router recognizes its AS in the AS path of received routes and determines it has discovered a routing loop; the routes are rejected.

Example In the following example, the router's AS number of 777 overrides the neighboring router's AS number of 100.

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 172.16.20.10 remote-as 100
host1:vr1(config-router)#neighbor 172.16.20.10 update-source loopback0
host1:vr1(config-router)#address-family ipv4 vrf vpn1
host1:vr1(config-router-af)#neighbor 172.25.14.12 remote-as 100
host1:vr1(config-router-af)#neighbor 172.25.14.12 as-override
```

neighbor as-override

- Use to enable the use of the same AS number for all CE sites by substituting the current router's AS number in routing tables for that of the neighboring router.
- If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.
- To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.
- Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Example 1

  ```
  host1:vr1(config-router)#neighbor 192.168.255.255 as-override
  ```

- Example 2

  ```
  host1(config-router)#neighbor 192.168.1.158 as-override
  ```

- Use the **no** version to halt the substitution of the AS numbers. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.

Preventing Routing Loops

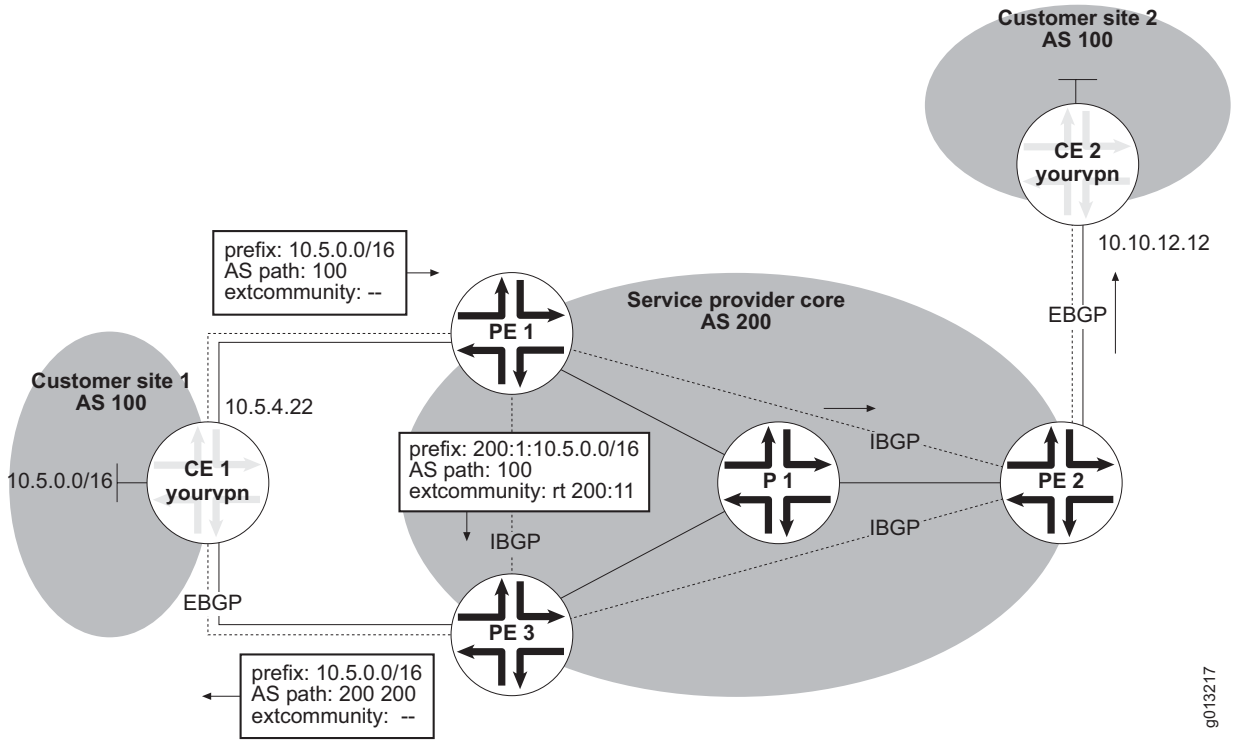
Routing loops can occur when routes learned from a peer are later advertised back to that peer. Normally such routing loops are prevented by the AS path attribute. However, the AS path cannot prevent routing loops in a network configuration with the following characteristics:

- BGP is running between CE and PE routers.
- You use a single AS number for all customer sites, and have issued the **neighbor as-override** command for the PE routers.
- A CE router is dual-homed to two or more PE routers.

The site-of-origin extended community attribute enables BGP to filter out such routes to prevent routing loops in this network. You can use the **set extcommunity** command to specify a site of origin and then use the **match extcommunity** command and an outbound route map to filter routes; for more information, see *JUNOSe IP Services Configuration Guide, Chapter 1, Configuring Routing Policy*.

Alternatively, you can use the **neighbor site-of-origin** command alone to achieve the same effect in such a network configuration. Consider the network shown in Figure 95, which enables PE 3 to advertise back to CE 1 routes that it learned from PE 1 that originated with CE 1. In a typical network configuration, CE 1 rejects these routes because it determines from the AS path that a routing loop exists. In this particular network, the **neighbor as-override** command prevents this method of detection.

Figure 95: Network with Potential Routing Loops



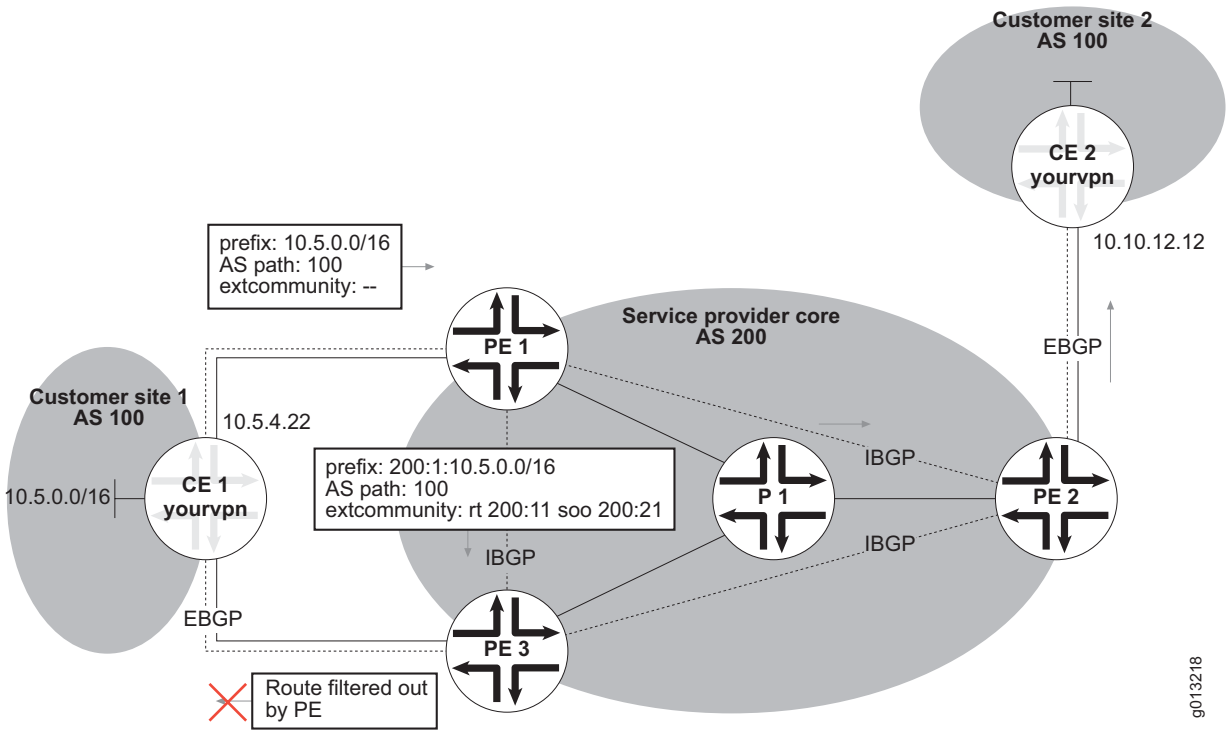
The following commands are relevant to the illustrated network:

```
host1:pe1(config)#ip vrf yourvpn
host1:pe1(config-vrf)#rd 200:1
host1:pe1(config-vrf)#route-target both 200:11
...
host1:pe1(config)#router bgp 200
host1:pe1(config-router)#address-family ipv4 unicast vrf yourvpn
host1:pe1(config-router)#neighbor 10.5.4.22 remote-as 100
host1:pe1(config-router)#neighbor 10.5.4.22 as-override
...
```

Now, suppose instead you assign a unique site of origin to each CE router in the network and configure the BGP session on each PE router with the site of origin. The result of the following (partial) configuration is shown in Figure 96.

```
host1:pe1(config)#ip vrf yourvpn
host1:pe1(config-vrf)#rd 200:1
host1:pe1(config-vrf)#route-target both 200:11
...
host1:pe1(config)#router bgp 200
host1:pe1(config-router)#address-family ipv4 unicast vrf yourvpn
host1:pe1(config-router)#neighbor 10.5.4.22 remote-as 100
host1:pe1(config-router)#neighbor 10.5.4.22 as-override
host1:pe1(config-router)#neighbor 10.5.4.22 site-of-origin 200:21
...
```

Figure 96: Preventing Potential Routing Loops in the Network



neighbor site-of-origin

- Use to set a site of origin that is included in the extended community list for routes received from the specified peer.
- If you use this command to configure a site of origin for routes from a peer, then routes advertised to that peer that contain this site of origin are filtered out and not advertised. This behavior is followed regardless of whether the **neighbor send-community extended** command has been issued for the peer.
- The configured site of origin does not override the site of origin if it is already present in the extended community list of a route.
- If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- The site of origin is applied to all routes that are received or advertised to all after you issue the command. The session is not bounced.
- To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.
- Example

  host1(config-router)#**neighbor 10.25.32.4 site-of-origin 200:21**

- Use the **no** version to remove the site of origin for routes received from the peer.

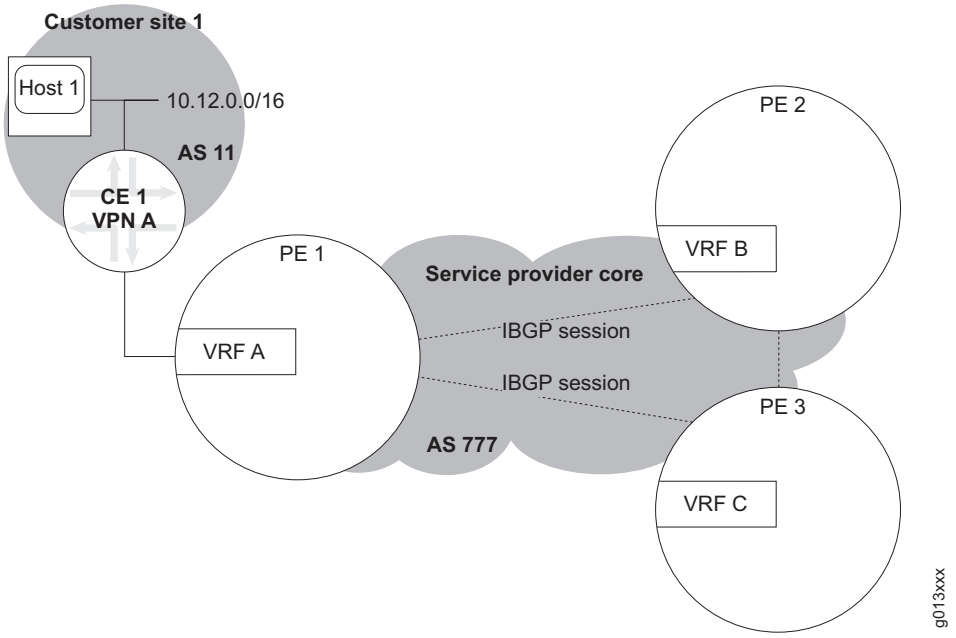
Advertising Prefixes with Duplicate AS Numbers

When a BGP speaker receives a route that has the speaker's AS number in its AS path, the speaker declares that route to be a loop and discards it. However, in some circumstances, as in the implementation of a hub-and-spoke VPN topology, this is not the desired behavior. You want the BGP speaker (hub) to accept such routes. You can use the **neighbor allowas-in** command to specify the number of times that a route's AS path can contain the BGP speaker's AS number.

The behavior is different within the VPNv4 address family than it is in other address families. For other address families, you must configure the feature on all the peers. In contrast, IBGP peers within the VPNv4 address family always accept routes containing their own AS number by default. Issuing this command in the VRF for such a peer has no effect on the behavior of IBGP peers in this address family. This behavior reduces the provisioning overhead for VPNv4 IBGP peers.

However, you must configure the feature on the peer router at the hub. Consider the hub-and-spoke topology shown in Figure 97. PE 1, PE 2, and PE 3 are peers in the VPNv4 address family. Routes received from CE 1 may contain the AS number (777) local to the PE routers. You must issue the **neighbor allowas-in** command for VRF A on PE 1.

Figure 97: Allowing Local AS in VPNv4 Address Family



neighbor allowas-in

- Use to enable the acceptance of all routes whose AS path contains the BGP speaker's AS number up to the specified number of times.
- If the AS path of a route contains the speaker's AS number more than the specified number of times, the route is determined to be a loop and is discarded.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.
- To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.
- Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Example

  host1(config-router)#**neighbor allowas-in**

- Use the **no** version to prevent the acceptance of these routes, resulting in the BGP speaker's discarding the routes.

Controlling Route Importation

You can control how many routes a PE router can add to a particular VRF's forwarding table by specifying a maximum limit and a warning threshold. When the router attempts to add a route, it compares the limit you configure against a route count it maintains for routes already in the VRF's forwarding table.

With a warning threshold configured, the following behavior takes place when the PE router attempts to add a route:

- When adding the route causes the route count to exceed the warning threshold for the first time, the router adds the route and generates a warning-threshold-exceeded log entry.
- As long as the route count stays above the warning threshold, adding more routes does not generate more warning-threshold-exceeded log entries.
- If the route count fluctuates below and above the warning threshold due to route deletions and additions, an interval of 5 minutes since the last warning-threshold-exceeded log entry must pass before another warning-threshold-exceeded log entry can be generated. This behavior prevents the system log from being flooded with log entries.

With a limit configured, the following behavior takes place when the PE router attempts to add a route:

- When adding the route causes the route count to exceed the limit for the first time, the router rejects the route and generates a limit-exceeded log entry.
- As long as the route count stays at the limit, further attempts to add routes fail, but do not generate any more limit-exceeded log entries.
- If the route count fluctuates below and up to the limit due to route deletions and additions, no further limit-exceeded log entries are generated until a 5-minute interval has passed since the last limit-exceeded log entry. This behavior prevents the system log from being flooded with log entries.

When you issue the command, the router immediately reevaluates the current number of routes against the new limit. If the current number of routes is greater than the maximum configured limit, the router might remove dynamically learned routes in order to enforce the new limit.

maximum routes

- Use to prevent a PE router from importing too many routes from attached CE routers into a particular VRF.
- When the router attempts to add a route that exceeds the *warningThreshold*, the router generates a warning-threshold log entry and adds the route. An interval of 5 minutes must pass before another warning-threshold-exceeded message can be generated.
- When the router attempts to add a route that exceeds the *limit*, the router generates a limit-exceeded warning and rejects the route. An interval of 5 minutes must pass before another limit-exceeded message can be generated.
- Messages are logged to ipRouteTable at severity warning.

- The interval timers for the limit and the warning threshold are independent.
- You can use the **warning-only** keyword to specify that the router add the route and generate a warning-threshold-exceeded log entry (instead of a limit-exceeded log entry) when the limit is exceeded.
- Issuing the command causes the router to evaluate the current route count and determine whether to generate new messages; any existing warning threshold or limit timers are reset to zero.
- Example

  host1(config-vrf)#**maximum routes 80 65**

- Use the **no** version to remove the limit and warning threshold.

Deleting Routes for a VRF

You can delete one or all IP routes assigned to a VRF or all VRFs.

clear ip routes

- Use to clear routes from the routing table of one or all VRFs.
- If you do not specify a VRF, routes are removed from all VRFs.
- You can specify either that a single route or all dynamic routes are to be removed.
- This command takes effect immediately.
- Example

  host1:vr1#**clear ip routes vrf vr3 ***

- There is no **no** version.

Enabling VRF–to–VR Peering

In some circumstances you might want a CE router, which connects to the PE router by means of a VRF, to be able to establish an EBGP peering session directly with the parent VR in which the VRF has been configured. The global instance of BGP for the PE router runs in the parent VR to exchange VPN routes with its peers by means of internal or external MP-BGP. BGP could also be learning IPv4 unicast Internet routes from one or more of its core-facing, internal or external BGP peers.

In the context of the VRF, you can use the **ip route parent-router** command to add a static host route to a stable interface (typically a loopback interface) in the parent VR by way of a hidden VRF-internal interface.

```
host1(config)#virtual-router PE1
host1:PE1(config)#interface loopback 1
host1:PE1(config-if)#ip address 10.20.20.2 255.255.255.255
host1:PE1(config-if)#exit
host1:PE1(config)#virtual-router :PE11
host1:PE1:PE11(config)#ip route parent-router loopback 1
```

In this example, assume that the global instance of BGP for the PE router runs in the parent VR, PE 1, to exchange VPN routes with its peers by means of internal or external MP-BGP. BGP can also be learning IPv4 unicast Internet routes from one or more of its core-facing, internal or external BGP peers.

By virtue of the static route configured in VRF PE 11, a CE router that connects to that VRF can establish an EBGP session directly to loopback 1 (10.20.20.2) in the parent VR, PE 1. The same PE router can therefore provide both VPN and Internet access to any attached CE routers.

You can display the static route to the parent VR with the **show ip route** and **show ip static** commands, as in the following examples:

```
host1:PE1:PE11#show ip route
Prefix/Length      Type    Next Hop           Dist/Met     Intf
------------------ ------- -----------------  ------------ ---------------
10.20.20.2/32     Static  0.0.0.0[V:PE1]   1/0          vrf-internal3

host1:PE1:PE11#show ip static
Prefix/Length      Next Hop   Met   Dist    Tag   Intf
10.20.20.2/32      0.0.0.0    0     1       0     vrf-internal3
```

ip route parent-router

- Use to establish a static route in a VRF to a remote interface in the parent VR.
- The specified interface must be preexisting and have an alias assigned with the **description** command.
- The route points to a next-hop interface that is internal to the VRF and created automatically when the VR comes up. This interface is hidden and cannot be displayed with the **show ip interface** command. You must use the **show ip route** or **show ip static** commands to display the interface.
- If the interface in the parent VR goes down or is deleted, the static route added in the VRF will continue to exist.
- Example

  host1(config-vrf)#**ip route parent-router vr1stat**

- Use the **no** version to remove the static route.

Achieving Fast Reconvergence in VPN Networks

By default, BGP does not confirm the reachability of the BGP indirect next hop of VPNv4 routes received over an MP-IBGP session until those routes have been imported into a VRF.

To BGP, the next hops of VPNv4 routes that are still in the global VPNv4 table (viewable with the **show ip bgp vpnv4 all** command) are always reachable. As a result, VPNv4 route reflectors that have multiple paths to the same prefix select the best route to reflect without taking into account the reachability of the BGP indirect next hop. Instead, best-path selection is based on weight, local preference, AS-path length, and other attributes.

After the route has been imported into a VRF, the reachability of the BGP indirect next hop is based on the presence of an MPLS tunnel (LDP or RSVP-TE) to the next-hop address and not on the presence of an IP route to the next-hop address.

Disregarding the reachability of the BGP indirect next hop when the router selects the best route to reflect can cause very slow reconvergence (up to 90 seconds) after a topology change in BGP/MPLS VPN networks that match all of the following conditions:

- Have a full mesh of LDP MPLS tunnels
- Have multihomed CE routers
- Use the same RD for multiple VRFs
- Rely on VPNv4 route reflectors as arbiters for selecting the best VPNv4 route from a set of clients

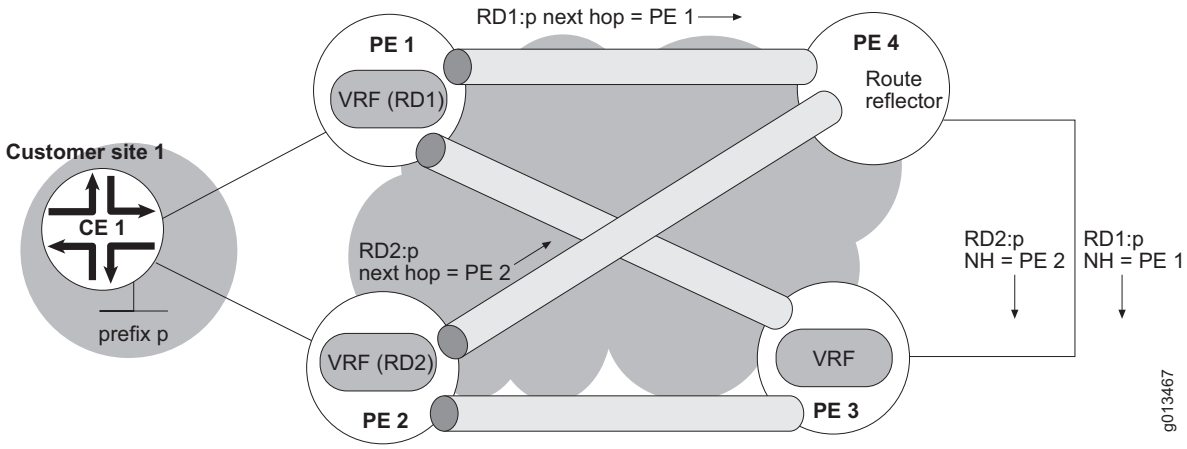
If a PE router fails in such a network, the route reflector must quickly reflect the VPNv4 route from the next-best PE router without having to wait for the BGP session to the failed PE router to time out. Depending on the network topology, you can achieve fast reconvergence by assigning unique RDs to each VRF or by enabling next-hop reachability checking.

Fast Reconvergence with Unique RDs

You can assign a unique RD for the VRFs in each PE router to avoid the slow reconvergence issue. The route reflectors in the network consider advertised routes with different RDs to be different prefixes and therefore reflect both routes.

In Figure 98, route reflector PE 4 reflects to PE 3 routes to the CE router through both PE 1 and PE 2. Suppose that the route through PE 1 is better than the route through PE 2. If you have assigned different RDs to the VRFs, then PE 4 reflects both routes to its client, PE 3.

Figure 98: Topology for Fast Reconvergence by Means of Unique VRF RDs, Before Tunnels Go Down



If PE 1 goes down, the MPLS tunnels to it (PE 4–PE 1 and PE 3–PE 1) are dropped immediately. However, because the route reflector does not take into account the reachability of the next hop, it still reflects both the PE 1 route and the PE 2 route.

When PE 3 imports these routes into its VRF, it resolves the routes and discovers that the tunnel to PE 1 is down. PE 3 declares the next hop for the route through PE 1 to be unreachable. It then selects the PE 2 route as the best route and installs it in the VRF's IP routing table.

On the other hand, if the VRFs in PE 1 and PE 2 share the same RD, the route reflector reflects only the best route, in this example the route through PE 1. If PE 1 goes down in this situation, PE 4 still reflects the route through PE 1. When PE 3 resolves the route, it finds that the tunnel is down and declares the next hop to be unreachable. Traffic then suffers a delay due to slow reconvergence.

Assigning a unique RD for each VRF can be useful for other reasons as well:

- PE-to-PE forwarding requires an MPLS tunnel from the ingress PE router to the egress PE router. In some topologies, such as networks with a sparse RSVP-TE mesh where the route reflector is not in the forwarding path, little correlation exists between the presence of an MPLS tunnel or IP connectivity from the route reflector to the egress PE router and the presence of the MPLS tunnel from the ingress PE router to the egress PE router.

  For these networks, relying on the ingress PE router is better than relying on the route reflector to decide which route is best. For this to work properly, the ingress PE router must be able to choose from all available paths, which in turn requires that each VRF have a unique RD.

- If each VRF has a unique RD and the ingress PE router has all feasible paths to choose from, you can configure IBGP multipath and ECMP traffic over multiple PE-to-PE MPLS tunnels. This configuration is not possible if you use the same RD on multiple VRFs, because the ingress PE router in that case picks a single route that resolves to a single MPLS tunnel that is used end-to-end.

Fast Reconvergence by Means of Reachability Checking

You might not want to assign different RDs for each VRF in some circumstances, such as the following:

- Allocating a unique RD for each VRF might be an administrative burden.

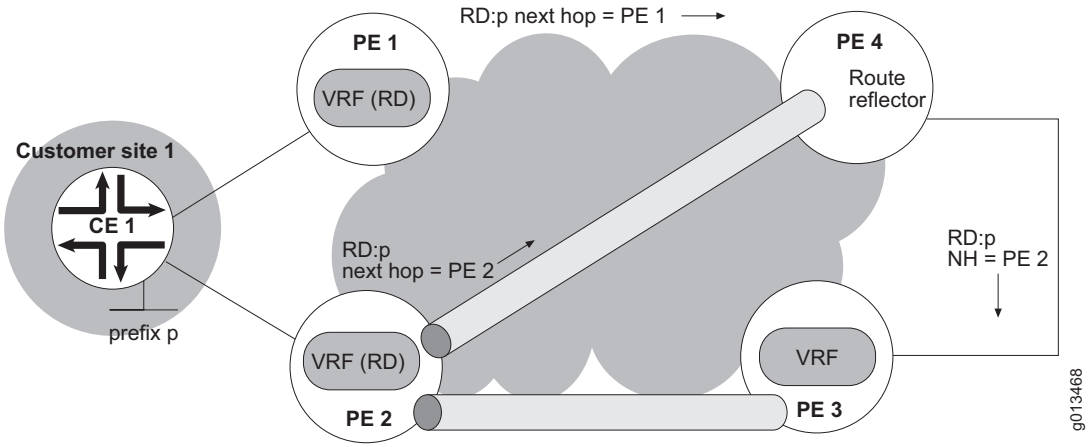
- If the network is already in operation and configured with the same RD for all VRFs in a given VPN, then changing the RDs might affect service.

- Each route reflector might act as an arbiter for a geographic area and be responsible for maintaining a list of all feasible paths to egress PE routers that can be used to reach a given prefix. Because the route reflector selects only one best path and reflects that single best path toward its clients and nonclients, the amount of state in the network is reduced. The core of the network and other geographic areas need only the one best route to each prefix in a given remote geographical area.

You can use the **check-vpn-next-hops** command to avoid the slow reconvergence problem without having to configure a unique RD for each VRF. When you issue this command, BGP verifies the reachability of the next hop on VPNv4 routes received from MP-IBGP peers before it imports those routes into a VRF. This behavior enables the VPNv4 route reflectors to take into account the reachability of the next hop when they select the best route to reflect.

Consider a topology similar to that discussed in the previous section. As before, the route through PE 1 is considered to be the best. VRFs share the same RD, but reachability checking has been enabled. In Figure 99, PE 1 has already failed, and tunnels PE 3–PE 1 and PE 4–PE 1 have gone down.

Figure 99: Topology for Fast Reconvergence by Means of Reachability Checking, After Tunnels Go Down



When the MPLS tunnel (RSVP-TE or LDP) to the next hop of the best route goes down, the VPNv4 route reflector immediately advertises the next-best route (if any) without waiting for the MP-IBGP session to go down. In this example, that route is through PE 2.

check-vpn-next-hops

- Use to enable a BGP speaker to consider the reachability of the next hop when the speaker determines which VPNv4 or VPNv6 route is the best path to a prefix.
- Verifying the reachability of the next hop is disabled by default.
- This command is available only in the context of the VPNv4 unicast and VPNv6 unicast address families. The behavior is the same for both address families.
- Use the **show ip bgp vpnv4 all summary** or **show bgp ipv6 vpnv6 all summary** command to view the status of next hop reachability checking.
- This command takes effect immediately.
- Example

  host1:pe1(config-router-af)#**check-vpn-next-hops**

- Use the **no** version to halt the verification of next-hop reachability by the BGP speaker.

Configuring BGP to Send Labeled and Unlabeled Unicast Routes

You can issue the **neighbor send-label** command to enable BGP to exchange both labeled and unlabeled unicast routes in the same address family (same AFI) over the same BGP peering session. The routes can be IPv4 or IPv6 routes. When you issue the **neighbor send-label** command, JUNOSe always proposes SAFI 4 and SAFI 1. If this command has not been configured, then JUNOSe proposes only SAFI 1.

A route is advertised as a labeled route within a given BGP peering session in either of the following cases:

- You issue the **neighbor send-label** command, but no outbound route map has been configured.
- You issue the **neighbor send-label** command and an outbound route map has been configured, and the route map executes a **set mpls-label** clause for the advertised route.

In all other cases, the route is advertised as an unlabeled route.

match mpls-label

- Use to match on MPLS-labeled routes by including as a clause in a route map. The clause matches the route only if the route has a label.
- By including this command in the appropriate route map (export, global export, global import route map), you can restrict importing or exporting to only labeled or only unlabeled routes.
- Example

  host1:pe1(config-route-map)#**match mpls-label**

- Use the **no** version to remove the configuration.

neighbor send-label

- Use to configure a BGP peer to distribute an MPLS label with the advertisements for its IPv4 and IPv6 routes.
- This command enables BGP to dynamically negotiate SAFI 1 and SAFI 4 with this neighbor.
- Example

  host1(config-router-af)#**neighbor 10.19.1.2 send-label**

- Use the **no** version to halt distribution of the MPLS label with route advertisements.

set mpls-label

- Use to configure BGP to advertise prefixes that match the route map as labeled prefixes.
- Example

  host1:pe1(config-route-map)#**set mpls-label**

- Use the **no** version to remove the configuration.

BGP Next-Hop-Self

When a BGP router reports itself as the next hop, whether because of an explicit **neighbor next-hop-self** configuration or implicitly as a result of participating in an EBGP session, BGP allocates a new in label and adds an entry to the MPLS forwarding table, creating a label-to-next-hop mapping.

When a BGP router does not report itself as the next hop, whether because of an explicit **neighbor next-hop-unchanged** configuration or implicitly as a result of a participating in an IBGP session, BGP does not allocate a new in label. Instead, if the route is advertised as a labeled route, BGP uses the existing out label. This feature is used mainly on route reflectors.

The determination to allocate an in label is made only after the outbound route map has been processed. Therefore, the in label allocation and the creation of the label-to-next-hop mapping are performed after the need is apparent, conserving the number of in labels allocated.

BGP Processing of Received Routes

BGP processes received routes differently depending on whether the route is labeled or unlabeled, unicast or VPN.

Labeled Unicast Routes

When BGP receives a labeled route from a directly connected peer, BGP uses the MPLS major interface that is next to the peer IP interface to resolve the route's BGP next hop. If the MPLS major interface exists and is up, then the next hop is reachable.

When the received labeled route is not from a directly connected peer, BGP attempts to resolve the BGP indirect next hop of the route in the IP tunnel routing table. When the BGP indirect next hop is reachable, BGP adds the route to both the IP routing table and to the IP tunnel routing table. The route is added as a U-T (unicast-tunnel-usable) route.

Unlabeled Unicast Routes

When BGP receives an unlabeled route from a directly connected peer, the route's next hop is resolved to the directly connected interface.

When the received unlabeled route is not from a directly connected peer, BGP resolves the BGP indirect next hop of the route in the IP routing table. If the BGP indirect next hop is reachable, BGP adds the route to the IP routing table as a U (unicast) route.

Resolving IPv6 Indirect Next Hops

When the address of the indirect next hop is an IPv4-mapped IPv6 address, BGP resolves the indirect next hop in the IPv4 routing table and IPv4 tunnel routing table. When the indirect next hop is a native IPv6 address, the indirect next hop is resolved in the IPv6 routing table and IPv6 tunnel routing table.

Labeled VPN Routes

In the core VRF, when BGP receives a BGP-labeled VPN route from a multihop VPN peer, it attempts to resolve the BGP indirect next hop in the IP tunnel routing table. If the labeled VPN route is received from a nonmultihop peer, then the BGP indirect next hop is always resolved, because a connected route to that peer exists in the IP tunnel routing table.

Table 36 summarizes indirect next hop resolution.

Table 36: Resolution of Indirect Next Hops

Route Type	Table in Which BGP Indirect Next Hop Resolves
Unlabeled unicast	IP routing table
Labeled unicast	IP tunnel routing table, IP routing table
Labeled VPN	IP tunnel routing table

BGP Advertising Rules for Labeled and Unlabeled Routes with the Same AFI

When BGP receives a route to a prefix with the same AFI in both labeled and unlabeled forms, only one of these routes can be selected as the best route. The action taken after the best route is selected depends on whether the best route is labeled or unlabeled, and on what SAFI was previously negotiated with peers other than the one from which it received the best route. Table 37 lists the advertising action taken for the best route, whether labeled or unlabeled.

Table 37: Advertising Action Taken Following Best Route Selection

Best Route	SAFI Negotiated with Peer	Action Taken
Unlabeled	SAFI 1 and SAFI 4 (unlabeled and labeled)	Advertises unlabeled route.
Unlabeled	SAFI 1 (unlabeled)	Advertises unlabeled route.
Unlabeled	SAFI 4 (labeled)	Withdraws labeled route.
Labeled	SAFI 1 and SAFI 4 (unlabeled and labeled)	Advertises labeled route.
Labeled	SAFI 1 (unlabeled)	Withdraws unlabeled route.
Labeled	SAFI 4 (labeled)	Advertises labeled route.

BGP sends a route-refresh message for each SAFI that it has negotiated with a peer. For example, if a speaker has negotiated both SAFI 1 and SAFI 4 with a particular peer, then when you issue the **clear ip bgp neighbor soft-in** command, BGP sends two route-refresh messages to this neighbor, one for each SAFI.

Providing Internet Access to and from VPNs

Normally, hosts in a VPN cannot communicate with hosts in the Internet because the routing table in a VRF contains only routes to sites in the VPN and not routes to sites in the Internet. The exchange of traffic between a VPN and the Internet requires both of the following:

- Traffic flow from the VPN to the Internet
- Traffic flow from the Internet to the VPN

The most common, and simplest, method for providing Internet access is to configure two separate logical circuits. One logical circuit runs between the CE router and the VRF and is used for VPN traffic. The other logical circuit runs between the CE router and the parent VR of the VRF and is used for Internet traffic. These logical circuits are typically FR circuits, ATM circuits, or VLANs.

The following sections describe alternative methods of providing Internet access for situations in which having two separate logical circuits is not acceptable or desirable.

Enabling Traffic Flow from the VPN to the Internet

Traffic from a CE router arrives on a PE interface that exists in the context of a VRF. The PE router then looks up the destination address of the IP packet in the context of the VRF routing table rather than the VR routing table.

Problems

The VRF routing table lookup introduces the following complication.

- The size of the Internet routing table. Placing a full default-free Internet routing table in the VRF routing table is not feasible because it does not scale. The PE router would have to support more than 100,000,000 routes, because the full default-free Internet routing table is currently about 120,000 routes and the router must support up to 1,000 VRFs.

Solutions

The following methods enable advertising of Internet routes to VPN sites and thus enable traffic flow from the VPNs to the Internet:

- Configure default routes instead of a full default-free Internet routing table in the VRF. The default routes must point to a shared IP interface that you create on top of the layer 2 interface that points to the Internet gateway.
- Configure a single full default-free Internet routing table in the context of the parent VR and share this one table among all VRFs with the fallback global feature. Fallback global enables an additional lookup in the IP routing table of the parent VR in the event that the IP route lookup in the child VRF fails.
- When reachability to a small number of networks in the Internet is required, then configure a global import map to import only the specific route to these networks into the VRF.

You can create multiple IP interfaces on top of a single layer 2 interface. One of those interfaces is the primary IP interface for receiving and sending IP packets. The other interfaces are shared IP interfaces that are used only to send traffic.

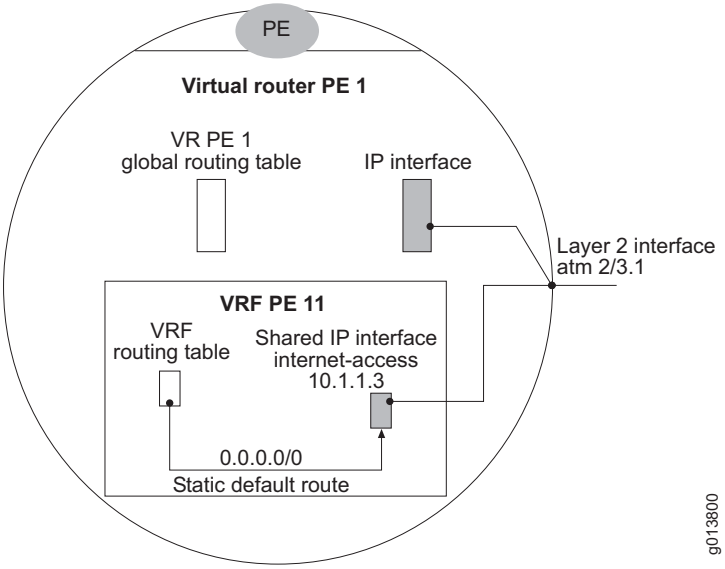
Configuring a Default Route to a Shared Interface

For the first solution you create a default route in the VRF that points to a shared IP interface. You must manually create the shared IP interface on top of the layer 2 interface that points to the Internet gateway. See Figure 100.

The main disadvantage of this approach is that if multiple Internet gateways are available, BGP cannot select the egress gateway that is optimal for each destination prefix. Because BGP has only a default route in the VRF, it has to point that single default route to a single uplink interface. All the Internet-bound traffic must flow out of that interface.

You cannot configure traffic for one prefix to flow out of one uplink interface and traffic to another prefix to flow out of another uplink interface. That behavior requires a full default-free Internet routing table in the VRF, which is a complication that you want to avoid.

Figure 100: Static Default Route for Internet Access



The following commands illustrate how to create a shared IP interface in the VRF and point a default route to it:

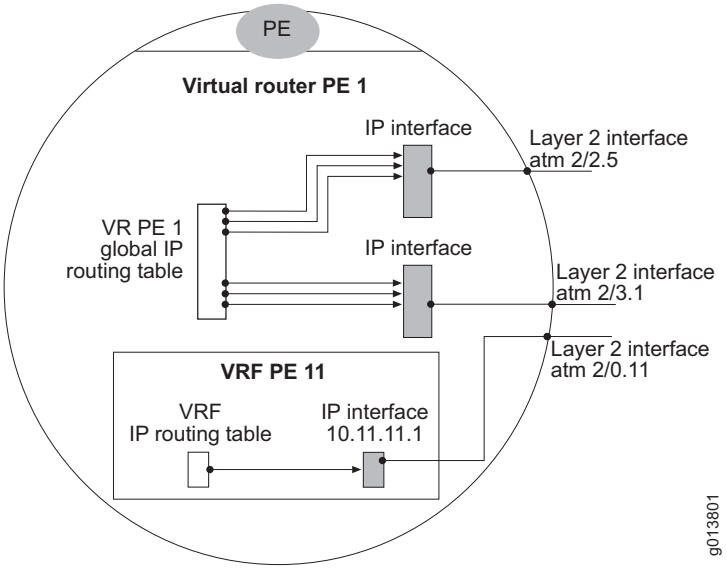
```
host1(config)#virtual-router pe1:pe11
host1:pe1:pe11(config)#interface ip internet-access
host1:pe1:pe11(config-if)#ip share-interface atm2/1.3
host1:pe1:pe11(config-if)#ip address 10.1.1.3 255.255.255.255
host1:pe1:pe11(config-if)#exit
host1:pe1:pe11(config)#ip route 0.0.0.0 0.0.0.0 ip internet-access
```

See *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 1, Configuring IP*, for information about shared IP interfaces and default routes.

Configuring a Fallback Global Option

For the second solution you use the fallback global option on the PE–CE IP interface (Figure 101). If you have configured this option, the PE router simultaneous performs two different lookups when a packet arrives from the CE router. One lookup is in the IP routing table of the VRF; the other lookup is in the IP routing table of the parent VR.

Figure 101: Fallback Global Option



If BGP finds a route in the VRF context, it uses that route. If BGP does not find a route in the VRF context but does find a route in the VR context, it falls back on the global route in the parent VR. BGP drops the packet if it does not find a route in either context.

To enable fallback global on a PE-CE IP interface:

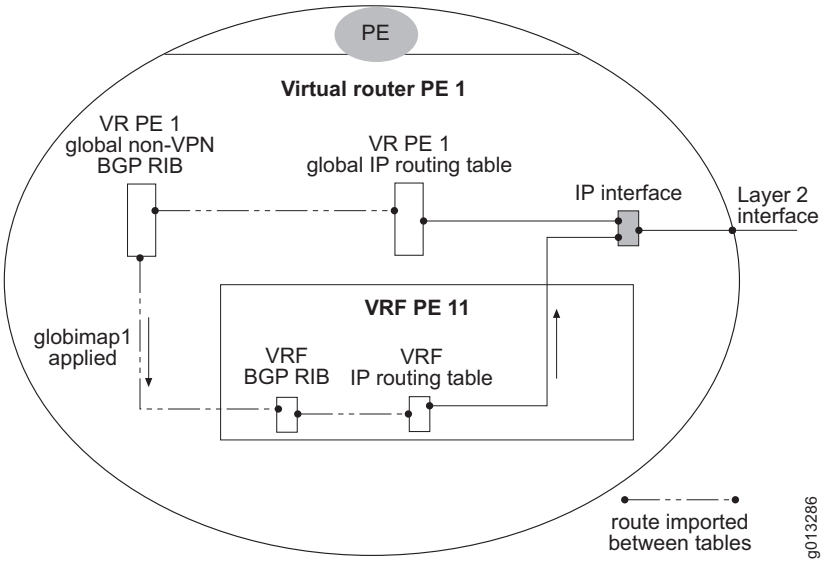
```
host1:pe1(config)#interface atm2/0.11
host1:pe1(config-if)#ip vrf forwarding pe11 fallback global
host1:pe1:pe11(config-if)#atm pvc 11 0 11 aal5snap
host1:pe1:pe11(config-if)#ip address 10.11.11.1 255.255.255.0
host1:pe1:pe11(config-if)#exit
```

See *Defining Secondary Routing Table Lookup* on page 416 for more information.

Configuring a Global Import Map for Specific Routes

For the third solution you create a global import map to import only the specific routes needed to reach the desired small number of networks in the Internet. See Figure 102 on page 446.

Figure 102: Global Import Map Applied to Routes Imported from VRF BGP RIB



The global import map enables global BGP routes to be automatically imported into the BGP RIB table in a VRF. The route map determines which routes are imported and which are not. When they are installed in the VRF routing table, the imported routes point to IP interfaces in the parent virtual router.

To configure a route map and global import map for importing specific routes.

```
host1(config)#virtual-router pe1
host1:pe1(config)#prefix-list internet-host permit 10.5.5.5/32
host1:pe1(config)#route-map globimap1
host1:pe1(config-route-map)#match ip address prefix-list internethost
host1:pe1(config-route-map)#exit
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)#route-target both 100:1
host1:pe1(config-vrf)#global import map globimap1
```

Creating a BGP Session Between the CE Router and the Parent VR

The fallback global option enables traffic that arrives at a VRF from the CE router to be sent out on the uplink determined to be optimal by using the full Internet routing table present in the parent VR.

If a CE router is multihomed to multiple PE routers, it must receive a full Internet routing table from each of the PE routers so that the CE router can determine which of the PE routers is optimal for a given Internet prefix.

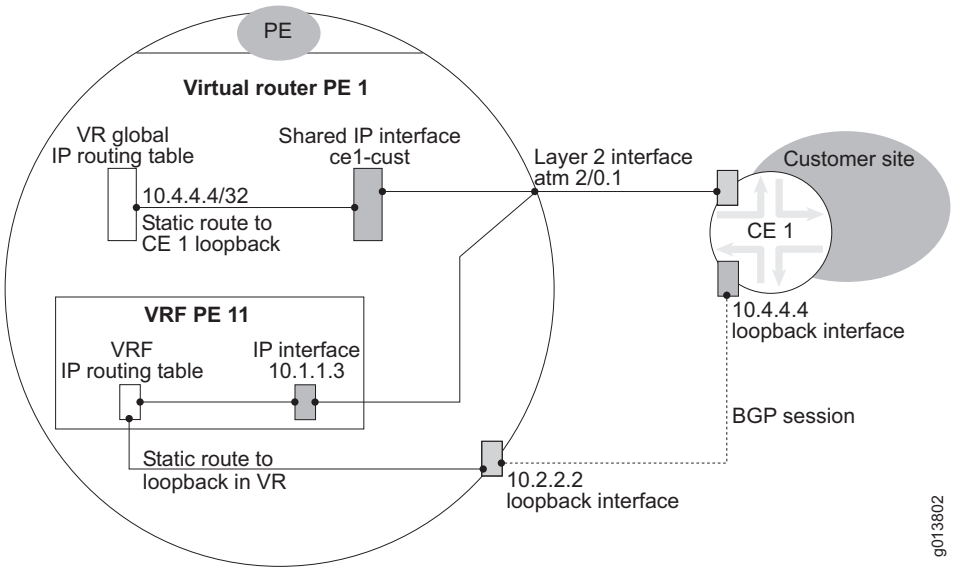
You can easily create a BGP session from the VRF to the CE router to advertise routes in the VRF to the CE router. However, doing this is insufficient because the VRF does not contain the full Internet routing table, which is present only in the parent VR.

This situation requires a BGP session from the parent VR to the CE router (Figure 103). This BGP session in turn requires a route in the VRF to the loopback interface in the parent VR that is used for BGP peering with the CE router. To achieve this configuration, you must do both of the following:

1. In the parent VR, create a shared IP interface for the PE-CE interface and point a static route to the loopback of the CE router to the shared interface.

2. Use a global import map in the VRF to import into the VRF the route to the loopback interface in the parent VR.

Figure 103: BGP Session Between CE Router and Parent VR



The following commands configure a shared IP interface in the parent VR and point a static route for the loopback in the CE router to it:

```
host1(config)#virtual-router pe1
host1:pe1(config)#interface ip ce1-cust
host1:pe1(config-if)#ip share-interface atm2/0.1
host1:pe1(config-if)#ip address 10.1.1.3 255.255.255.255
host1:pe1(config-if)#exit
host1:pe1(config)#ip route 10.4.4.4 255.255.255.255 ip ce1-cust
```

The following commands make the loopback in the parent VR reachable from the VRF by means of a global import map:

```
host1(config)#virtual-router pe1
host1:pe1(config)#prefix-list VRloop permit 10.2.2.2/32
host1:pe1(config)#route-map globimaploop
host1:pe1(config-route-map)#match ip address prefix-list VRloop
host1:pe1(config-route-map)#exit
```

```
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)#route-target both 100:1
host1:pe1(config-vrf)#global import map globimaploop
```

The following commands create a BGP session between the CE router and the parent VR.

On host 1, VR PE 1:

```
host1(config)#virtual-router pe1
host1:pe1(config)#router bgp 100
host1:pe1(config-router)#neighbor 10.4.4.4 remote-as 200
host1:pe1(config-router)# neighbor 10.4.4.4 ebgp-multihop
host1:pe1(config-router)#neighbor 10.4.4.4 update-source loopback1
host1:pe1(config-router)#exit
```

On host 2, VR CE 1:

```
host2(config)#virtual-router ce1
host2:ce1(config)#interface loopback 1
host2:ce1(config-if)#ip address 10.4.4.4 255.255.255.255
host2:ce1(config-if)#exit
host2:ce1(config)#ip route 10.2.2.2 255.255.255.255 atm2/1.1
host2:ce1(config)#router bgp 200
host2:ce1(config-router)#neighbor 10.2.2.2 remote-as 100
host2:ce1(config-router)#neighbor 10.2.2.2 ebgp-multihop
host2:ce1(config-router)#neighbor 10.2.2.2 update-source loopback1
host2:ce1(config-router)#exit
```

You must also configure either fallback global or a default route to a manually created shared interface in the VRF. See *Configuring a Fallback Global Option* on page 445 or *Configuring a Default Route to a Shared Interface* on page 444 for details.

You can use the BGP session between the CE router and the parent VR to enable the CE router to advertise prefixes within the VPN site that can be reachable from the global Internet. An alternative configuration is to use a global export map as described in *Setting Import and Export Maps for a VRF* on page 410.

Enabling Traffic Flow from the Internet to the VPN

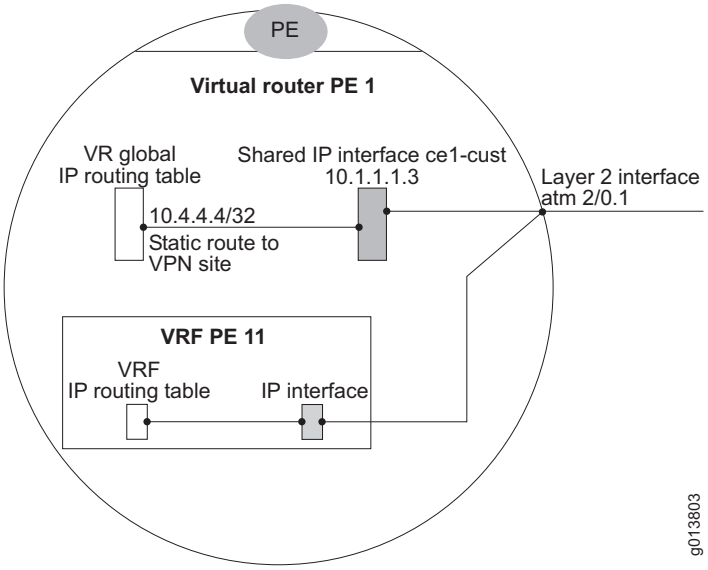
When traffic flows from the Internet to a VPN, the traffic arrives at the PE router on an interface in the global context. BGP performs a lookup in the global IP routing table, which normally does not contain VPN routes. You can use one of the following methods to advertise public VPN routes to the Internet (get the routes into the global routing table) and thus enable traffic flow from the Internet to those VPNS.

- Manually create shared interfaces in the parent VR and manually add static routes to those shared interfaces. See *Enabling VRF-to-VR Peering* on page 435 for more information.
- Export VPN routes to the global BGP RIB. See *Setting Import and Export Maps for a VRF* on page 410.

Static Routes to a Shared IP Interface

You can introduce routes to VPN sites into the global routing table by placing static routes to the VPN sites into the global table. The static routes must point to shared IP interfaces that are shares of the PE-CE interface for each particular VPN site. The static routes must then be injected into BGP (possibly as part of an aggregate) so that they can be reached from the Internet. Figure 104 illustrates this approach:

Figure 104: Static Route to Shared IP Interface



The following commands configure the shared interface and a static route:

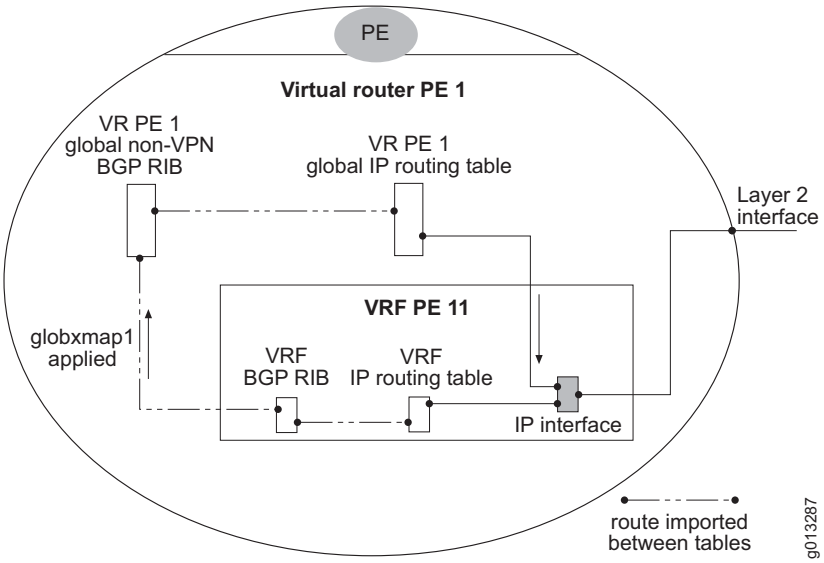
```
host1(config)#virtual-router pe1
host1:pe1(config)#interface ip ce1-cust
host1:pe1(config-if)#ip share-interface atm2/0.1
host1:pe1(config-if)# ip address 10.1.1.3 255.255.255.0
host1:pe1(config-if)#exit
host1:pe1(config)#ip route 10.4.4.4 255.255.255.255 ip ce1-cust
```

Global Export Map

The global export map enables VPN routes to be automatically exported from the BGP RIB table in a VRF to the global BGP RIB table (the BGP RIB table of the parent VR) based on policy. A route map determines which routes are exported and which are not.

When they are installed in the global IP routing table, these exported routes point to the IP interface in the VRF as shown in Figure 105. See *Global Export Maps* on page 413 for more information.

Figure 105: Global Export Map Applied to Routes Exported from VRF BGP RIB



The following commands configure the route map and global export map:

```
host1(config)#virtual-router pe1
host1:pe1(config)#access-list dot-one permit 0.0.0.1 255.255.255.0
host1:pe1(config)#route-map globxmap1
host1:pe1(config-route-map)#match ip address dot-one
host1:pe1(config-route-map)#set local-pref 200
host1:pe1(config-route-map)#exit
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)# route-target both 100:1
host1:pe1(config-vrf)#global export map globxmap1
host1:pe1(config-vrf)#exit
```

Carrier-of-Carriers IPv4 VPNs

A carrier-of-carriers VPN is a two-tiered relationship between a provider carrier and a customer carrier. In a carrier-of-carriers VPN, the provider carrier provides a VPN backbone network for the customer carrier (Tier 1). The customer carrier, in turn, provides layer 3 VPN or Internet services to its end customers (Tier 2).

This section provides the background you need to understand carrier-of-carriers VPNs in general, but deals with IPv4 VPNs. For information about carrier-of-carriers IPv6 VPNs, see *Carrier-of-Carriers IPv6 VPNs* on page 457.

The carrier-of-carriers VPN enables the customer carrier to provide the following services for its end customers:

- Traditional IP services—The customer carrier provides Internet connections for its customers and uses the provider carrier's VPN to connect its dispersed networks.
- Layer 3 VPN services—The customer carrier provides VPN services for its customers and uses the provider carrier's VPN for the backbone that connects the customer carrier's VPN sites. This environment is called a hierarchical VPN, because there are multiple tiers of VPNs—the tier-1 backbone VPN of the provider carrier and the tier-2 VPNs of the customer carrier.

In a hierarchical carrier-of-carriers VPN environment, each carrier (or ISP) maintains the internal routes of its customers in VRF tables on its PE routers. Therefore, the customer carrier's internal routes are installed into the VRF routing tables of the provider carrier's PE routers and advertised across the provider carrier's core. Similarly, the internal routes of the customer carrier's customers are installed into the VRF routing tables of the customer carrier's PE routers. The customer carrier's external routing information is exchanged by its PE routers (which connect to the provider carrier's VPN) over their own IBGP session.

NOTE: To the customer carrier, the router it uses to connect to the provider carrier's VPN is a PE router. However, the provider carrier views this device as a CE router.

Carrier-of-carriers VPNs provide the following benefits to the customer carriers:

- Reduced VPN administration—The VPN backbone is managed by the provider carrier.
- Reduced routing management—Intersite routing issues are the responsibility of the provider carrier.
- Flexibility—The VPN backbone can be used to deliver both VPN services and Internet connectivity services.

The following benefits are provided to the provider carriers:

- Reduced VPN administration—Provider carriers do not have to maintain separate VPNs for each customer carrier's end customer.
- Reduced router management—Customer carriers manage their own CE routers.
- Scalability—The provider carrier's PE routers do not maintain the end customer's external routes (as required in a traditional networking environment); the carrier-of-carriers network easily scales as the number of external routes and VPNs increases.

The following sections describe the two types of carrier-of-carriers environments.

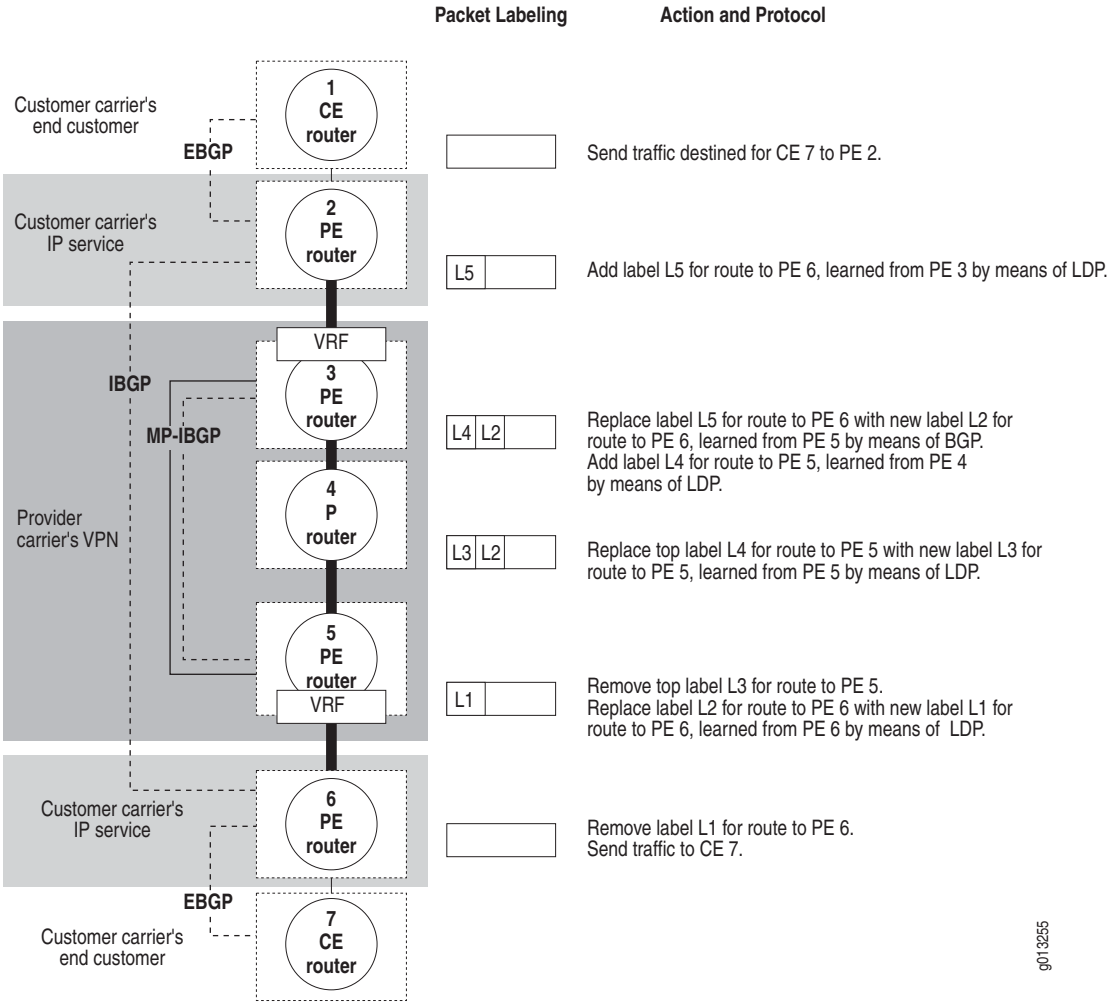
Customer Carrier as an Internet Service Provider

The provider carrier's VPN can function as the backbone for a customer carrier that provides Internet services for its customers at multiple sites. In this type of carrier-of-carriers environment, MPLS label-switched paths are established among the customer carrier's PE routers that connect to the provider carrier at each site. Routes are learned and maintained as follows:

- The customer carrier's internal routes are learned and advertised across the provider carrier's VPN. The customer carrier's external routes are *not* installed in the provider's VPN.
- The customer carrier's PE routers that connect to the provider's VPN use LDP to exchange labels for the internal routes between themselves and the provider carrier's PE router.
- The customer carrier's PE routers that connect to the provider's VPN learn external routes through IBGP sessions among themselves.

Figure 106 shows a sample carrier-of-carriers environment in which the customer carrier provides Internet connectivity services to its customers. The figure shows how the labels are added and removed as the traffic traverses the network. The label-signaling protocol is assumed to be LDP.

Figure 106: Carrier-of-Carriers Internet Service



Configuration Steps

You must complete the following configuration process when the customer carrier provides Internet connectivity for its customers.

On the provider carrier's PE router:

1. Configure MPLS.
2. Configure BGP.
3. Configure an IGP.
4. Configure LDP.
5. Configure VRF.

6. Enable carrier-of-carriers support on the VRF; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable MPLS support.

7. Enable LDP on the interface in the VRF that connects to the customer carrier's PE router.

8. Use the **show ip bgp vpnv4 vrf** *vrfname* **summary** command to verify that carrier-of-carriers support is enabled.

On the customer carrier's PE router that connects to the provider carrier's PE router:

1. Configure MPLS.

2. Configure BGP.

3. Configure an IGP.

4. Configure LDP—Enable carrier-of-carriers support on the VR; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable LDP support.

5. Enable LDP on the interface in the VR that connects to the provider carrier's PE router.

Customer Carrier as a VPN Service Provider

The carrier-of-carriers VPN can be used to create two-tiered hierarchical VPNs. In a hierarchical VPN, the provider carrier's VPN is the backbone, or tier-1 VPN, and the customer carrier provides the tier-2 VPN services to its customers.

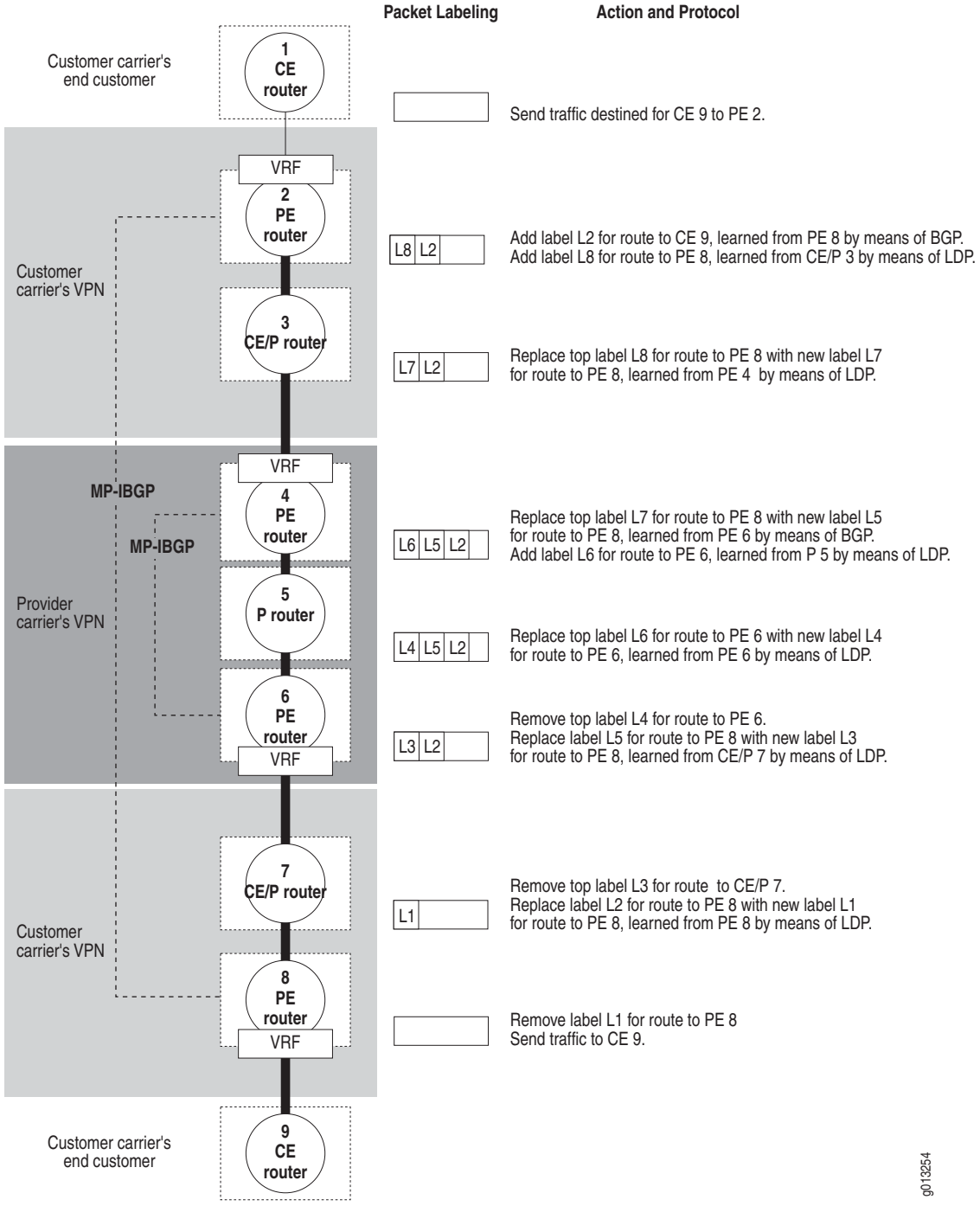
In a hierarchical VPN environment, each carrier maintains the internal routes of its customers in VRF tables on its PE routers. Routes are learned and maintained as follows:

- In the provider carrier's VPN, PE routers use MP-IBGP to exchange labeled VPN routes that correspond to the internal routes of the customer carrier's VPN sites.

- In the customer carrier's VPN, PE routers use MP-IBGP sessions to exchange labeled VPN routes that correspond to the end customer's VPN routes.

Figure 107 shows a sample carrier-of-carriers environment in which the customer carrier provides VPN services to its customers.

Figure 107: Carrier-of-Carriers VPN Service



Configuration Steps

You must complete the following configuration process when the customer carrier provides VPN services for its customers.

On the provider carrier's PE router:

1. Configure MPLS.
2. Configure BGP.
3. Configure an IGP.
4. Configure LDP.
5. Configure VRF.
6. Enable carrier-of-carriers support on the VRF; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable MPLS support.
7. Enable LDP on the interface in the VRF that connects to the customer carrier's PE router.
8. Use the **show ip bgp vpnv4 vrf** *vrfname* **summary** command to verify that carrier-of-carriers support is enabled.

On all of the customer carrier's routers, configure:

1. MPLS
2. An IGP
3. LDP

On the customer carrier's PE router that connects to the end customer's CE router, additionally configure:

1. BGP
2. VRF

Enabling Carrier-of-Carriers Support on a VRF

In a carrier-of-carriers environment, a provider carrier creates a backbone VPN that is used by a customer carrier. You must enable carrier-of-carriers support on the VRF of the provider carrier's PE device that connects to the PE device of the customer carrier.

mpls topology-driven-lsp

- Use in the context of the VRF virtual router to enable carrier-of-carriers support in a VRF. The VRF is on a PE router that is in the provider carrier's VPN and that connects to the customer carrier's PE router.
- Use the **show ip bgp vpnv4 vrf** *vrfName* **summary** command to verify whether carrier-of-carriers mode is enabled on a VRF. The output includes a line indicating the status:

  ```
  Carrier's carrier mode is enabled.
  ```

- Example

  host1:vr1:VrfA(config)#**mpls topology-driven-lsp**

- Use the **no** version to disable carrier-of-carriers mode on the VRF.

Carrier-of-Carriers Using BGP as the Label Distribution Protocol

You can run BGP instead of LDP as the label distribution protocol on the PE-CE link between the Tier 1 and the Tier 2 carriers in a carrier-of-carriers topology. This capability is available for carriers providing Internet access or VPN service to end users.

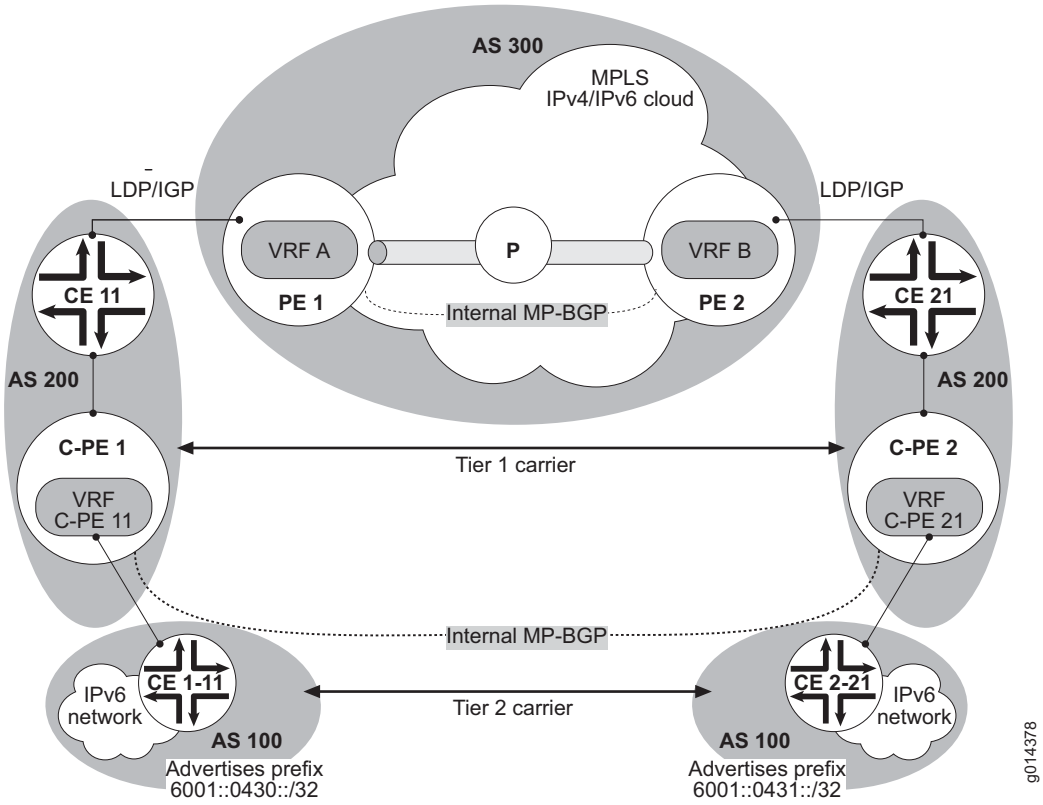
Carrier-of-Carriers IPv6 VPNs

Figure 108 illustrates a carrier-of-carrier scenario with IPv6 VPNs. MPLS labels are exchanged on the PE–CE link for customer-internal routes, but customer-external routes are not imported either into the VRFs on the PE router or into the core. VRFs maintain a routing table only for the customer-internal routes. Forwarding is accomplished primarily by label switching, without a routing table lookup.

Only customer-external routes (Tier 2 ISP routes as shown in Figure 108) can be native IPv6 addresses. Because LDP over TCP over IPv6 is not currently supported, the customer-internal routes for which LDP can give out labels (Tier 1 ISP routes in Figure 108) must be IPv4 addresses; they cannot be IPv6 addresses, whether native or IPv4-mapped.

For more information about carrier-of-carriers VPNs, see *Carrier-of-Carriers IPv4 VPNs* on page 451.

Figure 108: Carrier-of-Carrier IPv6 VPNs



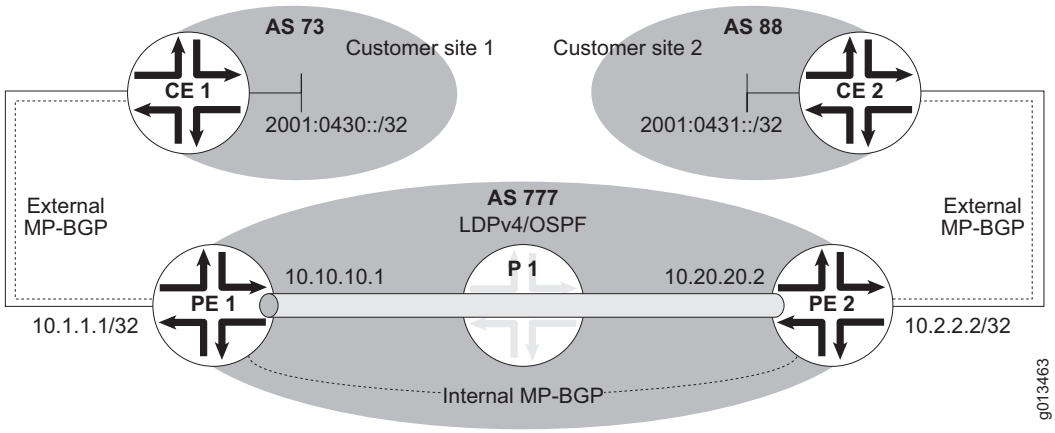
Connecting IPv6 Islands Across IPv4 Clouds with BGP

If you have not upgraded your core to IPv6, you can still provide IPv6 services to customers by connecting remote IPv6 islands across IPv4 clouds by means of MP-BGP and MPLS. An IPv6 island is a network employing IPv6 addressing, such as a customer site. The IPv4 cloud consists of the PE–P–PE core.

NOTE: You must configure an IPv6 interface in the parent VR for this feature to work.

Consider Figure 109. Each customer site is connected by means of a CE router to a PE router. The PE routers in this implementation are referred to as dual-stack BGP (DS-BGP) routers because they run both the IPv6 and IPv4 protocol stack.

Figure 109: IPv6 Tunneled over MPLS-IPv4



The PE routers learn IPv6 routes using MP-BGP over TCPv4 or TCPv6 from the CE devices. Alternatively, you can configure IPv6 static routes on the PE routers to reach the customer IPv6 networks through the CE IPv6 link. You can use any IPv6-enabled routing protocol to access the CE routers.

Use any MPLS signaling protocol to establish an MPLS base tunnel in the IPv4 core network. Each PE router runs MP-BGP over an IPv4 stack (MP-BGP/TCP/IPv4). MP-BGP advertises the customer IPv6 routes by exchanging IPv6 NLRI reachability information across the IPv4 cloud.

Each PE router announces the IPv4 address of its core-facing interface (the tunnel endpoint) to its PE peers as the BGP next hop. Because MP-BGP requires the next hop to be in the same address family as the NLRI, the IPv4 next-hop address must be embedded in an IPv6 format. The PE router advertises the IPv6 routes as labeled routes and an IPv6 next hop.

In the topology shown in Figure 109, OSPF advertises reachability of the loopback (10.1.1.1/32 and 10.2.2.1/32) and core-facing (10.10.10.1/32 and 10.20.20.2/32) interfaces of the PE routers. LDP binds label L1 to 10.1.1.1/32 on the P router.

Router CE 1 establishes an MP-BGP session over TCPv4 to PE 1 and advertises its ability to reach the IPv6 network 2001:0430::/32. The MP-BGP update message specifies an AFI value of 2 (IPv6) and a SAFI value of 1 (unicast). As the next hop in the MP-REACH-NLRI attribute, CE 1 advertises the IPv6 address of the CE 1 interface that links to PE 1.

Both IPv4 and IPv6 addresses must be configured on the PE-CE link. The IPv6 address defaults to an IPv4-compatible address that can be overridden with policy.

PE 1 and PE 2 establish an MP-BGP session using their remote loopback IPv4 addresses as neighbor addresses. Router PE 1 installs in its IPv6 global routing table the route advertised by CE 1. MP-BGP on PE 1 then binds a second-level label, L2, and advertises the route to PE 2 with an AFI value of 2 (IPv6) and a SAFI value of 4 (labeled routes). The next hop that PE 1 advertises in the MP-REACH-NLRI attribute is the IPv4 address of its loopback interface, 10.1.1.1, encoded in IPv6 format as ::10.1.1.1.

When MP-BGP on router PE 2 receives the advertisement, it associates the base tunnel (to 10.1.1.0/24, label L1) with the next hop (::10.1.1.1) that was advertised by PE 1 to reach the customer IPv6 island, 2001:0430::/32. Router PE 2 then uses MP-BGP (AFI = 2, SAFI = 1) to advertise to CE 2 its ability to reach this network.

CE 2 sends native IPv6 packets destined for the 2001:0430::/32 network to PE 2. On receipt, PE 2 performs a lookup in its global IPv6 routing table. PE 2 prepends two labels to the IPv6 header (L1–L2–IPv6) and then forwards the packet out its core-facing interface (10.2.2.2).

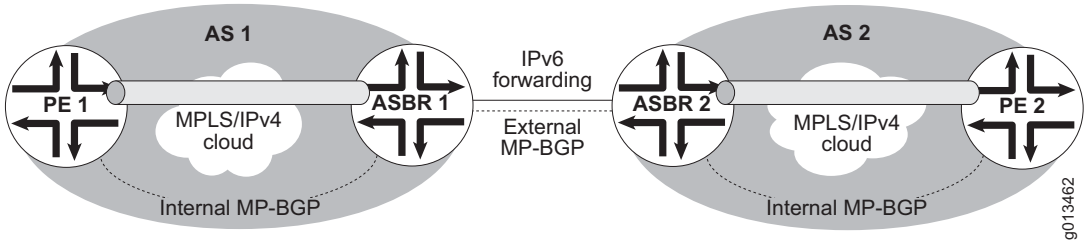
The P router does a lookup on L1 and label switches the packet toward PE 1. The P router can either replace L1 with another label or pop L1 if PE 1 requested PHP.

When PE 1 receives the packet on its core-facing interface, it pops all the labels and does a lookup in the global IPv6 routing table using the destination address in the IPv6 header. PE 1 then forwards the native IPv6 packet out to CE 1 on the IPv6 link.

Connecting IPv6 Islands Across Multiple IPv4 Domains

When the IPv6 islands are separated by multiple IPv4 domains, the autonomous system boundary routers between the IPv4 domains must be DS-BGP routers (Figure 110).

Figure 110: IPv6 Tunneled Across IPv4 Domains



Each of these AS boundary routers establishes a peer relationship with the DS-BGP routers in its own domain, creating a separate mesh of tunnels among the DS-BGP routers of each domain. Routing between PE 1–ASBR 1 in AS 1 and between PE 2–ASBR 2 in AS 2 is accomplished by means of label-switched paths.

IPv6 unlabeled routes are exchanged through the external MP-BGP session between ASBR 1 and ASBR 2. Interdomain MPLS tunnels spanning multiple ASs are not supported.

Configuring IPv6 Tunneling over IPv4 MPLS

To configure IPv6 tunneling over MPLS:

1. On PE 1, configure both an IPv4 and an IPv6 interface toward the CE router. Use an IPv4-compatible IPv6 address.

   host1(config)#**interface atm2/0.1**
   host1(config)#**atm pvc 1 0 1 aal5snap**
   host1(config)#**ip address 11.19.1.1 255.255.255.0**
   host1(config)#**ipv6 address ::11.19.1.1/126**

2. On PE 1, configure an IPv4 interface facing the core.

NOTE: For forwarding to work, you must configure at least one IPv6 interface on each line module where MPLS-encapsulated IPv6 traffic is expected. You can easily accomplish this by also configuring an IPv6 address on the core-facing interface.

   host1(config)#**interface atm3/0.1**
   host1(config)#**atm pvc 30 0 30 aal5snap**
   host1(config)#**ip address 10.10.10.1 255.255.255.0**
   host1(config)#**ip address ::10.10.10.1/120**

3. On PE 1, configure a loopback interface.

   host1(config)#**interface loopback 1**
   host1(config)#**ip address 1.1.1.1 255.255.255.0**

4. On PE 1, configure an IPv4 IGP and an MPLS signaling protocol in the core.

5. On PE 1, set up a base tunnel, or verify that one exists between the loopback addresses on the PE routers.

6. On PE 1, configure MP-BGP.

   a. Enable BGP.

      host1(config)#**router bgp 100**

   b. Configure the MP-BGP CE and PE neighbors.

      host1(config-router)#**neighbor 11.19.1.2 remote-as 65000**
      host1(config-router)#**neighbor 2.2.2.2 remote-as 100**

   c. Activate the neighbors in the IPv6 address-family.

      host1(config-router)#**address-family ipv6 unicast**
      host1(config-router-af)#**neighbor 11.19.1.2 activate**
      host1(config-router-af)#**neighbor 2.2.2.2 activate**

d. Configure the MP-BGP PE neighbor to send labeled IPv6 prefixes.

```
host1(config-router-af)#neighbor 2.2.2.2 send-label
host1(config-router-af)#neighbor 2.2.2.2 update-source loopback 1
host1(config-router-af)#neighbor 2.2.2.2 next-hop-self
host1(config-router-af)#exit-address-family
```

7. Configure the P router with an IPv4 IGP and an MPLS signaling protocol.

8. Configure the PE 2 router as you did PE 1 in Steps 1–6.

9. Configure the CE 1 and CE 2 routers.

   a. Configure both an IPv4 and an IPv6 interface toward the PE router. Use an IPv4-compatible IPv6 address.

   b. Configure an MP-BGP session to the PE router over TCPv4, and activate the IPv6 unicast address family.

neighbor send-label

- Use to cause an MP-BGP neighbor to distribute an MPLS label with its IPv6 prefix advertisements.
- If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Example

  ```
  host1(config-router-af)#neighbor 192.168.5.1 send-label
  ```

- Use the **no** version to halt distribution of the MPLS label with route advertisements.

OSPF and BGP/MPLS VPNs

Before reading this section, we recommend you be thoroughly familiar with OSPF. For detailed information about that protocol, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 5, Configuring OSPF*.

You can use BGP/MPLS VPNs to connect OSPF domains without creating OSPF adjacencies between the domains. The BGP/MPLS VPN backbone acts as either an OSPF backbone (area 0) or an OSPF area above the backbone.

In this topology, OSPF is the routing protocol between the CE router and the PE router. This OSPF link can be configured in area 0 or any other OSPF area. However, if the customer site has any connections to area 0, then at least one OSPF router configured on a CE router must have an area 0 link to a PE site. In this case, the BGP/MPLS VPN acts as if it is in an area above the OSPF backbone area. When the PE-CE link is in a nonbackbone area, the BGP/MPLS VPN acts as an OSPF backbone.

In either case, the OSPF router configured as a PE router in the BGP/MPLS VPN is always treated as an area border router (ABR) and functions as an area 0 router so that it can distribute interarea routes to the CE router. The BGP/MPLS VPN distributes both interarea and intra-area routes between PE routers as interarea, type 3 summary routes.

Distributing OSPF Routes from CE Router to PE Router

You configure OSPF in the VRF associated with the VPN and associate the interface connected to the CE router with the VRF. OSPF routes can then propagate from a CE router to a PE router when an OSPF adjacency has formed between the two routers. OSPF adds routes to the VRF's forwarding table at the PE router side with routes learned from the CE router.

Distributing Routes Between PE Routers

The OSPF routes in the VRF forwarding table are OSPF IPv4 routes, but BGP/MPLS VPNs distribute VPN-IPv4 routes by means of MP-BGP. You must configure the VRF to redistribute the OSPF routes into MP-BGP. MP-BGP converts each imported OSPF route to a VPN-IPv4 route, applies export policy to the route, and then propagates the route to a remote PE site by means of the MPLS/VPN backbone. At the destination PE router, MP-BGP places each route in the appropriate VRF forwarding table based on the import policy for each VRF and the route target associated with the route.

Preserving OSPF Routing Information Across the MPLS/VPN Backbone

MP-BGP attaches two new extended community attributes to the routes redistributed from OSPF:

- OSPF domain identifier extended community attribute
- OSPF route type extended community attribute

MP-BGP uses these attributes and the MED to preserve OSPF routing information across the BGP/MPLS VPN backbone.

OSPF Domain Identifier Attribute

The OSPF domain identifier attribute uniquely identifies the OSPF domain from which a route was redistributed into MP-BGP.

You must configure an OSPF domain ID for the VRFs on the PE router with the **domain-id** command. All VRFs that belong to a given OSPF domain must be configured with the same domain ID. If not configured, the domain ID defaults to zero. If you configure a value of zero, MP-BGP does not attach an OSPF domain identifier attribute.

If the OSPF domain ID for the destination PE router differs from the originating PE router, MP-BGP redistributes the route into OSPF as an OSPF type 5 external route.

OSPF Route Type Attribute

The route type attribute carries the OSPF area ID and LSA type, as indicated in Table 38:

Table 38: Route Types and Route Origins

Type of Route	Origin of Route
1 – intra-area route	Type 1 LSA
2 – intra-area route	Type 2 LSA
3 – interarea summary route	Type 3 LSA
5 – external route (area ID = 0)	Type 5 LSA
7 – external route (area ID = 0)	Type 7 LSA

MP-BGP uses the route type conveyed by this extended community attribute to determine the best OSPF route when it installs the routes in the VRF forwarding table on the destination PE router.

Distributing OSPF Routes from PE Router to CE Router

At the remote PE site, MP-BGP converts the OSPF routes to BGP VPN-IPv4 routes and sends them across the BGP/MPLS VPN backbone. At the destination PE router, MP-BGP must redistribute the BGP VPN-IPv4 routes back into OSPF IPv4 routes. The PE OSPF router becomes the originator of the routes, which are either type 5 external routes or type 3 internal routes. The PE router can announce the OSPF routes to the appropriate CE router through its directly connected PE-CE OSPF link.

If the route has a route type of inter or intra, it is redistributed as a type 3 summary interarea route and the destination PE router generates a type 3 LSA for it.

A route is redistributed as an external route if the route:

- Originates in an OSPF domain that is different from that of the destination PE router.
- Has a route type of 5 or 7, both of which indicate an external route.

In the first case, the PE router advertises the route as an external type 2 route. In the second case, the PE router advertises the route as an external type 2 route if the least-significant bit is set in the option byte in the route type extended community attribute; otherwise the PE router advertises the route as external type 1 route.

Preventing Routing Loops

PE routes disregard OSPF routes received from a CE router if the routes are advertised by:

- A type 3 LSA with the most-significant bit set in the LSA options field.
- A type 5 LSA that has a tag value equal to the VPN route tag associated with the OSPF VRF on that PE router.

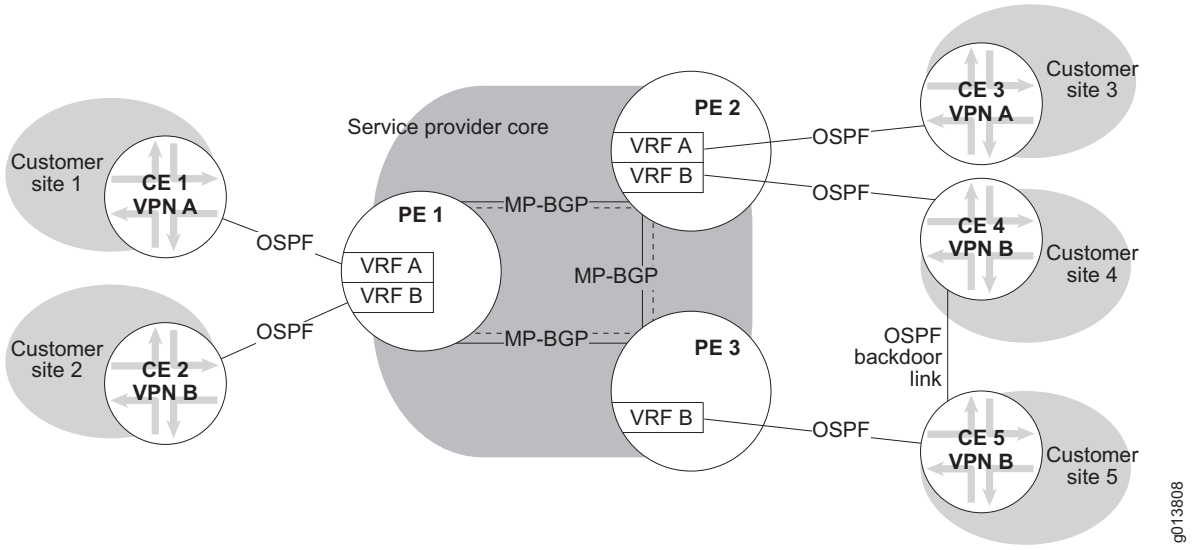
When the destination PE router originates a type 3 LSA learned from BGP to a CE router, the PE router sets the most-significant bit in the LSA options field to identify the LSA as being generated from a PE router. Doing this prevents the LSA from being passed back to the BGP/MPLS VPN through a different PE router.

When the destination PE router originates a type 5 LSA learned from BGP to a CE router, the PE router replaces the external route tag in the LSA with the VPN route tag. You configure the VPN route tag for the OPSF VRF on the PE router with the **domain-tag** command. The value of a VPN route tag must be unique within an OSPF domain, so that the same external route is not propagated back to the BGP/MPLS VPN backbone through another PE-CE link.

Using Remote Neighbors to Configure OSPF Sham Links

When you employ OSPF as the PE-CE routing protocol in a BGP/MPLS VPN and also configure OSPF backdoor links between VPN sites outside the backbone, the backdoor links are always preferred over the backbone paths between the VPN links. OSPF sham links prevent this problem, and you can implement them with OSPF remote neighbors. Consider the topology shown in Figure 111.

Figure 111: OSPF Topology with Backdoor Link



The PE routers are each running a separate logical OSPF instance for each VRF. Each of these OSPF instances has adjacencies with their directly connected CE routers and exchanges LSAs with those CE routers. The OSPF routes that are learned from a directly connected CE router are installed into the IP routing table of the VRF associated with that CE router.

The OSPF routes in the VRF's IP routing table are then redistributed into MP-BGP and advertised as VPNv4 routes to other PE routers. MP-BGP attaches extended communities to the advertised routes to carry OSPF-specific attributes such as the route type and the domain ID across the backbone.

At the remote PE router, the BGP routes are installed in the IP routing table of the VRF and then redistributed back into the logical OSPF instance for that VRF. The remote PE router uses the BGP extended communities to determine the type of LSA to send to CE routers.

As a result the intra-area OSPF routes in one VPN site appear as interarea OSPF routes at the remote VPN sites.

OSPF Backdoor Links

OSPF backdoor links typically serve as backup paths, providing a way for traffic to flow from one VPN site to the other only if the path over the backbone is broken.

However, when the OSPF backdoor link connects two sites that are in the same OSPF area, the undesired result is that the path over the OSPF backdoor link is always preferred over the path over the backbone.

In Figure 111, the OSPF backdoor link connects customer site 4 to customer site 5 directly, without going through the backbone. OSPF uses the backdoor path for traffic flow between these two sites for the following reasons:

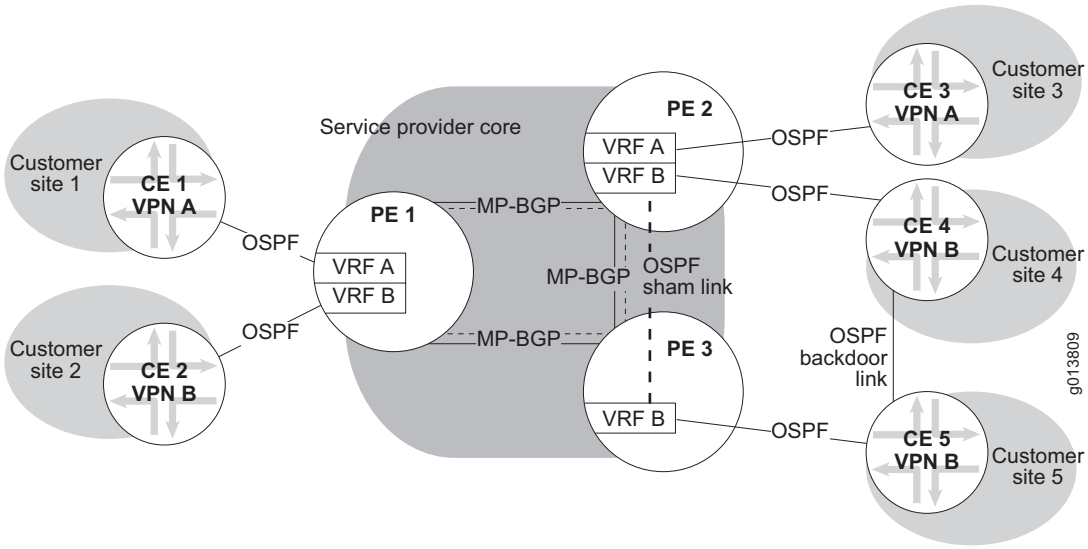
- At CE 4 and CE 5, the path over the OSPF backdoor link is an intra-area path, whereas the path over the backbone is an interarea path. OSPF always uses intra-area paths before interarea paths.
- At PE 2 and PE 3, the OSPF routes received from the respective directly connected CE routers have a better administrative distance than the IBGP routes received from the remote PE router. OSPF uses routes with better administrative distances.

OSPF Sham Links

Figure 112 shows how you can use OSPF sham links to avoid the problem created by the intra-area backdoor link. The sham link is a logical intra-area link between VRF B on PE 2 and PE 3. OSPF creates an adjacency and exchanges LSAs across the sham link. As a result, OSPF sees both the path over the backdoor link and the path over the backbone as intra-area paths. OSPF then selects the best path based on the metrics of the links and selects the sham link path, ensuring that the backdoor link is not used.

NOTE: If the VPN sites are not connected by an OSPF backdoor link or if the VPN sites are in different OSPF areas, the problem does not exist and you do not need to configure an OSPF sham link.

Figure 112: OSPF Sham Link



Use the **remote-neighbor** command to configure the OSPF sham link on both VRFs joined by the link. If a BGP route and an OSPF route to the same destination are both installed in the IP routing table, OSPF uses the OSPF route because it has a better administrative distance by definition.

If you redistribute OSPF routes into BGP in each VRF, you do not want the OSPF routes that point to sham links to be redistributed into BGP. If they were redistributed, multiple BGP routes for a single OSPF route would exist: one BGP route at each endpoint of a sham link.

Use the **dont-install-routes** command to prevent OSPF routes pointing to the sham link from being installed in the IP routing table of the VRF, and thus to prevent them from being redistributed into BGP. Forwarding still works using the MP-IBGP routes received from the remote PE router.

Use the **ttl** command to configure a TTL for the remote neighbor because the neighbor might be more than a single hop away. Use the **update-source** command to specify the loopback address used as the source address for the OSPF connection to the remote neighbor.

If you do not configure a sham link between each pair of PE routers for which a backdoor link exists, then you need to redistribute BGP routes back into OSPF.

For more information about OSPF remote neighbors, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 5, Configuring OSPF*.

dont-install-routes

- Use to prevent any OSPF routes that point directly to the OSPF remote neighbor from being installed in the IP routing table of the VR or VRF in which OSPF is running.
- Using this command avoids having many BGP routes to the same prefix by preventing OSPF routes learned over the sham link from being redistributed back into BGP even when you have configured redistribution of OSPF routes into BGP.
- Example

  host1:pe1(config-router-rn)#**dont-install-routes**

- Use the **no** version to restore the default behavior, which installs these routes in the relevant IP routing table.

remote-neighbor

- Use to configure an OSPF remote neighbor.
- Example

  host1:pe1(config-router)#**remote-neighbor 10.25.100.14 area 35672**

- Use the **no** version to remove the remote neighbor and any attributes configured for the remote neighbor.

ttl

- Use to configure a hop count by setting the value of the time-to-live field used by packets sent to an OSPF remote neighbor.
- Specify a value in the range 1–255 seconds; the default value is 1 second.
- Example

  host1:pe1(config-router-rn)#**ttl 35**

- Use the **no** version to restore the default value, 1 second.

update-source

- Use to specify the loopback interface whose local IP address is used as the source address for the OSPF connection to a remote neighbor.
- Example

  host1:pe1(config-router-rn)#**update-source loopback 1**

- Use the **no** version to delete the source address from the connection to the remote neighbor.

Configuration Tasks

At a minimum, perform the following tasks on each PE router to configure them for OSPF. For other OSPF configuration tasks, see *JUNOSe IP, IPv6, and IGP Configuration Guide, Chapter 5, Configuring OSPF*.

1. Create the VRF.

   ```
   host1(config)#ip vrf ospf2
   Proceed with new VRF creation? [confirm]
   host1(config-vrf)#rd 100:85
   host1(config-vrf)#exit
   ```

2. Start OSPF on the VRF, either from the parent VR or directly from the VRF.

   From the parent VR:

   ```
   host1(config)#router ospf 5 vrf ospf2
   ```

   From the VRF:

   ```
   host1(config)#virtual-router :ospf2
   host1:default:ospf2(config)#router ospf 5
   ```

   The command prompts in the remaining steps reflect using the latter method for starting OSPF.

3. Configure the OSPF domain ID.

   ```
   host1:default:ospf2(config-router)#domain-id 45
   ```

4. Configure the VPN route tag.

   ```
   host1:default:ospf2(config-router)#domain-tag 1200
   ```

5. Redistribute routes learned from other PE routers back into OSPF.

   ```
   host1:default:ospf2(config-router)#redistribute bgp
   ```

6. Create an address family in BGP.

   ```
   host1:default(config)#router bgp 100
   host1:default(config-router)#address-family ipv4 unicast vrf ospf2
   ```

7. Redistribute OSPF routes into BGP.

   ```
   host1:default(config-router)#redistribute ospf
   ```

domain-id

- Use to set the OSPF domain ID for an OSPF VRF on a PE router; the default value is zero.
- Use the same domain ID for all OSPF VRFs in a given OSPF domain.
- When the value is zero, MP-BGP does not attach an OSPF domain identifier attribute when it converts an OSPF route to an MP-BGP route to cross the BGP/MPLS VPN.
- Example

  host1:default:ospf2(config-router)#**domain-id 45**

- Use the **no** version to restore the default value.

domain-tag

- Use to set the VPN route tag for an OSPF VRF on a PE router.
- The default value is a 32-bit number based on the AS number of the BGP/MPLS VPN backbone, with the first 16 bits set to 1110 0000 0000 0000, followed by the 16 bits representing the AS number.
- Example

  host1:default:ospf2(config-router)#**domain-tag 1200**

- Use the **no** version to restore the default value.

Configuring VPLS

You can configure one or more instances of the Virtual Private LAN Service (VPLS), referred to as *VPLS instances*, on the router. VPLS is a BGP-MPLS application that has much in common with BGP/MPLS VPNs. VPLS employs a layer 2 virtual private network (VPN) to connect multiple individual LANs across a service provider's MPLS core network. The geographically dispersed multiple LANs functions as a single virtual LAN.

For details about configuring and using VPLS, see *Chapter 8, Configuring VPLS*.

Configuring L2VPNs

You can configure one or more instances of a Layer 2 Virtual Private Network (L2VPN), referred to as *L2VPN instances*, on the router. An L2VPN, sometimes referred to as Virtual Private Wire Service (VPWS), is a BGP-MPLS application that has much in common with BGP/MPLS VPNs. L2VPNs employ layer 2 services over MPLS to build a topology of point-to-point connections that connect end customer sites in a VPN. L2VPNs provide an alternative to private networks that have been provisioned by means of dedicated leased lines or by means of layer 2 virtual circuits that employ ATM or Frame Relay. L2VPNs enable the sharing of a provider's core network infrastructure between IP and L2VPN services, reducing the cost of providing those services.

For details about configuring and using L2VPNs, see *Chapter 11, Configuring L2VPNs*.

Monitoring BGP/MPLS VPNs

To view BGP/MPLS VPN settings, you can issue the following **show** commands as well as any of the **show ip bgp** commands and some of the **show bgp ipv6** commands described in *Chapter 1, Configuring BGP Routing*. See ***Chapter 2, Configuring MPLS***, for information about **show** commands to monitor MPLS settings.

Use the **debug ip mbgp** command to get information about problems with BGP or the network.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E-series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot*/*adapter*/*port*).

debug ip mbgp

- Use to display information about MP-BGP logs for inbound or outbound events, or both.
- Example

  host1#**debug ip mbgp**

- There is no **no** version, but you can use the **undebug ip mbgp** command to disable display of information previously enabled with the **debug ip mbgp** command.

show ip bgp next-hops

- Use to display information about BGP next hops.
- Specify all VRFs or a particular VRF, and all indirect next hops or a particular indirect next hop.
- Field descriptions
  - Indirect next-hop—BGP next-hop attribute as received in the BGP update message
  - Resolution—Describes where the indirect next hop is resolved: the IP routing table, the IP tunnel routing table, or both, and whether this is in a VR or VRF
  - IP indirect next-hop index—Index number of the IP indirect next hop that this BGP indirect next hop resolves to
  - MPLS indirect next-hop index—Index number of the MPLS indirect next hop that this BGP indirect next hop resolves to

- Reachable—Indicates whether or not the indirect next hop is reachable.

  For labeled unicast routes, the following rules apply:

  - When it is received from a nonmultihop peer, the indirect next hop is reachable if the MPLS major interface next to the peer IP interface exists and is operationally up.
  - When it is received from other types of peers,the indirect next hop is reachable if an entry exists in the IP tunnel routing table that resolves this indirect next-hop address.

  For unlabeled unicast routes, the following rules apply:

  - When it is received from an nonmultihop peer, the indirect next hop is reachable through the directly connected peer interface.
  - When it is received from other type of peers, the indirect next hop is reachable if an entry exists in the IP routing table that resolves this indirect next-hop address.

  For VPN labeled routes in a VRF, the following rules apply:

  - When it is received in a core VRF from a remote multihop IBGP or EBGP VPN peer, the indirect next hop is reachable if an entry exists in the IP tunnel routing table that resolves the next-hop address.
  - When it is received in a core VRF from a nonmultihop peer, the indirect next hop is reachable if the MPLS major interface next to the peer IP interface exists and is operationally up.
- Metric—Metric of the BGP indirect next hop
- Number of direct next-hops—Number of the equal-cost legs of direct next hops that this indirect next hop resolves to
- Direct next-hop—IP interface and next-hop IP address that resolve the BGP indirect next hop; the direct next hop can also be an IP indirect next hop or an MPLS indirect next hop when chains of next hops are in use
- Reference count—Number of label mappings of BGP routes that use this next hop

- Examples

```
host1:pe2#show ip bgp vpnv4 all next-hops
Indirect next-hop 10.1.1.1
  Resolution in IP route table of VR
       IP indirect next-hop index 10
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM4/1.20 (10.20.20.1)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 17
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 18
  Reference count is 1
```

```
Indirect next-hop 10.21.21.2
  Resolution in IP route table of VR
       IP indirect next-hop index 5
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.21 (10.21.21.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 14
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.21.mpls
  Reference count is 3

host1:pe2#show ip bgp vpnv4 vrf pe22 next-hops
Indirect next-hop 10.61.61.2
  Resolution in IP route table of VRF pe22
       IP indirect next-hop index 3
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.61 (10.61.61.2)
  Resolution in IP tunnel-route table of VRF pe22
       Not reachable
  Reference count is 2
```

show ip interface vrf

- Use to display information about the interfaces associated with the specified VRF.
- Field descriptions
  - interface—Interface type and interface specifier
  - interface status—Status of the interface
  - line protocol—Status of the line protocol
  - Link up/down trap—Status of SNMP link up/down traps on the interface
  - Internet address—IP address of the interface
  - Operational MTU—Actual MTU for the interface
  - Administrative MTU—Configured MTU for the interface
  - Operational speed—Actual speed
  - Administrative speed—Configured speed
  - Discontinuity Time—Value of sysUpTime the last time the integrity of the interface statistics was compromised
  - Router advertisement—Whether routes are advertised; enabled or disabled
  - Administrative debounce-time—Configured debounce behavior, enabled or disabled. If enabled, indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. If the state changes back before the debounce timer expires, no state change is reported.
  - Operational debounce-time—Current debounce behavior, enabled or disabled. If enabled, indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. If the state changes back before the debounce timer expires, no state change is reported.

- Access routing—When enabled, an *access* route is installed to the host on the other end of the interface
- Multipath mode—Algorithm used for ECMP: hashing of destination address and source address, or round-robin
- In Received Packets, Bytes—Total number of packets and bytes received on an IP interface
  - Unicast—Number of unicast packets and bytes received on an IP interface
  - Multicast—Number of multicast packets and bytes received on an IP interface
- In Policed Packets—Number of packets discarded on a receive IP interface because of token bucket limiting
- In Error Packets—Number of packets discarded on a receive IP interface because of IP header errors
- In Invalid Source Address Packets—Number of packets discarded on a receive IP interface because of invalid IP source address (sa-validate enabled)
- Out Forwarded Packets, Bytes—Number of packets and bytes forwarded out an IP interface
  - Unicast—Number of unicast packets and bytes forwarded out an IP interface
  - Multicast—Number of multicast packets and bytes forwarded out an IP interface
- Out Scheduler Drops Committed Packets—Number of committed packets dropped because of out queue threshold limit
- Out Scheduler Drops Conformed Packets—Number of conformed packets dropped because of out queue threshold limit
- Out Scheduler Drops Exceeded Packets—Number of exceeded packets dropped because of out queue threshold limit
- Out Policed Packets—Number of packets discarded on a forwarding IP interface because of token bucket limiting

- Examples

```
host1#show ip interface vrf vpn1
null0 is up, line protocol is up
  Network Protocols: IP
  Internet address is 255.255.255.255/255.255.255.255
  Broadcast address is 255.255.255.255
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed
```

```
atm4/0.77 is up, line protocol is up
  Network Protocols: IP
  Internet address is 7.8.7.7/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 9180  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0

host1#show ip interface vrf vpn1 brief
Interface              IP-Address     Status Protocol Description
null0                  255.255.255.255  up         up
atm4/0.77              7.8.7.7          up         up
```

show ip protocols

- Use to display information about the routing protocols associated with the VRF.
- You must specify the name of the VRF for which the protocols are displayed; otherwise, the command displays all protocols configured on the router
- Field descriptions
  - For BGP:
    - Redistributing—Protocol to which BGP is redistributing routes
    - Default local preference—Local preference value
    - IGP synchronization—Status of IGP synchronization: enabled, disabled
    - Always compare MED—Status of multiexit discrimination: enabled, disabled
    - Router flap damping—Status of route dampening: enabled, disabled
    - Administrative Distance—External, internal, and local administrative distances
    - Neighbor Address—IP address of the BGP neighbor
    - Neighbor Incoming/Outgoing update distribute list—Number of the access list for outgoing routes
    - Neighbor Incoming/Outgoing update prefix list—Number of the prefix list for incoming or outgoing routes

- Neighbor Incoming/Outgoing update prefix tree—Number of the prefix tree for incoming or outgoing routes
- Neighbor Incoming/Outgoing update filter list—Number of filter list for incoming routes
- Routing for Networks—The network for which BGP is currently injecting routes

- For IS-IS:
  - System Id—6-byte value of the router
  - IS-Type—Routing type of the router: Level 1, Level 2
  - Distance—Administrative distance for IS-IS learned routes
  - Address Summarization—Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned from other routing protocols
  - Routing for Networks—Network for which IS-IS is currently injecting routes
- For OSPF:
  - Router ID—OSPF process ID for the router
  - Distance—Administrative distance for OSPF learned routes
  - Redistributing—Protocol to which OSPF is redistributing routes
  - Address Summarization—Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned form other routing protocols
  - Routing for Networks—Network for which OSPF is currently injecting routes
- For RIP:
  - Router Administrative State—RIP protocol state. Enable means it is allowed to send and receive updates. Disable means that it may be configured but it is *not* allowed to run yet.
  - System Version—RIP versions allowed for sending and receiving RIP updates. The system version is currently set to RIP1, which sends RIP version 1 but will receive version 1 or 2. If the version is set to RIP2, the system will send and receive version 2 only. The default is configured for RIP1.
  - Update interval—Current setting of the update timer (in seconds)
  - Invalid after—Current setting of the invalid timer (in seconds)
  - hold down time—Current setting of the hold down timer (in seconds)
  - flushed interval—Current setting of the flush timer (in seconds)
  - Filter applied to outgoing route update—Access list applied to outgoing RIP route updates
  - Filter applied to incoming route update—Access list applied to incoming RIP route updates
  - Global route map—Route map that specifies all RIP interfaces on the router

- Distance—Value added to RIP routes added to the IP routing table. The default is 120.
- Interface—Interface type on which RIP protocol is running
- Redistributing—Protocol to which RIP is redistributing routes
- Routing for Networks—Network for which RIP is currently injecting routes

- Example

```
host1:pe1#show ip protocols vrf pe13
Routing Protocol is “ospf 1" with Router ID 13.13.13.1
  Distance is 110
Redistributing: bgp
  Address Summarization:
    None
  Routing for Networks:
    13.13.13.0/255.255.255.0 area 0.0.0.0
```

show ip route vrf

- Use to display the routing table of the specified VRF.
- Field descriptions
  - Protocol/Route type codes—Type of route
  - Prefix/Length—Network prefix for route in VRF routing table
  - Type—Protocol of route
  - Next Hop—IP address of the next hop to reach route
  - Dist/Met—Administrative distance and metric applied to route
  - Intf—Outgoing interface to reach route
- Example

```
host1#show ip route vrf vpn2
Protocol/Route type codes:
I1- ISIS level 1, I2- ISIS level2,
I- route type intra, IA- route type inter, E- route type external,
i- metric type internal, e- metric type external,
O- OSPF, E1- external type 1, E2- external type2,
N1- NSSA external type1, N2- NSSA external type2

Prefix/Length   Type     Next Hop    Dist/Met     Intf
--------------- -------  ----------  --------      ------------
45.5.5.5/32     Connect  45.5.5.5    0/1           fastEthernet3/0
56.5.5.0/24     Connect  56.5.5.5    0/1           atm4/0.21
```

show ip vrf

- Use to display brief information about the VRFs in this virtual router: The route target of each VRF and the interfaces attached to each VRF.
- Specify the VRF name to display the brief information only about that VRF. You must be within the context of the virtual router to which the VRF belongs.

- Field descriptions
  - VRF Name—Name of each VRF
  - Default RD—Default route distinguisher for the VRF
  - Interfaces—Interfaces configured for the VRF
- Examples

```
host1#show ip vrf
VRF Name        Default RD         Interfaces
 vpn1           1:1                null0
                                   atm4/0.77
 vpn2           1:3                null0
                                   fastEthernet3/0
                                   atm4/0.21

host1#show ip vrf vpn1
VRF Name        Default RD         Interfaces
 vpn1           1:1                null0
                                   atm4/0.77
```

show ip vrf detail

- Use to display detailed information about the VRFs in this virtual router.
- Specify the VRF name to display the brief information only about that VRF. You must be within the context of the virtual router to which the VRF belongs.
- Field descriptions
  - VRF—Name of the VRF
  - Default RD—Default route distinguisher for the VRF
  - VRF IP Router Id—IP address that uniquely identifies the router
  - Default TTL—Time to live value in the IP header
  - Reassemble Timeout—Value to time out reassembled packets
  - Interface Configured—Interface configured for the VRF
  - Import VPN Route Target Extended Communities—List of VPNs from which the VRF accepts routing information
  - Export VPN Route Target Extended Communities—List of VPNs to which the VRF sends update messages
  - Import Route-map—Route map associated with the VRF that filters and modifies routes imported to the VRF from the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).
  - Export Route-map—Route map associated with the VRF that modifies and filters routes exported by the VRF to the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes). The can filter routes text appears only if the **filter** keyword was issued for export map.

- Global Import Route-map—Route map associated with the VRF that modifies routes imported to the VRF from the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies to only IPv4 routes) or IPv6 (applies to only IPv6 routes).
- Global Export Route-map—Route map associated with the VRF that modifies routes exported by the VRF to the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).

- Example

```
host1:pe1#show ip vrf detail
VRF  pe11; Default RD 100:11
  VRF IP Router Id: 10.11.11.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.11   tun mpls:vpnEgL17-3  ip dyn-24
  Import VPN Route Target Extended Communities:
     100:1
  Export VPN Route Target Extended Communities:
     100:1
  IPv4 Import Route-map: my-v4-import-map
  IPv6 Import Route-map: my-v6-import-map
  IPv4 Export Route-map: my-v4-export-map (can not filter routes)
  IPv6 Export Route-map: my-v6-export-map (can filter routes)
  IPv4 Global Import Route-map: my-v4-global-import-map (max routes 5000)
  IPv6 Global Import Route-map: my-v6-global-import-map (max routes 1000)
  IPv4 Global Export Route-map: my-global-v4-export-map
  IPv6 Global Export Route-map: my-global-v6-export-map
  VRF  pe12; Default RD 100:12
  VRF IP Router Id: 10.12.12.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.12   tun mpls:vpnEgL18-4  ip dyn-25
  Import VPN Route Target Extended Communities:
     100:2
  Export VPN Route Target Extended Communities:
     100:2
  Import Route-map : importmap1
  Export Route-map : exportmap23 (can filter routes)
  Global Import Route-map : globalimportmap2
  Global Export Route-map : globalexportmap3
  VRF  pe13; Default RD 100:13
  VRF IP Router Id: 10.13.13.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.13 tun mpls:vpnEgL19-5 ip dyn-26
  Import VPN Route Target Extended Communities:
     100:3
  Export VPN Route Target Extended Communities:
     100:3
  No Import Route-map
  No Export Route-map
  No Global Import Route-map
  No Global Export Route-map
```

show ip vrf interfaces

- Use to display summary information about all interfaces associated with all VRFs configured in a virtual router.
- Use the **detail** keyword to display detailed information about the interfaces.
- Field descriptions
  - Interface—Interface type and interface specifier
  - IP-Address—IP address of the interface
  - Status—Status of the interface
  - Protocol—Status of the line protocol
  - VRF—Name of the VRF with which the interface is associated
  - interface status—Status of the interface
  - line protocol—Status of the line protocol
  - Link up/down trap—Status of SNMP link up/down traps on the interface
  - Internet address—IP address of the interface
  - IP Statistics Rcvd:
    - local destination—Frames with this router as their destination
    - hdr errors—Number of packets containing header errors
    - addr errors—Number of packets containing addressing errors
    - unkn proto—Number of packets received containing unknown protocols
    - discards—Number of discarded packets
  - IP Statistics Frags:
    - reasm ok—Number of reassembled packets
    - reasm req—Number of requests for reassembly
    - reasm fails—Number of reassembly failures
    - frag ok—Number of packets fragmented successfully
    - frag creates—Number of frames requiring fragmentation
    - frag fails—Number of packets unsuccessfully fragmented
  - IP Statistics Sent:
    - generated—Number of packets generated
    - no routes—Number of packets that could not be routed
    - discards—Number of packets that could not be routed that were discarded
  - ICMP Statistics Rcvd:
    - errors—Number of error packets received
    - dst unreach—Number of packets received with destination unreachable
    - time exceed—Number of packets received with time-to-live exceeded

  - param probs—Number of packets received with parameter errors
  - src quench—Number of source quench packets received
  - redirect—Number of receive packet redirects
  - echo req—Number of echo request (PING) packets
  - echo rpy—Number of echo replies received
  - timestamp req—Number of requests for a timestamp
  - timestamp rpy—Number of replies to timestamp requests
  - addr mask req—Number of address mask requests
  - addr mask rpy—Number of address mask replies
- ICMP Statistics Sent:
  - errors—Number of error packets sent
  - dst unreach—Number of packets sent with destination unreachable
  - time excd—Number of packets sent with time-to-live exceeded
  - param probs—Number of packets sent with parameter errors
  - src quench—Number of source quench packets sent
  - redirect—Number of send packet redirects
  - timestamp req—Number of requests for a timestamp
  - timestamp rpy—Number of replies to timestamp requests
  - addr mask req—Number of address mask requests
  - addr mask rpy—Number of address mask replies
- In Received Packets, Bytes—Total number of packets and bytes received on an IP interface
  - Unicast—Number of unicast packets and bytes received on an IP interface
  - Multicast—Number of multicast packets and bytes received on an IP interface
- In Forwarded Packets, Bytes—Number of packets and bytes forwarded into an output IP interface
- In Total Dropped Packets, Bytes—Total number of packets and bytes discarded on a receive IP interface
- In Policed Packets—Number of packets discarded on a receive IP interface because of token bucket limiting
- In Invalid Source Address Packets—Number of packets discarded on a receive IP interface because of invalid IP source address (sa-validate enabled)
- In Error Packets—Number of packets discarded on a receive IP interface because of IP header errors
- In Discarded Packets—Number of packets discarded on the ingress interface because of a configuration problem rather than a problem with the packet itself

- In Fabric Dropped Packets—Number of packets discarded on a receive IP interface because of internal fabric congestion
- Out Forwarded Packets, Bytes—Number of packets and bytes forwarded out an IP interface
  - Unicast—Number of unicast packets and bytes forwarded out an IP interface
  - Multicast—Number of multicast packets and bytes forwarded out an IP interface
- Out Requested Packets, Bytes—Number of packets and bytes requested to be forwarded out an IP interface
- Out Total Dropped Packets, Bytes—Total number packets and bytes dropped by an IP interface on output
- Out Scheduler Drops Committed Packets, Bytes—Number of committed packets and bytes dropped because of out queue threshold limit
- Out Scheduler Drops Conformed Packets, Bytes—Number of conformed packets and bytes dropped because of out queue threshold limit
- Out Scheduler Drops Exceeded Packets, Bytes—Number of exceeded packets and bytes dropped because of out queue threshold limit
- Out Policed Packets—Number of packets discarded on the egress interface because of token bucket limiting
- Out Discarded Packets—Number of packets discarded on the egress interface because of a configuration problem rather than a problem with the packet itself
- Out Fabric Dropped Packets—Number of packets dropped because of internal fabric congestion

- Examples

```
host1:PE1#show ip vrf interfaces
Interface              IP-Address        Status Protocol   VRF
null0                  255.255.255.255/32  up         up   pe11
atm4/0.134             4.4.4.2/24          up         up   pe11
null0                  255.255.255.255/32  up         up   pe12
ip0                    6.6.6.8/24          up         up   pe12
null0                  255.255.255.255/32  up         up   pe13
loopback1              7.7.7.2/24          up         up   pe13

host1:PE1#show ip vrf interfaces detail
null0 is up, line protocol is up
  VRF: pe11
  Link up/down trap is disabled

  Internet address is 255.255.255.255/255.255.255.255
IP statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Frags: 0 reasm ok, 0 reasm req, 0 reasm fails
         0 frag ok, 0 frag creates, 0 frag fails
  Sent:  0 generated, 0 no routes, 0 discards
```

```
ICMP statistics:
  Rcvd:  0 errors, 0 dst unreach, 0 time exceed
         0 param probs, 0 src quench, 0 redirect,
         0 echo req, 0 echo rpy
         0 timestmp req, 0 timestmp rpy
         0 addr mask req, 0 addr mask rpy
  Sent:  0 errors, 0 dst unreach, 0 time excd
         0 param probs, 0 src qnch, 0 redirect
         0 timestamp req, 0 timestamp rpy
         0 addr mask req, 0 addr mask rpy

atm4/0.134 is up, line protocol is up
  VRF: pe11
  Link up/down trap is disabled

  Internet address is 4.4.4.2/255.255.255.0
IP statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Frags: 0 reasm ok, 0 reasm req, 0 reasm fails
         0 frag ok, 0 frag creates, 0 frag fails
  Sent:  0 generated, 0 no routes, 0 discards
ICMP statistics:
  Rcvd:  0 errors, 0 dst unreach, 0 time exceed
         0 param probs, 0 src quench, 0 redirect,
         0 echo req, 0 echo rpy
         0 timestmp req, 0 timestmp rpy
         0 addr mask req, 0 addr mask rpy
  Sent:  0 errors, 0 dst unreach, 0 time excd
         0 param probs, 0 src qnch, 0 redirect
         0 timestamp req, 0 timestamp rpy
         0 addr mask req, 0 addr mask rpy
In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Forwarded Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0
    In Fabric Dropped Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  Out Requested Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Drops Committed Packets 0, Bytes 0
    Out Scheduler Drops Conformed Packets 0, Bytes 0
    Out Scheduler Drops Exceeded  Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0
    Out Fabric Dropped Packets 0
```

show mpls l2transport load-balancing-group

- Use to display information about load-balanced Martini circuits.
- For a simpler view, the **show mpls l2transport interface** command displays only the currently active VLAN or S-VLAN subinterface. Because load-balanced circuits are configured on subinterfaces on multiple ports, only one of which is active at a given time, this command does not give a complete picture of the configuration.
- Use the **member-circuits** keyword to display circuit information for the group.
- Field descriptions
  - routed to/base LSP—Identifies address of the router at the other end of the tunnel and the base tunnel that is selected to forward the traffic
  - load-balancing group—Group number
  - Martini group-id—Martini group ID number for the interface
  - state—State of the interface
  - vc-id—VC ID number for the interface
  - mtu—Maximum transmission unit for the interface
  - In label—Label sent to upstream neighbor for route; statistics below this field are the aggregate statistics for traffic from the core
  - Out label—Label received from downstream neighbor for route; statistics below this field are the aggregate statistics for traffic to the core
  - pkts—Number of packets sent across tunnel
  - hcPkts—Number of high-capacity (64-bit) packets sent across tunnel
  - octets—Number of octets sent across tunnel
  - hcOctets—Number of high-capacity (64-bit) octets sent across tunnel
  - errors—Number of packets dropped for some reason before being sent
  - queue 0—Number of the queue for which statistics are being displayed and whether the queue is under traffic class control
  - traffic class—Name of traffic class
  - bound to—Interface to which queue is bound
  - Queue length—Size of queue in length and bytes
  - Forwarded—Number of forwarded packets and bytes
  - Dropped committed—Number of committed packets and bytes dropped
  - Dropped conformed—Number of conformed packets and bytes dropped
  - Dropped exceeded—Number of exceeded packets and bytes dropped
  - discardPkts—Number of packets discarded due to lack of buffer space before being sent
  - Member Interfaces—Information about the member interfaces for the circuit
  - Interface—Interface specifier and status; active indicates it is being used for traffic from the core; if active is not displayed, interface is not currently being used for traffic, but the statistics may be valid

- member ports—Number and type of candidate ports configured for the group, including interface specifiers and state
- member circuits—Number of member circuits configured for each port and for the group

- Example 1

```
host1#show mpls l2transport load-balancing-group 100 member-circuits
routed to 10.9.1.3  on base LSP  tun mpls:lsp-de090103-32-3e
  load-balancing-group 100
  Martini group-id 2 vc-id 200002 mtu 1500
  State UP
  In  Label 57 on stack
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts

  Out Label 59 on  tun mpls:lsp-de090103-32-3e
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
    queue 0: traffic class best-effort, bound to atm-vc ATM6/0.1
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0
  Member Interfaces
    Interface fastEthernet 2/0.2  active
      Incoming Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
      Outgoing Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
    Interface fastEthernet 3/0.2
      Incoming Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
      Outgoing Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
```

- Example 2

```
host1#show mpls l2transport load-balancing-group member-circuits brief

4 member ports:
      fastEthernet 2/0  down
      fastEthernet 3/0  30 member circuits
      fastEthernet 4/0  30 member circuits
      fastEthernet 5/0  30 member circuits
    90 member circuits
```

show mpls tunnels

- Use to display status and configuration for all tunnels or for a specific tunnel in the current router context.
- A result of Incomplete Configuration in the display indicates either no tunnel endpoint or no label distribution protocol.
- Field descriptions
  - State—Status of tunnel, up or down
  - Out Label—In the default case for a BGP/MPLS VPN, the Variable Interface, which indicates that a packet exiting the interface is going through a variable interface and that one of the labels listed further in the display will be prepended to the packet
  - Mpls Statistics
    - pkts—Number of packets sent across tunnel
    - hcpkts—Number of high-capacity (64-bit) packets sent across tunnel
    - octets—Number of octets sent across tunnel
    - hcoctets—Number of high-capacity (64-bit) octets sent across tunnel
    - errors—Number of packets that are dropped for some reason before being sent
    - discardPkts—Number of packets that are discarded due to lack of buffer space before being sent
  - Labels—List of labels associated with the variable interface; one will be selected to be prepended to packets before being sent across tunnel
- Example

```
host12#show mpls tunnels

LSP vpnIngress-21 to   3.3.3.3
State:  Up
Out label is Variable Interface
102 pkts, 0 hcPkts, 13464 octets
0 hcOctets, 0 errors, 0 discardPkts
Labels:
16 17 18 19
```

undebug ip mbgp

- Use to disable the display of information about MP-BGP logs that was previously enabled with the **debug ip mbgp** command.
- Example

```
host1#undebug ip mbgp
```

- There is no **no** version.