ERICSSON

Configuring BGP/MPLS VPN

SYSTEM ADMINISTRATOR GUIDE

43/1543-CRA 119 1170/1-V1 Uen G

Contents

1 Overview

This document provides an overview of the Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network (BGP/MPLS VPN) and describes the tasks and commands used to configure, monitor, troubleshoot, and administer BGP/MPLS VPN features on the SmartEdge router.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **`Product Overview`** folder of this Customer Product Information library.

1.1 Virtual Private Networks

In its most general definition, a Virtual Private Network (VPN) is a network in which customer connectivity among multiple remote sites is deployed across a shared central infrastructure, yet still provides the same access or security as a private network.

More specifically, a BGP/MPLS VPN is a collection of policies, and these policies control connectivity among a set of sites. A customer site is connected to the service provider network, often called a backbone, by one or more ports, where the service provider associates each port with a VPN context.

BGP/MPLS VPN allows you to implement a wide range of policies; for example, within a given VPN, you can allow every site to have a direct route to every other site (full mesh), or you can restrict certain pairs of sites from having direct routes to each other (partial mesh).

1.2 VPN Topology

A typical BGP/MPLS VPN topology consists of multiple customer sites connected to a service-provider network. Customer edge (CE) routers provide customer access to the service-provider network over a data link to one or more provider edge (PE) routers. The CE routers establish an adjacency with their directly-connected PE routers, and the CE routers advertise IPv4 routes to the

PE router. The CE routers also learn IPv4 routes from their PE routers. These IPv4 routes only become VPNv4 routes once they enter the provider backbone.

In the SmartEdge™ implementation, PE routers maintain a separate VPN context for each private network. Connections to CE routers are bound to the appropriate context. Access to the service provider core is through the local context in each PE router. Because the VPN runs from private VPN context to private VPN context, the customer can have visibility into the entire network, including the private context inside the SmartEdge router, without having any visibility into the public space or to other private contexts.

PE routers can be directly connected, or can be connected through provider (P) routers. P routers have no visibility into private networks; they simply provide connectivity from one PE router to another.

PE routers can exchange routing information with CE routers using static routing, Routing Information Protocol Version 2 (RIPv2) or, for IPv6 traffic, RIPng), Open Shortest Path First (OSPF or, for IPv6 traffic, OSPFv3), or external Border Gateway Protocol (eBGP). PE routers maintain VPN routing information for the VPNs to which they are directly attached.

PE routers advertise VPN routes learned from CE routers across the service provider core by using interior Border Gateway Protocol (iBGP). All iBGP features, including route reflectors, are available to ensure scalable iBGP connectivity across the service provider core. The PE routers use Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP) to build label-switched paths (LSPs); the PE routers function as edge label-switching routers (LSRs), and each private network has its own set of LSPs. Multiprotocol Label Switching (MPLS) is then used to forward VPN data traffic across the provider's backbone.

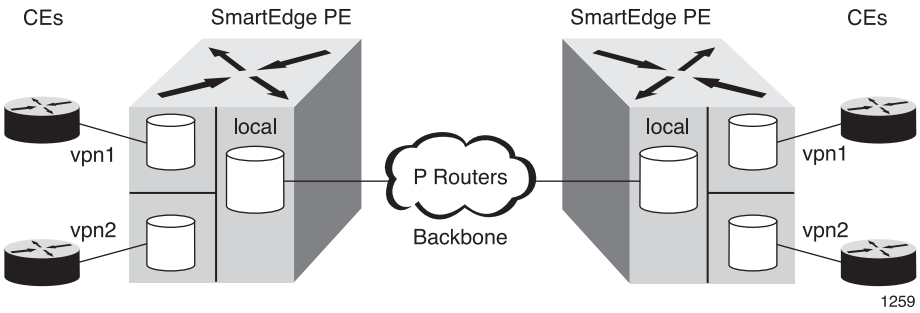

Figure 1 BGP/MPLS VPN Topology

An MPLS/BGP VPN has several components that must be operational for the VPN to function:

- The provider network routers—–PE and P routers—must run either OSPF or IS-IS to support LDP or RSVP. The link-state routing protocol discovers the paths from PE router to PE router, which is used by LDP, a signaling protocol, to build LSPs.
- PE routers configured as iBGP peers.

- Routes from the private networks are transported by the provider network, and are associated with a Forwarding Equivalence Class (FEC). BGP then assigns a next-hop and an additional tunnel label to the FEC.

  For every IP prefix in the local VPN, BGP notifies the remote VPN sites the label to attach to traffic destined to that prefix. When that traffic arrives from the remote end, the PE sends it to the next-hop given by the nexthop-label mapping.

  LSPs are then built using LDP or RSVP; the PEs function as edge LSRs, with MPLS providing the label-switching intelligence to transport VPN data across the provider backbone. For more information about LDP, see *Configuring LDP*. For more information about RSVP, see ‘‘Configuring RSVP’’ in *Configuring MPLS*.

1.3 Packet Labels

With BGP/MPLS VPNs, there are typically two labels in a packet: a tunnel label and a BGP label. The tunnel label is used in delivering the packet from an ingress PE router to the egress PE router, where the CE router is attached. The BGP label is used by the egress PE router to deliver the packet out of the interface connected to the proper CE router.

The BGP label is imposed on the packet. It is allocated to the ingress PE router by the egress PE router and installed in the routing table of the ingress router along with the route advertised by the egress router. The BGP label requires a label-switched path (LSP) across the MPLS core to identify the BGP next hop of that route. The tunnel label is imposed on the BGP-labelled packet. It is allocated by the label distribution protocol (LDP or RSVP) and corresponds to the IPv4-signaled LSP across the MPLS core between the ingress and egress PE routers. The packet is then label-switched through the core using the tunnel label. After the tunnel label is popped at the egress router, the BGP label is processed by the egress router to determine the next hop for the packet.

1.4 Multiple VPN Contexts

PE routers maintain a separate VPN context for each VPN connection. Each customer connection, such as a Frame Relay permanent virtual circuit (PVC), Asynchronous Transfer Mode (ATM) PVC, or virtual LAN (VLAN), is mapped to a specific VPN context. Multiple ports on a PE router can be associated with a single VPN context; however, it is the ability of PE routers to maintain multiple VPN contexts that supports the per-VPN segregation of routing information.

PE routers advertise VPN routes learned from CE routers using internal Border Gateway Protocol (iBGP). PE routers can maintain iBGP sessions to route reflectors as an alternative to a full mesh of iBGP sessions. Deploying multiple route reflectors enhances network scalability because it eliminates the need for any single network component to maintain all VPN routes.

MPLS is used to forward VPN data traffic across the provider's backbone, the ingress PE router functions as the ingress label edge router (LER), and the egress PE router functions as the egress LER.

1.5 VPN-IPv4 and VPN-IPv6 Address Families

VPN customers often manage their own networks with their own address space and can use private IP addresses. If globally unique IP addresses are not used, the same IP address can be used to identify different systems in different VPNs; however, BGP assumes that each IP address it carries is globally unique, so routing problems can occur. BGP/MPLS VPNs solves this problem using MP-BGP extensions, which allow BGP to carry routes from multiple address families.

Address families ensure globally unique addresses. Two address families are supported for BGP/MPLS VPNs:

- VPN-IPv4 for IPv4 addresses
- VPN-IPv6 for IPv6 addresses

A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte route distinguisher (RD), and ending with a 4-byte IPv4 address. A VPN-IPv6 address is a 24-byte quantity, beginning with an 8-byte route distinguisher (RD), and ending with a 16-byte IPv6 address. If two VPNs use the same IPv4 or IPv6 address prefix, the PE routers translate these into unique VPN-IPv4 or VPN-IPv6 address prefixes by prepending unique route distinguishers, which ensures that, if the same address is used in two different VPNs, it is possible to install two completely different routes to that address, one for each VPN.

Note: The RD contains no information about the origin of the route, or about the set of VPNs to which the route is to be advertised. The purpose of the RD is to allow you to create distinct routes to a common IP address prefix.

A PE router must be configured to associate routes that lead to a particular CE router with a particular RD. The PE router can be configured to associate all routes leading to the same CE router with the same RD, or it can be configured to associate different routes with different RDs, even if they lead to the same CE router.

1.6 Route Advertisement Among PE Routers by BGP

PE routers attached to a particular BGP/MPLS VPN must learn the addresses from that VPN. The PE router translates these addresses into VPN-IPv4 or VPN-IPv6 addresses using a configured RD. The PE router then uses the VPN-IPv4 or VPN-IPv6 routes as input to iBGP.

Within an autonomous system (AS), PE routers advertise VPN-IPv4 or VPN-IPv6 routes to each other over iPGP sessions. When a PE router advertises a route using BGP, it provides its own address as the BGP next hop. It also assigns and advertises a BGP label. The other PE routers use the advertised route as the bestpath to the destination and add the BGP label on the data packets they send. When a PE router processes a received packet that has a BGP label it assigned at the top of the stack, the PE router pops the stack, and sends the packet directly to the site to which the route leads. This usually means that it just sends the packet to the CE router from which it learned the route.

The BGP label that is advertised by a PE router requires a label-switched path (LSP) between the router that installs a route and the BGP next hop of that route. That is, an MPLS LSP must be configured for VPN route advertisements to operate.

1.7 Route Target Attributes

When a VPN-IPv4 or VPN-IPv6 route is created by a PE router, it is associated with one or more BGP extended community route target attributes. These route targets are configured with the `export route-target` command in the VPN context. The route target attribute identifies a collection of sites to which a PE router advertises routes. A PE router uses this attribute to constrain the import of remote routes into its routing tables.

Before accepting routes that have been advertised by another PE router, each VPN context on a PE router is configured with an import route target policy. These route targets are configured with the `import route-target` command in the VPN context. A PE router can only add a VPN-IPv4 or VPN-IPv6 route to a routing table for the VPN if one of the route target attributes carried with the route matches one of the import route targets on the PE router for the VPN.

1.8 Site of Origin Attribute

The site of origin attribute uniquely identifies the site from which the PE router learned the route. All routes learned from a particular site must be assigned the same site of origin attribute, even if a site has multiple connections to a single PE router, or is connected to multiple PE routers. Distinct site of origin attributes must be used for distinct sites.

The site of origin attribute is used to avoid routing loops in situations where multiple VPN sites using the AS override feature are internally connected.

1.9 PE-to-CE Route Advertisement

Access to the BGP/MPLS core can be either IPv4 or IPv6. To support IPv6 packets across the BGP/MPLS core, dual-stack PE routers are required. Possible CE-to-PE route advertising methods include:

- Static routing
- CE and PE routers can be Routing Information Protocol (RIP) peers (RIPng for IPv6), and the CE router can use RIP (or RIPng) to advertise to the PE router the set of address prefixes which are reachable at the CE router's site.
- CE and PE routers can be OSPF peers (OSPFv3 for IPv6). If the CE routers at the customer site contain more than one OSPF area, the PE-to-CE connection should be in area 0, and the CE and PE routers should be configured as area border routers (ABRs). If the CE routers at the customer site only contain a single OSPF area, then the PE-to-CE connection can be in that area, or area 0.
- CE and PE routers can be BGP peers, and the CE router can use eBGP to advertise to the PE router the set of address prefixes that are reachable at the CE router's site.

1.10 Multihop Route Redistribution for Inter-AS L3VPNs

The multihop route redistribution feature enables you to configure an inter-AS VPN; that is, a VPN between PE routers that crosses autonomous system (AS) boundaries and advertises labeled VPN-IPv4 or VPN-IPv6 routes between the source and destination PE routers in the separate ASs. This feature requires an LSP from the ingress router to the egress router.

For more information about this topology, see the "Multi-AS Backbones" section (option C) in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* .

The autonomous system border routers (ASBRs) do not maintain or advertise VPN-IPv4 or VPN-IPv6 routes. Instead, each ASBR must maintain labeled IPv4/IPv6 unicast routes to the PE router within its AS.

There are two methods to configure route redistribution for inter-AS L3VPNs. Either method creates the necessary LSP from the ingress PE router to the egress PE router:

- Using eBGP—Configure the ASBRs to use eBGP to advertise routes between autonomous systems. This method produces MPLS packets with three labels (the ASBR, the PE router, and the VPN).
- Using LDP—Configure the ASBRs to use LDP to advertise routes from one AS to the other. Use this method when the L3VPN network topology requires interoperating with Cisco equipment. This method produces MPLS packets with two labels (the PE router and the VPN).

Both methods have the following configuration prerequisites:

- On each PE router, configure OSPF (or IS-IS), MPLS, LDP and optionally RSVP, BGP, and external and internal neighbors in the local context. Also configure a VPN in a VPN context and add BGP with address families and external neighbors.
- On the P routers configure OSPF (or IS-IS), MPLS, and LDP on the interfaces to the connected routers.
- On the ASBR routers, configure OSPF (or IS-IS), MPLS, LDP and optionally RSVP, and BGP with address families, as well as external and internal neighbors.

For more information, see *Configuring OSPF*, *Configuring IS-ISConfiguring MPLS*, *Configuring LDP*, and *Configuring BGP*.

Note: All routes can be IPv4 or IPv6 routes or both.

For the tasks to configure these methods, see Section 2.9 on page 24.

For diagrams of typical topologies and configuration examples for these methods, see Section 3.4 on page 58.

1.10.1 Route Redistribution With eBGP

In this network topology using eBGP for multihop route redistribution:

- PE1 is configured to have ASBR1 as its iBGP neighbor and PE2 as its eBGP neighbor, and advertises labeled routes with ASBR1 and VPN routes with PE2.
- PE2 is configured to have ASBR2 as its iBGP neighbor and PE1 as its eBGP neighbor. It maintains and advertises labeled routes with ASBR2 an VPN routes with PE1.
- ASBR1 is configured to have PE1 as its iBGP neighbor and ASBR2 as its eBGP neighbor. It maintains a labeled route to the PE1 router and exchanges labeled VPN routes with ASBR2 using eBGP.
- ASBR2 router is configured to have PE2 as its iBGP neighbor and ASBR1 as its eBGP neighbor. It maintains a labeled route to the PE2 router and exchanges labeled VPN routes with ASBR1 using eBGP.

Note: To preserve VPN label next-hop information across the autonomous systems, the next-hop information for IPv4 VPN routes must not be changed on the local PE router when advertising to the remote PE router through multihop eBGP peering.

For a configuration example showing multihop eBGP route redistribution of labeled IPv4 and IPv4-VPN routes between two PEs across two ASs, see Section 3.4.1 on page 58.

1.10.2 Route Redistribution With LDP

In this network topology, using LDP for multihop route redistribution:

- PE1 is configured to have ASBR1 as its internal neighbor and PE2 as its external neighbor.
- PE2 router is configured to have ASBR2 as its internal neighbor and PE1 as its external neighbor.
- P1 is configured to route traffic unchanged to it's neighbors PE1 and ASBR1.
- P2 is configured to route traffic unchanged to it's neighbors PE2 and ASBR2.
- ASBR1 is configured to have PE1 as its internal neighbor and ASBR2 as its external neighbor. It maintains a labeled route to the PE1 router and exchanges labeled VPN routes with the ASBR2 router using LDP.
- ASBR2, a third-party router, is configured to have PE2 as its internal neighbor and ASBR1 as its external neighbor.

For a configuration example of multihop route redistribution using LDP, see Section 3.4.2 on page 61.

1.11 IPsec Tunnels Over BGP/MPLS VPNs

SmartEdge routers that are PE routers in an MPLS-enabled network can terminate IPsec tunnels within a BGP/MPLS VPN that traverses the MPLS core. To act as peer gateways with support for terminating IPsec tunnels, the SmartEdge routers must be equipped with an Advanced Service Engine (ASE) card and use the ASE-based IPsec VPN security service in a security-enabled routing context. With this service, clear IPv4 traffic is encrypted by the SmartEdge router on ingress and decrypted on egress. Both economical (circuitless) and circuit-based IPsec tunnel modes are supported. This allows support for deployments for services such as secure corporate access to a mobile backbone network.

To configure an IPsec tunnel over a BGP/MPLS VPN:

- Configure the BGP/MPLS VPN at both ends of the IPsec tunnel, as described in this document.
- Configure each end of the IPsec tunnel. The routing context can be the MPLS VPN context or a separate IPsec context. In the latter case, enable inter-context routing between the MPLS VPN context and the IPsec context. For more details, see Enabling Inter-Context Routing for IPsec Tunnels Over MPLS VPN.

To terminate IKE packets, an IPsec tunnel is configured with a loopback interface whose IP address is used as the local endpoint. This interface does not need to be in the routing context used by the IPsec tunnel. This loopback interface is sometimes referred to as the gateway interface.

To terminate IPsec packets, an IPsec tunnel is configured with a statically bound interface known as the tunnel interface, to which all IP routes that use the tunnel are directed. The tunnel interface is configured in the routing context in which the IPsec tunnel is configured.

When the IPsec tunnel is set up, the iBGP instance running in the VPN context installs tunnel and BGP label mappings in the routing table, and exports the local endpoint IP address and its BGP label binding to the remote gateway. The remote end uses this information to define the IPsec tunnel remote end point IP address and the BGP next hop.

For information on how to:

- Enable a security-enabled routing context, see ‘‘*Enabling a Context to Provide an ASE-Based Service*” in *Advanced Services Configuration and Operation Using the SmartEdge OS CLI*.
- Configure IPsec tunnels, see *IPsec VPN Configuration and Operation Using the SmartEdge OS CLI*.
- Enable inter-context routing among non-local contexts, see the `service inter-context routing` command in *Commands: s through show a*.

The forwarding path for IKE packets is the same for both circuit-based (the default) and economical (circuitless) IPsec tunnels. Regardless of the context in which the loopback interface is configured:

- The destination of an IKE packet received from the MPLS core is the IP address of the local endpoint for the IPsec tunnel. After the tunnel and BGP labels are removed, the IKE packet is sent to the ASE card for IPsec tunnel creation and maintenance.
- An IKE packet sent from the ASE card goes through the normal IP forwarding path. The LSP and VPN labels installed in the routing table by the iBGP instance running in the VPN context are appended to the IKE packet and the IKE packet is encapsulated with an MPLS LSP label and L2 header and sent out over the MPLS VPN.

Although there are internal differences between circuit-based and economical (circuitless) IPsec tunnels, the basic forwarding path for IPsec packets is:

- The destination of an IPsec packet received from the MPLS core is the IP address of the local endpoint for the IPsec tunnel. After the tunnel and BGP labels are removed, the IPsec packet is sent to the ASE card to remove IPsec encapsulation and decrypt the packet. The clear inner IP packet is ultimately forwarded to the CE router.

- A clear IP packet from a CE router with the destination IP address of the IPsec tunnel interface configured as its next hop is appended with the tunnel and BGP labels, sent to the ASE card for encryption and is encapsulated with the IP address of the remote IPsec tunnel endpoint into an IPsec packet, then the IPsec packet is encapsulated with an MPLS LSP label and L2 header and sent out over the MPLS VPN.

1.12 Tunneling IPv6 Over an IPv4 MPLS Core

The SmartEdge OS supports three methods for tunneling IPv6 over an IPv4 MPLS core:

- IPv6 VPN on Provider Edge router (IPv6 VPN on PE), which conforms to guidelines specified in RFC 2547bis *BGP/MPLS IP VPNs*.

  IPv6 unicast packets on an IPv6 VPN arrives from a CE on an interface in a VPN context and the VPN itself is labeled and encapsulated in an IPv4 tunnel over the IPv4 MPLS core and is routed to the corresponding VPN context of the egress PE router where the label and encapsulation is removed and the IPv6 unicast traffic continues on the IPv6 VPN to its destination. See Enabling Transport of IPv6 VPN Routes over an IPv4 MPLS Core (IPv6 VPN on PE).

- IPv6 on Provider Edge router (IPV6 on PE), as specified in RFC 4798, *Connecting IPV6 Islands over IPV4 MPLS Using IPV6 Provider Edge Routers*

  IPv6 unicast packets arrives from a CE router on an interface in the local context of the ingress PE router and is labeled and encapsulated in an IPv4 tunnel over the IPv4 MPLS core and is routed to the local context of the egress PE router where the label and encapsulation is removed and the IPv6 unicast traffic is forwarded to its destination. See Enabling Transport of IPv6 Non-VPN Routes over an IPv4 MPLS Core (IPv6 on PE).

- IPv6 overlay tunnels, as specified in RFC 2893, *Transition Mechanisms for IPv6 Hosts and Routers*.

  Overlay tunnels encapsulate IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure, such as a BGP/MPLS VPN. Configuration of overlay tunnels is described in *Configuring Single Circuit Tunnels*.

IPv6 on PE allows for global IPv6 reachability only and does not support IPv6 VPNs. 6VPE supports IPv6 VPNs, allowing the IPv6 traffic to be secured within an IPV6 VPN between the participating CEs. Either can be configured to use a BGP/MPLS or soft GRE tunnel.

Both methods require the PE routers to be dual-stack BGP-speaking routers, with both an IPv4 address and an IPv6 address. The IPv4 address must be routable in the IPv4 core so that the IPv4 route can be bound to an LDP label (provided by an IPv4 label distribution protocol, typically LDP or RSVP, enabled on each PE router). Each PE router distributes its routes to its peers, providing its own IPv4 loopback address as its BGP next hop, and a BGP label. PE routers can then exchange IPv6 prefixes over MP-BGP sessions running over IPv4 to provide the needed level of IPv6 reachability. The IPv6 address is used for the access interfaces to the CEs.

For both methods, enable the following protocols on the participating PE routers for PE-to-PE route distribution:

- MPLS
- LDP or RSVP
- An IGP (commonly OSPF)
- iBGP

  All iBGP neighbors must be configured to use IPv6 unicast address prefixes. If 6VPE is deployed, a Virtual Routing and Forwarding (VRF) table is maintained in the VPN context of each IPv6 VPN, and iBGP neighbors, configured to use IPv6 VPN address prefixes, must also be enabled in each VPN context.

Both methods require the ingress router to impose both the tunnel and BGP labels on each IPv6 packet that traverses the MPLS core, although with differences:

- The BGP label identifies the next hop on the route:
  - For 6PE: The next hop is usually the egress interface in the local context for the IPv6 packet.
  - For 6VPE: The next hop is usually the egress interface for the IPv6 VPN.

  The actual next hop for the IPv6 packet or IPv6 VPN may differ if a link group or ECMP is configured.
- The tunnel label, imposed on the BGP-labeled IPv6 packet, corresponds to the IPv4-signaled LSP across the MPLS core between the ingress and egress PE routers. The IPv6 packet is then label switched through the core using the tunnel label.

For detailed 6PE and 6VPE on BGP/MPLS VPN configuration examples, see Configuration Examples.

1.13 BGP/MPLS VPN over GRE (Soft GRE)

Encapsulating packets through Generic Routing Encapsulation (GRE) from an ingress PE router to an egress PE router is called soft GRE tunneling. Soft GRE tunnels are not Interior Gateway Protocol (IGP) visible links, and routing adjacencies are not supported across these tunnels. As a result, soft GRE tunnels have little in common with traditional (hard) GRE tunnels. The tunnel exists only in the sense of GRE encapsulation and decapsulation.

Only the ingress PE router and the egress PE router need to support the soft GRE functionality, and the PE routers can span over multiple autonomous systems.

Using soft GRE tunnels to transport MPLS-encapsulated packets is called BGP/MPLS VPN over GRE, and is used to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. BGP/MPLS VPN over GRE does not require preconfiguration of the remote GRE endpoint. These endpoints are the BGP next-hop addresses of the VPN routes and are learned dynamically through BGP.

1.14 GRE over MPLS

GRE over MPLS provides a way to establish a GRE tunnel over an MPLS LSP, allowing you to run applications, such as multicast, over the GRE tunnel. For GRE to work properly over MPLS, VPN contexts must be configured at both ends of the GRE tunnel.

To configure GRE over MPLS, you must perform the following tasks:

1. Configure BGP/MPLS VPN at both ends of the GRE tunnel.

2. Configure the GRE tunnel in the local VPN context. The tunnel remote IP address for the GRE tunnel must be an IP address in the remote VPN context.

For a detailed GRE over MPLS configuration example, see Configuration Examples.

2 Configuration and Operations Tasks

Note: In this section, the command syntax in the task tables displays only the root command.

For information about troubleshooting L3VPNs, see *Troubleshooting L3VPNs*.

To configure BGP/MPLS VPNs, perform the tasks described in the following sections.

2.1 Configuring Address Families for BGP Sessions Between Routers

2.1.1 Configuring a VPN-IPv4 Address Family for BGP Sessions Between PE Routers

To configure a VPN-IPv4 address family for BGP sessions between PE routers, perform the tasks described in Table 1. The Notes column lists the configuration mode in which you enter commands.

Table 1 Configure a VPN-IPv4 Address Family for BGP Sessions Between PE Routers

Task	Root Command	Notes
Configure a BGP routing instance in the local context, and access BGP configuration mode.	*router bgp*	Enter this command in context configuration mode. For detailed information about this command, see *Configuring BGP*.
Enable VPN-IPv4 prefixes for a BGP routing instance and enter BGP address family configuration mode.	*address-family ipv4 vpn*	Enter this command in BGP configuration mode. This command cannot be used in non-local contexts.
Enable VPN-IPv4 prefixes for a specified BGP neighbor in an iBGP session, and to access BGP neighbor address family configuration mode.	*address-family ipv4 vpn*	Enter this command in BGP neighbor configuration mode. This command cannot be used in non-local contexts.
Enable VPN-IPv4 prefixes for a specified BGP peer group, and to enter BGP peer group address family configuration mode.	*address-family ipv4 vpn*	Enter this command in BGP peer group configuration mode. This command cannot be used in non-local contexts.

2.1.2 Configuring IPv4 VPN Address Family Attributes for a BGP Routing Instance

To configure the IPv4 address family attributes for a BGP routing instance, perform the tasks described in Table 2. Enter all commands in BGP address family configuration mode, unless otherwise noted.

Table 2 Configure IPv4 Address Family Attributes for a BGP Routing Instance

Task	Root Command	Notes
Specify the use of standard IP Version 4 (IPv4) multicast or unicast address prefixes for the BGP routing instance, and access BGP address family configuration mode.	**`address-family ipv4`** command See *address-family ipv4 (BGP)*.	Enter this command in BGP router configuration mode. Include the **`uni`** or **`nni`** keyword in the **`address-family ipv4`** command.
Configure the administrative distance values for a BGP address family.	*distance (BGP address family)*	BGP uses distances to compare and prioritize routes. The lower the distance, the more preferred the route.
Enable route-flap statistics accounting for the BGP address family.	*flap-statistics*	
Enable automatic VPN route-target filtering.	*route-target filter*	
Assign a traffic index to routes installed for a BGP address family.	*table-map*	Traffic index counters are maintained on interfaces with traffic index accounting enabled. For more information about BGP attribute-based accounting, see *Configuring BGP Attribute-Based Accounting* in *Configuring Routing Policies*.
Enable the triggering of immediate BGP best-path calculation on notification of a next-hop withdrawal by the RIB, and configure next-hop scan parameters.	*router-id (contexts)*	(1)

(1) The nexthop triggered command is not available in NNI IPV4 mode.

2.2 Creating a New VPN Context

To configure a new VPN context, perform the tasks described in Table 3. Enter all commands in global configuration mode.

Table 3 Configure a New VPN Context

Task	Root Command	Notes
Enable the multiple context feature.	*service multiple-contexts*	For more information about the **`service multiple-contexts`** command, see *Configuring Contexts and Interfaces*.
Create a new VPN context and enter context configuration mode.	*context vpn-rd*	You cannot create new contexts on the system unless you have enabled the multiple context feature using the **`service multiple-contexts`** command in global configuration mode. Entering the full **`context vpn-rd`** command is required to configure a VPN context. Entering the command without the **`vpn-rd`** portion creates a context that will not be recognized as VPN-enabled.

2.3 Configuring a BGP Routing Instance in a VPN Context

To configure a BGP routing instance in a VPN context, perform the task described in Table 4. Enter the command in context configuration mode.

Table 4 Configure a BGP Routing Instance in a VPN Context

Task	Root Command	Notes
Configure a BGP routing instance in a VPN context and enter BGP configuration mode.	*router bgp*	A BGP instance is always required within a VPN context for the following reasons: • Customer routes must be distributed into BGP so they can be advertised across the iBGP sessions that connect PE routers. Customer routes can be distributed into BGP either statically or from other active routing protocols. • Route targets must also be configured within BGP address family configuration mode. BGP does not function properly in a VPN context until it is first configured in the local context. Even though an ASN is not used when configuring a BGP instance in a VPN context, this instance uses the ASN from the BGP instance in the local context for peering with CE routers. When configuring BGP peering sessions within a VPN context, only external neighbor sessions can be configured, because peering in a VPN context must only be configured with CE routers. Also, the only permitted address family is IPv4 unicast, and peer groups cannot be configured.

2.4 Configuring Multipath Load Balancing in a BGP/MPLS VPN

To configure multipath load balancing in a BGP/MPLS VPN, perform the task described in Table 5. Enter the command in BGP router configuration mode.

Table 5 Configure Multipath Load Balancing in a BGP/MPLS VPN

Task	Root Command	Notes
Configure multipath load balancing using both eBGP and iBGP equal-cost paths in a BGP/MPLS VPN.	*multi-paths eibgp*	The **`eibgp`** keyword is not supported for IPv6 traffic.

2.5 Configuring the Next-Hop Reachability Check for VPN Routes

To configure the next-hop reachability check for VPN routes, perform the task described in Table 6. Enter the command in BGP router configuration mode.

Table 6 Configure the Next-Hop Reachability Check for VPN Routes

Task	Root Command	Notes
Require the next hop of a BGP VPN path to be reachable through an MPLS LSP or a tunnel in order for a VPN route to be considered active.	*next-hop-on-lsp*	Use the **no** form of this command to enable a BGP VPN path to be considered active without requiring the next hop of a VPN path to be reachable through an MPLS LSP or a tunnel. One common application for this command is when configuring a BGP route reflector that is not part of an MPLS network, but is used to reflect BGP VPN routes to its clients within that MPLS network. In this configuration, the next hops of the VPN paths may not be reachable through an MPLS LSP or a tunnel from the route reflector's point of view. To solve the problem, use the **no** form of the this command to disable the LSP or tunnel reachability check for the next hops, and therefore allow the BGP route reflector to correctly select the best paths and reflect the best paths to its clients.

2.6 Configuring Route Targets

To configure route targets, perform the tasks described in Table 7. Enter all commands in BGP address family configuration mode.

Table 7 Configure Route Targets

Task	Root Command	Notes
Create a list of export route target extended communities for a specified VPN context.	*export route-target*	Use the ***ext-com*** argument to configure a single route target extended community, or use the **route-map** ***route-map*** construct to configure an export route map for finer control over exported Border Gateway Protocol (BGP) routes. You can configure a single route target extended community, an export route map, or both. You can add multiple export route targets on the same line, or you can issue the command multiple times with individual route targets. Export route targets are sent as extended community attributes to other provider edge (PE) routers. A route map allows you to filter routes or change attributes such as the export route target based on policy requirements. A route map may only be used when a target community value has not yet been configured. Use the optional ***ctx-name*** argument to reference a route-map in another context. If the optional ***ctx-name*** argument is not specified, then the route maps in the current context are referenced. This command can only be used in VPN contexts.

Table 7 Configure Route Targets

Task	Root Command	Notes
Create a list of import route target extended communities for a specified VPN context.	*import route-target*	You can add multiple target communities on the same line, or you can issue the command multiple times with a single target as the parameter. BGP routes learned from other PE routers that carry a specific route target extended community are imported into all VPN contexts configured with that extended community as an import route target. This command can only be used in VPN contexts.
Enable automatic BGP route target community filtering.	*route-target filter*	This command configures the local router, if it is not configured as a route reflector, to ignore all VPN routes received that are not imported into any VPN context. You can control the number of IPv4 VPN routes that the local ASBR advertise to the remote ASBR by configuring a community for exportable routes on the inbound interface of the PE router, and configuring a community based filter on the outbound interface of the local ASBR to advertise only routes that match the community.

2.7 Configuring PE-to-CE Routing

To configure PE-to-CE routing, perform the tasks described in Table 8. Enter all commands in BGP router configuration mode, unless otherwise noted.

Table 8 Configure PE-to-CE Routing

Task	Root Command	Notes
Disable the AS_PATH loop detection by accepting a route advertisement that contains the local ASN in the AS_PATH attribute.	*asloop-in*	Because enabling the **asloop-in** command disables AS_PATH loop detection, it must only be used for specific applications that require this type of behavior, and in situations with strict network control; for example, the BGP/MPLS VPN hub-and-spoke configuration, in which a hub PE router may receive routes containing its own ASN from a hub CE router. To disable AS_PATH loop detection, use the **asloop-in** command on the exporting context of the hub PE router. The **asloop-in** command is useful only when BGP is used for PE-to-CE routing. For a CE router to send a route advertisement back to the PE router from which the route is learned, the CE router must be configured as a BGP peer with the PE router configured as a member of the peer group. By default, routes are not sent back to the neighbor AS from where they are received.

Table 8 Configure PE-to-CE Routing

Task	Root Command	Notes
Replace all occurrences of a peer's ASN in the AS_PATH attribute of a route with the local ASN, when advertising the route to the peer.	*as-override*	When multiple VPN sites share the same ASN, enabling the AS override feature allows routes originating from an AS to be accepted by a router residing in the same AS. By default, the receiving router rejects the received route advertisement if the AS_PATH attribute shows that the route originated from its own AS to prevent routing loops. The **`as-override`** command is useful only when BGP is used for PE-to-CE routing. Enabling the AS override feature may result in route loops. This feature should only be used for specific applications that require this type of behavior, and in situations with strict network control. The **`as-override`** command can only be used in VPN contexts.
Enable an OSPF instance within a VPN context to treat redistributed BGP routes as VPN routes.	*vpn*	When a CE site is connected to multiple areas, the CE router's connection to a PE router should be in area 0 to allow correct handling of summary link-state advertisements (LSAs). The **`vpn`** command is useful only when OSPF is used for PE-to-CE routing.

2.8 Identifying the Specific Site from Where a Route Has Originated

To identify the specific site from where a route has originated, perform the task described in Table 9. Enter the command in BGP address family configuration mode.

Table 9 Identify the Specific Site from Where a Route Has Originated

Task	Root Command	Notes
Identify the specific site from where a route has originated.	*route-origin*	When routes are received by a PE router, the route's route-origin attribute is checked against the route origin associated with the VPN for the receive site. Received routes are rejected if the route origin values are the same. This prevents the readvertisement of routes back to their originating sites. This command is useful only when BGP is used for PE-to-CE routing.

2.9 Configuring Multihop Route Redistribution for Inter-AS L3VPNs

This section describes the tasks to configure PE, P, and ASBR routers to redistribute the routes from one AS to the other using both eBGP and LDP to redistribute the routes. Several steps are different for the two methods.

Note: For LDP route redistribution, the P routers in each AS should be configured to pass through packets unchanged in both directions.

For an overview of this feature, see Section 1.10 on page 6.

For diagrams of sample topologies and configuration examples, see Section 3.4 on page 58.

2.9.1 Configure the PE Routers

To configure each of the PE routers, in the local context, perform the steps in context configuration mode, in the local context, unless otherwise noted.

Table 10 Configure the PE Routers

Task	Root Command	Notes
For both eBGP and LDP route redistribution, configure routing in the local context:		
Enable an IGP, such as OSPF, IS-IS, or RIP on an interface.		For more information, see *Configuring OSPF*, *Configuring IS-IS*, or *Configuring RIP*
Enable MPLS on at least one interface.	*router mpls* *interface (MPLS)*	For more information, see *Configuring MPLS*.

Table 10 Configure the PE Routers

Task	Root Command	Notes
Enable LDP on at least one interface. Optional. You can also configure RSVP.	*router ldp* *interface (LDP)*	For more information, see *Configuring LDP*.
Enable BGP.	*router bgp asn*	For more information, see *Configuring BGP*
Add Unicast and VPN address families (IPv4 or IPv6 or both) as needed according to your IP and VPN configuration.	**address-family**	See Section 2.1 on page 15.
For eBGP route redistribution only, configure the external neighbor:		
Specify the external neighbor.	*neighbor*	Use the command with the **external** keyword. The IP address should be the one for the PE router in the remote AS. Perform the following tasks in BGP neighbor configuration mode.
Specify the remote AS number.	*remote-as asn*	
Enable eBGP multihop.	*ebgp-multihop*	
Add Unicast and VPN address families (IPv4 or IPv6 or both) as appropriate and enter BGP peer address-family configuration mode.	**address-family**	See Section 2.1 on page 15.
Configure the router to pass iBGP packets through to the next hop unchanged.	**next-hop-unchanged**	
Specify the internal neighbor.	*neighbor*	Use the command with the **internal** keyword. The IP address should be the ASBR. Perform the following tasks in BGP neighbor configuration mode.
Add Unicast and VPN address families as above, under adding the external neighbor.	**address-family**	See Section 2.1 on page 15.
For LDP route redistribution only, configure the internal and external neighbors:		

Table 10 Configure the PE Routers

Task	Root Command	Notes
Configure the internal neighbor.	*neighbor*	Use the command with the **`internal`** keyword. The IP address should be the one for the ASBR router in the local AS. You can also configure advertisement intervals, update sources, and address families; see the examples for PE1 and PE2 in Section 3.4 on page 58.
Configure the external neighbor.	*neighbor*	Use the command with the **`external`** keyword. The IP address should be the one for the PE router in the remote AS. You can also add the options that you added for the internal neighbor, and you can enable ebgp-multihop and other options; also see Section 3.4 on page 58.
For both eBGP and LDP route redistribution, configure a VPN:		
Create a VPN in a VPN context.	*context name vpn-rd*	Enter the command in global configuration mode. For more information, see Section 2.2 on page 16.
Enable BGP for the VPN.	*router bgp vpn*	Enter the command in context configuration mode.
Specify the internal neighbor.	*neighbor*	Use the **`internal`** keyword. The IP address should be: • For eBGP route redistribution, the local ASBR. • For LDP route redistribution, the P router on the path to the ASBR.
Also add Unicast address families (IPv4 or IPv6 or both) as appropriate.	**`address-family`**	See Section 2.1 on page 15.
For eBGP route redistribution only, enable redistribution of BGP routes:		

Table 10 Configure the PE Routers

Task	Root Command	Notes
Redistribute connected routes.	**`redistribute connected`** (IPv4 or IPv6)	See *Command List* for **`redistribute`** commands for connected devices (IPv4 or IPv6), or for IS-IS (IPv4), IS-IS (IPv6), OSPF, or RIP.
Redistribute routes from the IGP	A **`redistribute`** command for the IGP used	

2.9.2 Configure P Routers

To configure each of the P routers, such as in Figure 6, perform the steps in

Table 11 Configure P Routers

Task	Root Command	Notes
Enable an IGP, MPLS, and LDP on the interfaces to the PE and ASBR routers. Optional. Also enable RSVP on the interfaces.		For more information, see *Configuring OSPF*, *Configuring IS-IS*, *Configuring RIP*, *Configuring MPLS*, and *Configuring LDP*. For more information on RSVP, see *Configuring MPLS*.

2.9.3 Configure a SmartEdge ASBR

To configure a SmartEdge ASBR, perform the steps in Table 12:

Table 12 Configure a SmartEdge ASBR

Task	Root Command	Notes
For both eBGP and LDP route redistribution:		
Enable an IGP and MPLS, as for the PE routers		For more information, see *Configuring OSPF*, *Configuring IS-IS*, *Configuring RIP* and *Configuring MPLS*.
Add one or more route-maps.	*route-map*	Optional with eBGP route redistribution. Enter this command in context configuration mode.
Permit routes with a destination IP address specified by the IP prefix list.	*match ip address prefix-list*	Enter this command in route map configuration mode.
Configure the next-hop prefix-address	*set ip next-hop prefix-address*	Enter this command in route map configuration mode.

Table 12 Configure a SmartEdge ASBR

Task	Root Command	Notes
Enable LDP and enter router ldp configuration mode, as for the PE routers.	*router ldp* *interface (LDP)*	
For LDP route redistribution only:		
Redistribute BGP routes into LDP.	*redistribute bgp*	Enter this command in `router ldp` configuration mode. If you are using route-maps, include the optional **`route-map`** ***`map-name`*** construct.
For both eBGP and LDP route redistribution:		
Enable BGP.	*router bgp asn*	
Add Unicast address families for IPv4 and IPv6 as appropriate	**`address-family`**	See Section 2.1 on page 15.
Redistribute IGP routes into BGP.	**`redistribute`**	See *Command List* for the**`redistribute`** commands for IS-IS (IPv4), IS-IS (IPv6), OSPF, or RIP.
Specify the internal neighbor.	*neighbor*	Use the **`internal`** keyword. The IP address should be: • For eBGP route redistribution, the PE router in the local AS. • For LDP route redistribution, the P router on the route to the PE router.
Specify the external neighbor: (the ASBR in the remote AS) using the command	*neighbor*	Use the **`external`** keyword. The IP address should be the ASBR in the remote AS.

2.10 Enabling Inter-Context Routing for IPsec Tunnels Over MPLS VPN

You can route a VPN that uses an IPsec tunnel across a BGP/MPLS VPN. You can configure the termination of IKE packets in the context that the BGP/MPLS VPN is configured, or in another context.

If you configure the termination of IKE packets for negotiating the set up of an IPsec tunnel over a BGP/MPLS VPN in a context other than the context in which the BGP/MPLS VPN is configured, you must enable inter-context static routing on the PE router and configure two static routes:

- A static route defined in the MPLS VPN context specifying the IP address of the local IPsec loopback interface and the name of the IPsec context

- A static route defined in the IPsec context specifying the IP address of the remote IPsec loopback interface and the name of the MPLS VPN context.

To enable inter-context routing:

1. Use the *configure* command to access global configuration mode.
2. Use the *service inter-context routing* command to enable inter-context routing:
3. Use the *context* command, specifying the MPLS VPN context, to enter context configuration mode:

   context ***MPLS-VPN-ctx-name***
4. Use the *ip route* command to create an intercontext static route to the IPsec tunnel context:

   ip route ***local-ipsec-loopback-interface-ip-address/prefix***
   context ***IPsec-ctx-name***
5. Use the *context* command, specifying the MPLS VPN context, to enter context configuration mode:.

   context ***IPsec-ctx-name***
6. Use the *ip route* command to create an intercontext static route to the MPLS VPN context:

   ip route ***remote-ipsec-loopback-interface-ip-address/prefix***
   context ***MPLS-VPN-ctx-name***

2.11 Enabling Transport of IPv6 VPN Routes over an IPv4 MPLS Core (IPv6 VPN on PE)

To enable transport of IPv6 VPN routes over an IPv4 MPLS core, perform the tasks described in Table 3.

Table 13 Enable Transport of IPv6 VPN Routes Over an IPv4 MPLS Core

#	Task	Root Command	Notes
1.	Specify the use of standard IPv6 unicast address prefixes for the neighbors in the BGP address family:		
	Enter context configuration mode.	***context ctx-name***	Replace *ctx-name* with the name of the context in which you want to enable IPv6 prefixes.

Table 13 Enable Transport of IPv6 VPN Routes Over an IPv4 MPLS Core

#	Task	Root Command	Notes
	Configure a BGP routing instance in the VPN context and access BGP configuration mode.	**router bgp**	Enter this command in context configuration mode. For detailed information about this command, see *Configuring BGP*.
	Enables the transport of IPv6 routes over an MPLS IPv4 network.	**address-family ipv6 vpn**	Be aware that MPLS must be enabled in the local context or IPv6 packets cannot be tunneled over the IPv4 MPLS core.
	Exit BGP address family configuration mode	**exit**	
	Enter BGP neighbor configuration mode for the specified IPv6 external BGP (eBGP) neighbor.	**neighbor** *ipv6-addr* **external**	Replace *ipv6-addr* with the IPv6 address of the external neighbor, in the form *A:B:C:D:E:F:G*.
	Globally enable the IPv6 VPN address-family for BGP.	**address-family ipv6 vpn**	
	Optional. Specifies the interface used for BGP peering.	**update-source** *if-name*	Replace *if-name* with the name of the interface to be used to bring up the BGP session.
	Verify the configuration.	**show bgp neighbor**	
	Exit BGP address family configuration mode	**exit**	
	Enter context configuration mode.	**context** *ctx-name*	Replace *ctx-name* with the name of the context in which you want to enable IPv6 VPN address family for the IPv4 iBGP neighbor.
	Enter BGP neighbor configuration mode for the specified IPv4 internal BGP (iBGP) neighbor.	**neighbor** *ip-addr* **internal**	Replace *ip-addr* with the IP address of the external neighbor, in the form *A.B.C.D*.
	Globally enable the IPv6 VPN address-family for BGP.	**address-family ipv6 vpn**	
	Verify the configuration.	**show bgp neighbor**	
2.	Configure the BGP routing instance in the appropriate VPN context:		

Table 13 Enable Transport of IPv6 VPN Routes Over an IPv4 MPLS Core

#	Task	Root Command	Notes
	Enter context configuration mode for a VPN context.	*context* ***ctx-name*** **vpn-rd** ***route-distinguisher***	Replace ***ctx-name*** with the name of the VPN context in which you want to enable IPv6 prefixes. Replace ***route-distinguisher*** with the VPN route distinguisher.
	Configure a BGP routing instance in the VPN context and access BGP configuration mode.	*router bgp vpn*	Enter this command in context configuration mode. For detailed information about this command, see *Configuring BGP*.
	Specify the use of IPv6 unicast address prefixes for the BGP routing instance and enter BGP address family configuration mode.	*address-family ipv6 unicast*	Enter this command in BGP configuration mode.
	Add a route target extended community to the export target list.	*export route-target* {ext-com | **route-map** ***route-map*** [***ctx-name***]}	This step exports IPv6 routes across the BGP VPN. Use the *ext-com* argument to specify a route target extended community value to add to the export target list. Use the **route-map** ***route-map*** [***ctx-name***] construct to specify a route map to be used for this VPN context.
	Add a route target extended community to the imports target list.	*import route-target* {***ext-com*** | **route-map** ***route-map*** [***ctx-name***]}	This step imports IPv6 routes across the BGP VPN.
	Optional. Redistributes routes learned through other routing protocols into the Border Gateway Protocol (BGP) routing domain.	*redistribute*	Redistributes IPv6 routes in other protocols (PSPF, RIPng, static IPv6)
3.	Configure external BGP peering to the CE:		

Table 13 Enable Transport of IPv6 VPN Routes Over an IPv4 MPLS Core

#	Task	Root Command	Notes
	Enter BGP neighbor configuration mode for the specified IPv6 external BGP (eBGP) neighbor.	*neighbor* **`ipv6-addr external`**	Replace *ipv6-addr* with the IPv6 address of the external neighbor, in the form *A:B:C:D:E:F:G*.
	Optional. Configures the autonomous system number (ASN) of the external Border Gateway Protocol (eBGP) neighbor.	*remote-as* {**`asn`** | **`nn:nn`**}	Use the **`asn`** or **`nn:nn`**argument to specify with the ASN in integer or 4-byte integer format.
	Specify the use of IPv6 unicast address prefixes for the neighbor and enter BGP address family configuration mode.	*address-family ipv6 unicast*	
	Verify your configuration.	*show bgp route ipv6 unicast* *show bgp route ipv6 vpn* (local context only)	

2.12 Enabling Transport of IPv6 Non-VPN Routes Over an MPLS Core (IPv6 on PE)

This configuration task describes how to configure IPv6 on a PE router (IPv6 on PE), as specified in RFC 4798, *Connecting IPV6 Islands over IPV4 MPLS*. Configuring IPv6 on PE in your MPLS network enables geographically dispersed CE routers to exchange IPv6 packets and provide IPv6 services over an IPv4 MPLS core. When 6PE is configured, IPv6 packets are transported between the PEs in the MPLS core inside IPv4 LSPs. To configure 6PE in your MPLS network, you must:

- Configure a loopback interface on the PEs. The IP address of this interface is used as the BGP endpoint.
- Configure physical LSP interfaces between the PE routers.
- Enable MPLS on the LSP interfaces.
- Enable LDP or RSVP on the LSP and loopback interfaces.
- Configure BGP peering on the PE routers.
- If you are not using static routes to transmit IPv6 packets, enable an IGP (OSPF, IS-IS, or RIP) on the LSPs and loopback interfaces.

The steps that follow describe these tasks in detail.

2.12.1 Prerequisites

- The PE routers must be dual-stack BGP-speaking routers.
- The PE routers must be configured with an IPv4 address and an IPv6 address.
- The IPv4 address must be routable in the IPv4 core so that the IPv4 route can be bound to an MPLS label (provided by an IPv4 label distribution protocol, typically LDP or RSVP, enabled on each PE router). In other words, there must be a route for that IP address and you can ping that IP address from any source in the core.
- Every IPv4 core router that connects the PE routers must be MPLS enabled.
- The subnet length must match on both endpoints of any directly connected link.

2.12.2 Restrictions

- To transport static IPv6 routes over an MPLS core, you must use the **`distance`** command in the appropriate mode to configure the administrative distance between PE routes to be 8 or more (greater than the distance value for the LSP).
- This configuration is supported in the local context only.
- The PE-PE LSP configuration must match on both endpoints or the connection will fail.
- IPv6 on PE allows for global IPv6 reachability only and does not support IPv6 VPNs or IPv6 overlay tunnels. To configure the transport of IPv6 VPN Routes Over an MPLS Core (6VPE), see Enabling Transport of IPv6 VPN Routes over an IPv4 MPLS Core (IPv6 VPN on PE). To configure overlay tunnels, see *Configuring Single Circuit Tunnels*.

2.12.3 Configuration Tasks

Perform these tasks in the local context to create an IPv6 (non-VPN) tunnel between two BGP PE peer routers. Note that you must perform these steps on both PE routers:

1 Configure a loopback interface whose IP address is used as the BGP endpoint. You must configure an IPv4 address and an IPv6 address on that interface. Use the following commands:

configure
context local
interface **`if-name`** **`loopback`**
ip address **`ip-addr`**
ipv6 address **`ip-addr`**

2 Use the following commands to enable LDP or RSVP on the interface you created in Step 1:

configure
context local
router ldp or *router rsvp*
interface **`if-name`**

When entering the **`interface`** command, replace the ***`if-name`*** argument with the name of the interface you created in Step 1.

Note: For more information about LDP, see *Configuring LDP*. For more information about RSVP, see "Configuring RSVP" in *Configuring MPLS*.

3 Use the following commands to create a second interface to host the LSP that connects the two PE routers:

configure
context local
interface **`if-name`**
ip address **`ip-addr`**

4 Use the following commands to enable MPLS on the LSP interface you created in Step 3:

configure
context local
router mpls
interface **`if-name`**

5 Use the following commands to enable LDP or RSVP on the LSP interface you created in Step 3:

configure
context local
router ldp or *router rsvp*
interface **`if-name`**

When entering the **`interface`** command, replace the ***`if-name`*** argument with the name of the interface you created in Step 3.

Note: For more information about LDP, see *Configuring LDP*. For more information about RSVP, see "Configuring RSVP" in *Configuring MPLS*.

6 To transmit dynamic IPv6 routes, enable the desired IGP on the loopback and LSP interfaces you created in Step 1 and Step 3. For instructions on how to enable your IGP on an interface, see the appropriate document:

– To enable IS-IS on your IPv6 PE interfaces, see *Configuring IS-IS*.

– To enable OSPF on your IPv6 PE interfaces, see *Configuring OSPF*.

– To enable RIP on your IPv6 PE interfaces, see *Configuring RIP*.

For static IPv6 PE configurations, skip this step and proceed to Step 7.

7 Use the following commands to access router BGP configuration mode:

configure
context local
router bgp ***asn***

8 Optional. In router BGP configuration mode, use the **address-family ipv6 unicast** command to access BGP address family configuration mode. Once you are in address family configuration mode, you can optionally use the **redistribute** command to redistribute the desired routes into the BGP routing domain:

address-family ipv6 unicast
redistribute ***options...***

9 In router BGP configuration mode, configure the IP address for the internal BGP neighbor (the BGP PE peer) and access BGP neighbor configuration mode:

neighbor ***ip-addr*** **internal**

Replace ***ip-addr*** with the address of the BGP PE peer.

10 In BGP neighbor configuration mode, use the *update-source* ***if-name*** command to specify the interface used for BGP peering (the loopback interface you created in Step 1). Replace ***if-name*** with the name of the loopback interface.

11 In BGP neighbor configuration mode, use the following commands to specify the use of both IPv4 and IPv6 unicast address prefixes for the BGP routing instance and access BGP neighbor address family configuration mode. You can optionally use the **send label** command to enable the PE router to send MPLS labels with BGP IPv4 or IPv6 routes the peer BGP router:

address-family ipv4 unicast
send label
exit
*address-family ipv6 unicas*t
send label

12 Configure the circuit that transmits the IPv6 packets between the PEs. Use the configuration that is appropriate for the type of circuit you are configuring, as described in *Configuring Circuits*.

13 Use the *bind interface* ***if-name*** command in the appropriate mode to bind the circuit you created in Step 12 to the interface you created in Step 3, as described in *Configuring ATM, Ethernet, and POS Ports*.

14 Perform Steps 1 through 13 on the BGP PE peer to bring up the other end of the connection.

15 Use the *ping ipv6* command to verify that the BGP neighbor is reachable and there is no packet loss. Use the *show ipv6 route* command to verify that your configuration is correct.

2.13 Enabling Soft GRE Tunneling

To enable soft GRE tunneling, perform the task described in Table 14. Enter the command in context configuration mode.

Table 14 Enable Soft GRE Tunneling

Task	Root Command	Notes
Enable soft GRE tunneling on the specified context.	*ip soft-gre*	Using soft GRE tunnels to transport MPLS-encapsulated packets is called BGP/MPLS VPN over GRE, and is used to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. BGP/MPLS VPN over GRE does not require a preconfiguration of the remote GRE endpoint. These endpoints are the BGP next-hop addresses of the VPN routes and are learned dynamically through BGP.

2.14 BGP/MPLS VPN Operations

To manage BGP/MPLS VPN functions, perform the appropriate tasks described in Table 15. Enter the **`show`** commands in any mode; enter the **`clear`** command (in exec mode).

Table 15 BGP/MPLS VPN Operations Tasks

Task	Root Command
Reset BGP IPv4 address connections, or apply new BGP routing policies to connections using VPN address prefixes without dropping the connections.	*clear bgp ipv4 vpn*
Display BGP attribute information for extended communities.	*show bgp attribute extended-community*
Display BGP routes for a specific route target extended community.	*show bgp route ext-community route-target*
Display information for BGP VPN-IPv4 prefix-based routes.	*show bgp route ipv4 vpn*
Display a summary report of BGP VPN-IPv4 routes in the BGP routing tables for all contexts.	*show bgp route ipv4 vpn summary*
Display Open Shortest Path First (OSPF) route information in a VPN context.	*show ospf route vpn*
Display VPN information and VPN redistributed route counts for all OSPF instances, or optionally, for a specific instance in a VPN context.	*show ospf vpn*

Caution!

Risk of dropped connection. A hard reset can impact network connectivity. When using any **`clear bgp`** command, the **`soft`** keyword for inbound only takes effect if the BGP neighbor supports the refresh capability. The **`soft`** keyword for outbound is a local matter, and does not require the capability. To see if a BGP neighbor supports the refresh capability, use the **`show bgp neighbor summary`** command (in exec mode). Specify the **`soft`** keyword if you do not want the BGP neighbor connection dropped. To reduce the risk, only use a hard reset as a last resort.

3 Configuration Examples

The following sections provide BGP/MPLS VPN configuration examples:

3.1 Backbone Connectivity

The backbone connectivity must be configured in the local context.

An IGP, such as OSPF, IS-IS, or LDP, must be enabled on backbone links. By default the loopback interface IP address is used as both the router ID and LDP transport address, so it needs to be reachable. Furthermore, MPLS switching must be enabled on the backbone links.

The following configuration allows two routers to carry BGP routes for VPN-IPv4 unicast addresses. A VPN-IPv4 unicast address is an 8- to 12-byte quantity, beginning with an 8-byte RD and ending with an IPv4 address.

Note: A VPN-IPv4 address family must be configured for the BGP PE peers. IPv4 unicast and multicast address families can be enabled for the same peers if needed.

The configuration for the **PE1** router is:

```
[local]PE1#config
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis ip-backbone
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 1.1.1.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the **PE2** router is:

```
[local]PE2#config
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.2/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.2/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis ip-backbone
[local]PE2(config-isis)#net 49.2222.0010.0100.1002.00
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

3.2 PE-to-CE Route Distribution

PE-to-CE route distribution can be configured using any of the following techniques:

- VPN Using Static Routing
- VPN Using RIP
- VPN Using OSPF
- VPN Using eBGP

Please be aware that you must configure the **`service multiple-context`** command in order to configure a VPN context.

Note: This section does not include the configuration for the backbone connectivity in the local context.

3.2.1 VPN Using Static Routing

The configuration for the **PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.10.1.1/24
[local]PE(config-if)#exit
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute static
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp-af)#exit
[local]PE(config-bgp)#exit
[local]PE(config-ctx)#ip route 192.1.1.0/24 10.10.1.2
[local]PE(config-bgp)#exit
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the **CE** router is:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface loop1 loopback
[local]CE(config-if)#ip address 192.1.1.2/32
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.10.1.2/24
[local]CE(config-ctx)#ip route 0.0.0.0/0 10.10.1.1
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

3.2.2 VPN Using RIP

The configuration for the **PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-if)#rip router CE
[local]PE(config-ctx)#router rip CE
[local]PE(config-rip)#redistribute bgp 100
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute rip CE
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the **CE** router is:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router rip PE
[local]CE(config-rip)#redistribute connected
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

3.2.3 VPN Using OSPF

The configuration for the **PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-ctx)#router ospf 1
[local]PE(config-ospf)#vpn domain-id 5.5.5.5 domain-tag 0x00000001 local-as 100
[local]PE(config-ospf)#area 0.0.0.0
[local]PE(config-ospf)#interface 12/1
[local]PE(config-ospf-interface)#cost 100
[local]PE(config-ospf)#redistribute bgp 100
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp-af)#redistribute ospf
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the **CE** router is:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router ospf 1
[local]CE(config-ospf)#area 0.0.0.0
[local]CE(config-ospf)#interface 2/2
[local]CE(config-ospf-interface)#cost 100
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

3.2.4 VPN Using eBGP

The configuration for the **PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp)#neighbor 10.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 200
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the **CE** router is:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router bgp 200
[local]CE(config-bgp)#address-family ipv4 unicast
[local]CE(config-bgp)#neighbor 10.1.1.1 external
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

3.3 Different BGP/MPLS VPN Topologies

The sections that follow provide configuration examples for typical BGP/MPLS VPNs, local imports, and hub-and-spoke VPNs.

Note: The examples shown in this section all assume eBGP is used for PE-to-CE router connectivity.

3.3.1 Typical BGP/MPLS VPN

The following example configures a typical BGP/MPLS VPN network configuration. Figure 2 shows the network topology for the configuration.

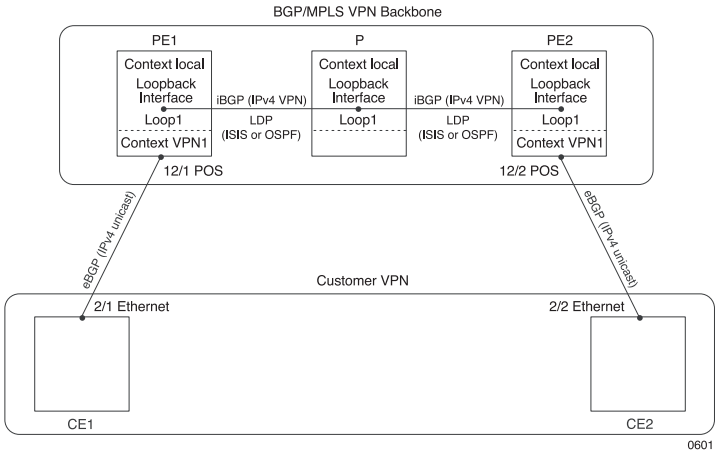

Figure 2 Typical BGP/MPLS VPN

The configuration for the **CE1** router is:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/2
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/2
[local]CE1(config-port)#bind interface 2/2 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the **PE1** router is:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 1.1.1.2/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis ip-backbone
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 1.1.1.1 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:100
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 100:101
[local]PE1(config-bgp-af)#import route-target 100:101
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 VPN1
[local]PE1(config-port)#no shutdown
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the **P** router is:

```
[local]P#config
[local]P(config)#context local
[local]P(config-ctx)#interface loop1 loopback
[local]P(config-if)#ip address 1.1.1.2/32
[local]P(config-if)#isis router isis-backbone
[local]P(config-if)#isis passive-interface
[local]P(config-ctx)#interface backbone1
[local]P(config-if)#ip address 2.2.2.2/24
[local]P(config-if)#isis router isis-backbone
[local]P(config-ctx)#router isis ip-backbone
[local]P(config-isis)#net 49.2222.0010.0100.1002.00
[local]P(config-ctx)#router mpls
[local]P(config-mpls)#interface backbone1
[local]P(config-ctx)#router ldp
[local]P(config-ldp)#interface backbone1
[local]P(config-ctx)#router bgp 100
[local]P(config-bgp)#neighbor 1.1.1.1 internal
[local]P(config-bgp-neighbor)#update-source loop1
[local]P(config-bgp-neighbor)#next-hop-self
[local]P(config-bgp-neighbor)#address-family ipv4 vpn
[local]P(config-bgp-peer-af)#route-reflector-client
[local]P(config-bgp)#neighbor 1.1.1.3 internal
[local]P(config-bgp-neighbor)#update-source loop1
[local]P(config-bgp-neighbor)#next-hop-self
[local]P(config-bgp-neighbor)#address-family ipv4 vpn
[local]P(config-bgp-peer-af)#route-reflector-client
[local]P(config)#port pos 6/1
[local]P(config-port)#bind interface backbone1 local
[local]P(config-port)#no shutdown
[local]P(config-port)#end
```

The configuration for the **PE2** router is:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.3/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis ip-backbone
[local]PE2(config-isis)#net 49.2222.0010.0100.1003.00
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.2 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:100
[local]PE2(config-ctx)#interface 12/2
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 100:101
[local]PE2(config-bgp-af)#import route-target 100:101
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 300
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/2
[local]PE2(config-port)#bind interface 12/2 VPN1
[local]PE2(config-port)#no shutdown
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

The configuration for the **CE2** router is:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 2/2
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.2 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 2/2
[local]CE2(config-port)#bind interface 2/2 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

3.3.2 Local Import

Two CE routers that belong to the same VPN site, and are also connected to the same PE router, are usually configured to be in the same VPN context on the PE router; however, local import can be used if the two CE routers have different import or export policies. The following example configures a local import network configuration. Figure 3 shows the network topology for the configuration.

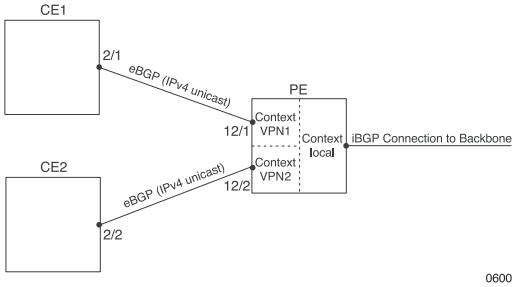

Figure 3 Local Import Network Topology

The configuration for the **CE1** router is:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the **CE2** router is:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 2/2
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 2/2
[local]CE2(config-port)#bind interface 2/2 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

The configuration for the **PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context local
[local]PE(config-ctx)#interface loop1 loopback
[local]PE(config-if)#ip address 1.1.1.1/32
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-if)#isis passive-interface
[local]PE(config-ctx)#interface backbone1
[local]PE(config-if)#ip address 2.2.2.1/24
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-ctx)#router isis ip-backbone
[local]PE(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE(config-ctx)#router mpls
[local]PE(config-mpls)#interface backbone1
[local]PE(config-ctx)#router ldp
[local]PE(config-ldp)#interface backbone1
[local]PE(config-ctx)#router bgp 100
[local]PE(config-bgp)#neighbor 1.1.1.2 internal
[local]PE(config-bgp-neighbor)#update-source loop1
[local]PE(config-bgp-neighbor)#next-hop-self
[local]PE(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE(config)#context VPN1 vpn-rd 1:1
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101 100:102
[local]PE(config-bgp-af)#import route-target 100:101 100:102
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 10.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 200
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
```

```
[local]PE(config)#context vpn1 vpn-rd 1:1
[local]PE(config-ctx)#interface 12/2
[local]PE(config-if)#ip address 11.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101 100:103
[local]PE(config-bgp-af)#import route-target 100:101 100:103
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 11.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 300
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config)#port ethernet 12/2
[local]PE(config-port)#bind interface 12/2 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config)#port pos 6/1
[local]PE(config-port)#bind interface backbone1 local
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

3.3.3 Hub-and-Spoke

Hub-and-Spoke topology allows all spoke sites to send their traffic to a central site location for various different reasons; for example, authentication. The following example configures a Hub-and-Spoke network with two spoke sites and one hub site. Figure 4 shows the network topology for the configuration.

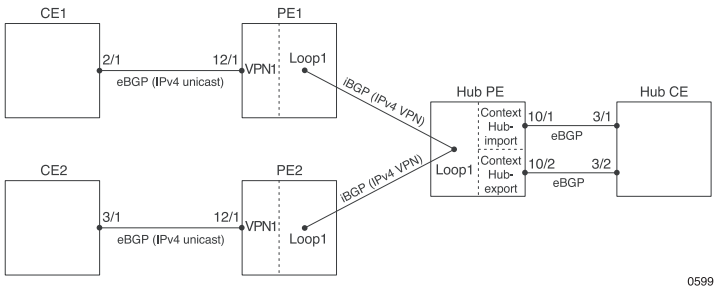

Figure 4 Hub and Spoke Network Topology

The configuration for the **CE1** router is:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the **PE1** router is:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis ip-backbone
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 1.1.1.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

Note: In a Hub-and-Spoke network topology, routes containing the ASN of a hub PE router can be advertised to that same hub PE router as route advertisements are forwarded from one spoke to another. In this scenario, the **asloop-in** command is used to disable the AS_PATH loop detection by accepting a route advertisement which contains the local AS number in AS_PATH. It is configured for the hub CE neighbor in the export context on the hub PE router.

The configuration for the **Hub PE** router is:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context local
[local]PE(config-ctx)#interface loop1 loopback
[local]PE(config-if)#ip address 1.1.1.1/32
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-if)#isis passive-interface
[local]PE(config-ctx)#interface backbone1
[local]PE(config-if)#ip address 2.2.2.2/24
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-ctx)#router isis ip-backbone
[local]PE(config-isis)#net 49.2222.0010.0100.1002.00
[local]PE(config-ctx)#router mpls
[local]PE(config-mpls)#interface backbone1
[local]PE(config-ctx)#router ldp
[local]PE(config-ldp)#interface backbone1
[local]PE(config-ctx)#router bgp 100
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp)#neighbor 1.1.1.2 internal
[local]PE(config-bgp-neighbor)#update-source loop1
[local]PE(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE(config-bgp)#neighbor 1.1.1.3 internal
[local]PE(config-bgp-neighbor)#update-source loop1
[local]PE(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE(config)#context HUB-import vpn-rd 1.1.1.1:1
[local]PE(config-ctx)#interface 10/1
[local]PE(config-if)#ip address 8.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#import route-target 1:1
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 8.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#context HUB-export vpn-rd 1.1.1.1:2
[local]PE(config-ctx)#interface 10/2
[local]PE(config-if)#ip address 9.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 2:2
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 9.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#asloop-in 2
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 10/1
[local]PE(config-port)#bind interface 10/1 HUB-import
[local]PE(config-port)#no shutdown
[local]PE(config)#port ethernet 10/2
[local]PE(config-port)#bind interface 10/2 HUB-export
[local]PE(config-port)#no shutdown
```

```
[local]PE(config)#port pos 6/1
[local]PE(config-port)#bind interface backbone1 local
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

Note: The Hub PE router must have two connections to the Hub CE router, one connection in the import context, and another in the export context. Additionally, the Hub PE router's exporting route target must be configured as an import route target on all spoke PE routers, and export route targets on the spoke PE routers must also be configured as import route targets on the Hub PE router. In this Hub-and-Spoke example, all spoke sites export 1:1 to the hub site, and the hub site exports 2:2 to all spoke sites.

The configuration for the **Hub CE** router is:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 3/1
[local]CE(config-if)#ip address 8.1.1.2/24
[local]CE(config-ctx)#interface 3/2
[local]CE(config-if)#ip address 9.1.1.2/24
[local]CE(config-ctx)#router bgp 400
[local]CE(config-bgp)#address-family ipv4 unicast
[local]CE(config-bgp)#peer-group HUB-pgrp external
[local]CE(config-peergroup)#address-family ipv4 unicast
[local]CE(config-bgp)#neighbor 8.1.1.1 external
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE(config-bgp)#neighbor 9.1.1.1 external
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp)#peer-group HUB-pgrp
[local]CE(config)#port ethernet 3/1
[local]CE(config-port)#bind interface 3/1 local
[local]CE(config-port)#no shutdown
[local]CE(config)#port ethernet 3/2
[local]CE(config-port)#bind interface 3/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

Note: A peer group must be configured for the eBGP peers on the Hub CE router to send back advertisements received from the Hub PE router. By default, routes will not be advertised back to the Hub PE router.

The configuration for the **PE2** router is:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.3/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis ip-backbone
[local]PE2(config-isis)#net 49.2222.0010.0100.1003.00
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#interface 12/1
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp-af)#redistributed connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 300
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/1
[local]PE2(config-port)#bind interface 12/1 VPN1
[local]PE2(config-port)#no shutdown
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

The configuration for the **CE2** router is:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 3/1
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 3/1
[local]CE2(config-port)#bind interface 3/1 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

3.4 Multihop Route Redistribution for an Inter-AS VPN

The following examples show configuring multihop route redistribution using eBGP and LDP.

For an overview of this feature, see Section 1.10 on page 6.

For the tasks to configure it, see Section 2.9 on page 24.

3.4.1 Using eBGP

Figure 5 displays the network topology for a typical eBGP multihop route redistribution configuration.

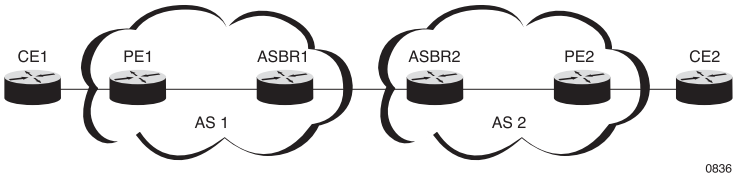

Figure 5 Typical eBGP Multihop Route Redistribution Network Topology

- The PE1 router is configured to have the ASBR1 router as its iBGP neighbor and the PE2 router as its eBGP neighbor.
- The ASBR1 router is configured to have the PE1 router as its iBGP neighbor and the ASBR2 router as its eBGP neighbor.

PE2 and ASBR2 are configured in the same general way, where:

- The PE2 router is configured to have the ASBR2 router as its iBGP neighbor and the PE1 router as its eBGP neighbor.
- The ASBR2 router is configured to have the PE2 router as its iBGP neighbor and the ASBR1 router as its eBGP neighbor.

Configuration for the **PE1** router:

```
[local]PE1#config
[local]PE1(config)#service multiple-contexts
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface 3/10
[local]PE1(config-if)#ip address 30.1.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 5.5.5.5/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface 3/10
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf-area)#interface lo1
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf)#exit
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface 3/10
[local]PE1(config-mpls-if)#exit
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface 3/10
[local]PE1(config-ldp)#exit
[local]PE1(config-ctx)#router bgp 400
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 2.2.2.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#advertisement-interval 1
[local]PE1(config-bgp-neighbor)#ebgp-multihop 10
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#next-hop-unchanged
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#neighbor 4.4.4.4 internal
[local]PE1(config-bgp-neighbor)#advertisement-interval 1
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-peer-af)#send label
[local]PE1(config-bgp-peer-af)#exit
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn1 vpn-rd 2:2
```

```
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 55.55.55.55/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 2:2
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#redistribute static
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#card ge-10-port 3
[local]PE1(config)#port ethernet 3/10
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#bind interface 3/10 local
[local]PE1(config-port)#end
```

Configuration for the **ASBR1** router is:

```
[local]ASBR1#config
[local]ASBR1(config)#service multiple-contexts
[local]ASBR1(config)#context local
[local]ASBR1(config-ctx)#no ip domain-lookup
[local]ASBR1(config-ctx)#interface 3/2
[local]ASBR1(config-if)#ip address 30.1.1.2/24
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#interface 3/4
[local]ASBR1(config-if)#ip address 40.1.1.1/24
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#interface lo1 loopback
[local]ASBR1(config-if)#ip address 4.4.4.4/32
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#router ospf 1
[local]ASBR1(config-ospf)#area 0.0.0.0
[local]ASBR1(config-ospf-area)#interface lo1
[local]ASBR1(config-ospf-if)#exit
[local]ASBR1(config-ospf-area)#interface 3/2
[local]ASBR1(config-ospf-if)#exit
[local]ASBR1(config-ospf-area)#exit
[local]ASBR1(config-ospf)#exit
[local]ASBR1(config-ctx)#router mpls
[local]ASBR1(config-mpls)#interface 3/2
[local]ASBR1(config-mpls-if)#exit
[local]ASBR1(config-mpls)#interface 3/4
[local]ASBR1(config-mpls-if)#exit
[local]ASBR1(config-mpls)#exit
[local]ASBR1(config-ctx)#router ldp
[local]ASBR1(config-ldp)#interface 3/2
[local]ASBR1(config-ldp)#exit
```

```
[local]ASBR1(config-ctx)#router bgp 400
[local]ASBR1(config-bgp)#address-family ipv4 unicast
[local]ASBR1(config-bgp-af)#redistribute ospf 1
[local]ASBR1(config-bgp-af)#exit
[local]ASBR1(config-bgp)#neighbor 5.5.5.5 internal
[local]ASBR1(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR1(config-bgp-neighbor)#update-source lo1
[local]ASBR1(config-bgp-neighbor)#next-hop-self
[local]ASBR1(config-bgp-neighbor)#address-family ipv4 unicast
[local]ASBR1(config-bgp-peer-af)#send label
[local]ASBR1(config-bgp-peer-af)#exit
[local]ASBR1(config-bgp-neighbor)#exit
[local]ASBR1(config-bgp)#neighbor 40.1.1.2 external
[local]ASBR1(config-bgp-neighbor)#remote-as 200
[local]ASBR1(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR1(config-bgp-neighbor)#address-family ipv4 unicast
[local]ASBR1(config-bgp-peer-af)#send label
[local]ASBR1(config-bgp-peer-af)#exit
[local]ASBR1(config-bgp-neighbor)#exit
[local]ASBR1(config-bgp)#exit
[local]ASBR1(config-ctx)#exit
[local]ASBR1(config)#card ge-10-port 3
[local]ASBR1(config)#port ethernet 3/2
[local]ASBR1(config-port)#no shutdown
[local]ASBR1(config-port)#bind interface 3/2 local
[local]ASBR1(config-port)#exit
[local]ASBR1(config)#port ethernet 3/4
[local]ASBR1(config-port)#no shutdown
[local]ASBR1(config-port)#bind interface 3/4 local
[local]ASBR1(config-port)#end
```

3.4.2 Using LDP

Figure 6 displays the network topology for a typical LDP multihop route redistribution configuration, where ASBR2 is a Cisco router and the rest are SmartEdge routers.

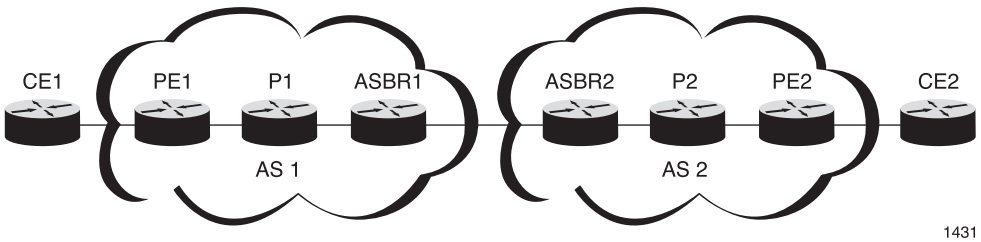

Figure 6 Typical LDP Multihop Route Redistribution Network Topology

The examples in this section show the configuration of PE1, P1 and ASBR1 in AS1 and P2, and PE2 in AS2.

Note: The configuration of ASBR2 is not included in these examples, because it is not a SmartEdge router and the specific configuration of third-party routers is outside the scope of this document. In general, configure ASBR2 to have ASBR1 as its external neighbor and P2 as its internal neighbor, and enable an IGP, MPLS, LDP, and BGP with its address families.

- Configure PE1 to have ASBR1 as its internal neighbor and PE2 as its external neighbor.

  On this router, also configure a VPN connecting with AS2.

- P1 is connected to PE1 and ASBR1 (passes traffic through unchanged).

- Assuming ASBR1 is a SmartEdge router, configure it to have PE1 as its internal neighbor and ASBR2 as its external neighbor.

  Also configure the **`redistribute bgp route-map`** ***`map-name`*** command in `router ldp` configuration mode.

  Also configure route-maps identifying the PE routers as next hops. Under each route-map also configure the **`set ip next-hop prefix-address`** command in route-map configuration mode.

- P2 is connected to PE2 and ASBR2 (passes traffic through unchanged).

- Configure PE2 to have ASBR2 as its internal neighbor and PE1 as its external neighbor.

  On this router, also configure a VPN connecting with AS1.

The configuration of the **PE1** router is:

```
service multiple-contexts
context local
!
 interface lo1 loopback
  ip address 5.5.5.5/32
!
 interface to_p1
  ip address 30.1.1.1/24
 logging console
!
 router ospf 1
  fast-convergence
  area 0.0.0.0
   interface to_p1
   interface lo1
!
 router mpls
  interface to_p1
!
```

```
 router rsvp
  interface to_p1
  interface lo1
!
 router ldp
  interface to_p1
!
 router bgp 400
  address-family ipv4 unicast
  address-family ipv4 vpn
  address-family ipv6 vpn
!
  neighbor 2.2.2.2 external <<<<<<<< PE2
    remote-as 200
    advertisement-interval 1
    ebgp-multihop 10
    update-source lo1
   address-family ipv4 unicast
   address-family ipv4 vpn
      next-hop-unchanged
   address-family ipv6 vpn
      next-hop-unchanged
!
  neighbor 4.4.4.4 internal <<<<<<< ASBR1
    advertisement-interval 1
    update-source lo1
   address-family ipv4 unicast
      send label
!
context vpn1 vpn-rd 2:2
!
 no ip domain-lookup
!
 interface lo1 loopback
  ip address 21.21.21.21/32
  ipv6 address 21::21/128
!
 interface to_CE1
  ip address 11.1.1.1/24
  ipv6 address 11::1/64
 no logging console
!
 router ospf 1
  fast-convergence
  router-id 21.21.21.21
  area 0.0.0.0
   interface lo1
   interface to_CE1
!
 router ospf3 1
  router-id 21.21.21.21
```

```
  area 0.0.0.0
   interface lo1
   interface to_CE1
!
 router bgp vpn
  address-family ipv4 unicast
   export route-target 2:2
   import route-target 2:2
   redistribute connected
   redistribute static
   redistribute ospf 1

  address-family ipv6 unicast
   export route-target 2:2
   import route-target 2:2
   redistribute ospf3 1
```

The configuration of the **P1** router is:

```
context local
!
 no ip domain-lookup
!
 interface lo1 loopback
  ip address 6.6.6.6/32
!
 interface to_pe1
  ip address 30.1.1.2/24
!
 interface to_asbr1
  ip address 40.1.1.2/24
 logging console
!
 router ospf 1
  fast-convergence
  area 0.0.0.0
   interface to_pe1
   interface lo1
   interface to_asbr1
!
 router mpls
  interface to_pe1
  interface to_asbr1
!
 router rsvp
  interface to_pe1
  interface to_asbr1
!
 router ldp
  interface to_pe1
  interface to_asbr1
```

The configuration of the **ASBR1** router is:

```
context local
!
 no ip domain-lookup
!
 interface lo1 loopback
  ip address 4.4.4.4/32
!
 interface to_asbr2
  ip address 50.1.1.1/24
!
 interface to_p1
  ip address 40.1.1.1/24
 logging console
!
 router ospf 3
  fast-convergence
```

```
  area 0.0.0.0
   interface lo1
   interface to_p1
  redistribute bgp 400 route-map r1
!
 ip prefix-list r1
  seq 10 permit 2.2.2.2/32
!
 ip prefix-list s1
  seq 10 permit 5.5.5.5/32
!
 route-map r1 permit 10
  match ip address prefix-list r1
  set ip next-hop prefix-address
!
 route-map s1 permit 20
  match ip address prefix-list s1
  set ip next-hop prefix-address
!
 router mpls
  interface to_asbr2
  interface to_p1
!
 router rsvp
  interface lo1
  interface to_p1
  explicit-route prim1
   next-hop 40.1.1.2
   next-hop 30.1.1.1
  lsp primary1
   egress 5.5.5.5
   source-path prim1
!
 router ldp
  redistribute bgp route-map r1 <<<<<<< Specifies that LDP redistribute the routes
  interface to_p1
!
 router bgp 400
  address-family ipv4 unicast
   redistribute ospf 3 route-map s1
!
  neighbor 5.5.5.5 internal <<<<<<< PE1
    advertisement-interval 1
    update-source lo1
    next-hop-self
   address-family ipv4 unicast
      send label
!
  neighbor 50.1.1.2 external <<<<<ASBR2
    remote-as 200
    advertisement-interval 1
```

```
   address-family ipv4 unicast
      send label
```

The configuration of the **P2** router is:

```
context local
!
 no ip domain-lookup
!
 interface loop loopback
  ip address 7.7.7.7/32
!
 interface to_pe2
  ip address 70.1.1.2/24
!
 interface to_asbr2
  ip address 60.1.1.2/24
 logging console
!
 router ospf 1
  fast-convergence
  area 0.0.0.0
   interface to_asbr2
   interface loop
   interface to_pe2
!
 router mpls
  interface to_asbr2
  interface to_pe2
!
 router ldp
  interface to_asbr2
  interface to_pe2
```

The configuration of the **PE2** router is:

```
service multiple-contexts
context local
!
 no ip domain-lookup
!
 interface lo1 loopback
  ip address 2.2.2.2/32
!
 interface to_p2
  ip address 70.1.1.1/24
 logging console
!
 router ospf 4
  fast-convergence
```

```
  area 0.0.0.0
   interface to_p2
   interface lo1
!
 router mpls
  interface to_p2
!
 router ldp
  interface to_p2
!
 router bgp 200
  address-family ipv4 unicast
  address-family ipv4 vpn
  address-family ipv6 vpn
!
  neighbor 3.3.3.3 internal <<<<<<< ASBR2
    advertisement-interval 1
    update-source lo1
   address-family ipv4 unicast
      send label
!
  neighbor 5.5.5.5 external <<<<<<< PE1
    remote-as 400
    advertisement-interval 1
    ebgp-multihop 10
    update-source lo1
   address-family ipv4 unicast
      send label
   address-family ipv4 vpn
      next-hop-unchanged
   address-family ipv6 vpn
      next-hop-unchanged
!
!
context vpn1 vpn-rd 2:2
!
 no ip domain-lookup
!
 interface lo1 loopback
  ip address 20.20.20.20/32
  ipv6 address 20::20/128
!

interface to_CE2
  ip address 19.1.1.1/24
  ipv6 address 19::1/64
 no logging console
!
 router ospf 1
  fast-convergence
  router-id 20.20.20.20
```

```
  area 0.0.0.0
   interface lo1
   interface to_CE2
!
 router ospf3 1
  router-id 20.20.20.20
  area 0.0.0.0
   interface lo1
   interface to_CE2
!
 router bgp vpn
  address-family ipv4 unicast
   export route-target 2:2
   import route-target 2:2
   redistribute connected
   redistribute static
   redistribute ospf 1
  address-family ipv6 unicast
   export route-target 2:2
   import route-target 2:2
   redistribute ospf3 1
```

3.5 IPsec Tunnels Over MPLS VPN

IPsec tunnels over MPLS VPN provide a way of offering secure corporate access services over an MPLS core such as mobile backbone network. You can configure the IPsec tunnel in its own context, or in the same context as the MPLS VPN. If the IPsec tunnel is configured in a separate context, you must enable inter-context routing between the two contexts.

Figure 7 illustrates how an IPsec tunnel between PE1 and PE2 is configured within a BGP/MPLS VPN. In this example, the VPN terminates its own context, and the IPsec tunnel endpoints, the IKE packets used to negotiate tunnel set up, and the IPsec packets carried by the tunnel, are all configured to terminate in a separate IPsec context. As a result, inter-context routing must be configured between the two contexts.

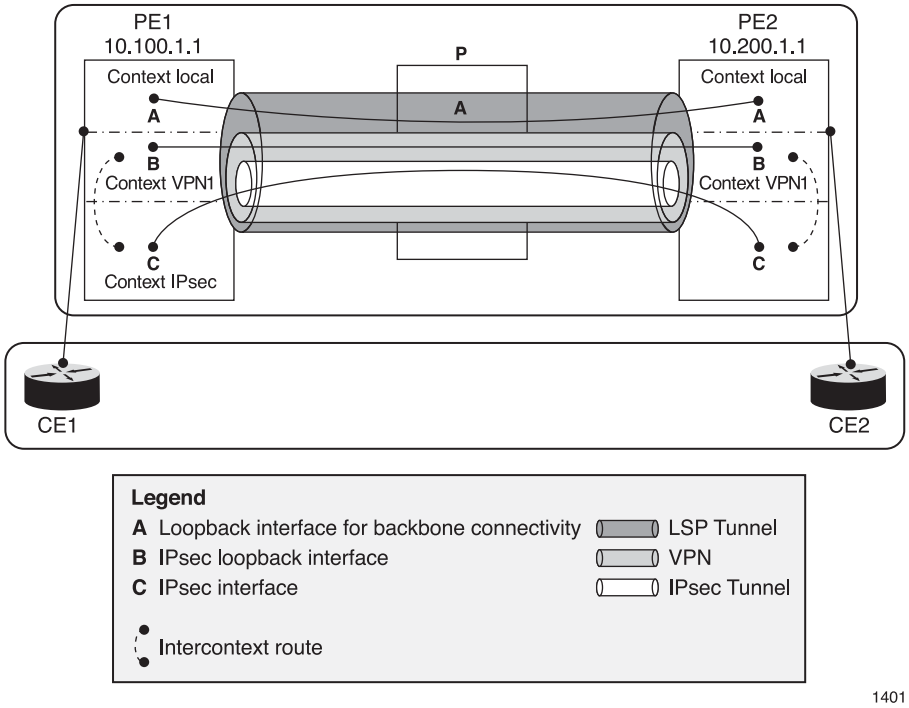

Figure 7 IPsec Tunnels Over BGP/MPLS VPN

3.5.1 IPsec Tunnel Configured in BGP/MPLS VPN Context

When IKE packets and the IPsec packets sent between the IPsec tunnel endpoints are terminated in the same context as the BGP/MPLS VPN, no inter-context routing is required, and no additional changes are required to backbone connectivity, PE-to-CE route distribution, or to the BGP/MPLS VPN, as shown in the other configuration examples in this section. You configure the endpoints of the IPsec tunnel on each PE router, including the IKE and IPsec policies, as described in *IPsec VPN Configuration and Operation Using the SmartEdge OS CLI*, in the same context as the BGP/MPLS context.

3.5.2 IPsec Tunnel Configured in IPsec VPN Context

In this scenario there are two possibilities. Configure the termination of:

- IKE packets in the VPN context and IPsec packets in the MPLS VPN context.
- IKE and IPsec packets in the IPsec context.

These are both variations of how you configure your IPsec tunnels.

When both IKE and IPsec packets are terminated in the IPsec context, inter-context routing must be enabled (in global configuration mode) and two static routes must be defined. One static route is defined in the MPLS VPN context specifying the IP address of the local IPsec loopback interface and the

name of the IPsec context. The other is defined in the IPsec context specifying the IP address of the remote IPsec loopback interface and the name of the MPLS VPN context.

These must be configured at both endpoints, as shown in the following examples.

On the **PE1** router, the MPLS VPN is configured in context `VPN-to-PE2`, and the IPsec tunnel, including IKE and IPsec termination, is configured in context `IPsec-Tun-to-PE2`. The IP address for the local IPsec loopback interface is `34.0.0.1/32`, and for the remote IPsec loopback interface it is `34.1.0.1/32`.

First, enable inter-context routing:

```
[local]PE1#configure
[local]PE1(config)#service inter-context routing
```

Next, create an IP route from the MPLS VPN context to the IPsec context:

```
[local]PE1(config)#context VPN-to-PE2
[local]PE1(config-ctx)#ip route 34.0.0.1/32 context IPsec-Tun-to-PE2
```

Then, create an IP route from the IPsec context to the MPLS VPN context:

```
[local]PE1(config)#context IPsec-Tun-to-PE2
[local]PE1(config-ctx)#ip route 34.1.0.1/32 context VPN-to-PE2
```

On the **PE2** router, the MPLS VPN is configured in context `VPN-to-PE1`, and the IPsec tunnel, including IKE and IPsec termination, is configured in context `IPsec-Tun-to-PE1`. The IP address for the local IPsec loopback interface is `34.1.0.1/32`, and for the remote IPsec loopback interface is `34.0.0.1/32`.

First, enable inter-context routing:

```
[local]PE2#configure
[local]PE2(config)#service inter-context routing
```

Next, create an IP route from the MPLS VPN context to the IPsec context:

```
[local]PE2(config)#context VPN-to-PE1
[local]PE2(config-ctx)#ip route 34.1.0.1/32 context
IPsec-Tun-to-PE1
```

Then, create an IP route from the IPsec context to the MPLS VPN context:

```
[local]PE2(config)#context IPsec-Tun-to-PE1
[local]PE2(config-ctx)#ip route 34.0.0.1/32 context VPN-to-PE1
```

3.6 IPv6 Routes Over an IPv4 MPLS Core

This section provides two examples for tunneling IPv6 over an IPv4 MPLS core:

- IPv6 VPN on Provider Edge router (IPv6 VPN on PE, or 6VPE)
- IPv6 on Provider Edge router (IPV6 on PE, or 6PE)

Note: To configure overlay tunnels to transport IPv6 packets through an MPLS network, see *Configuring Single Circuit Tunnels*.

3.6.1 IPv6 VPN on PE (6VPE)

Figure 8 illustrates a configuration where the PE routers (PE-1 and PE-2) enable two IPV6 networks to exchange routes across a network that has an IPv4 MPLS core.

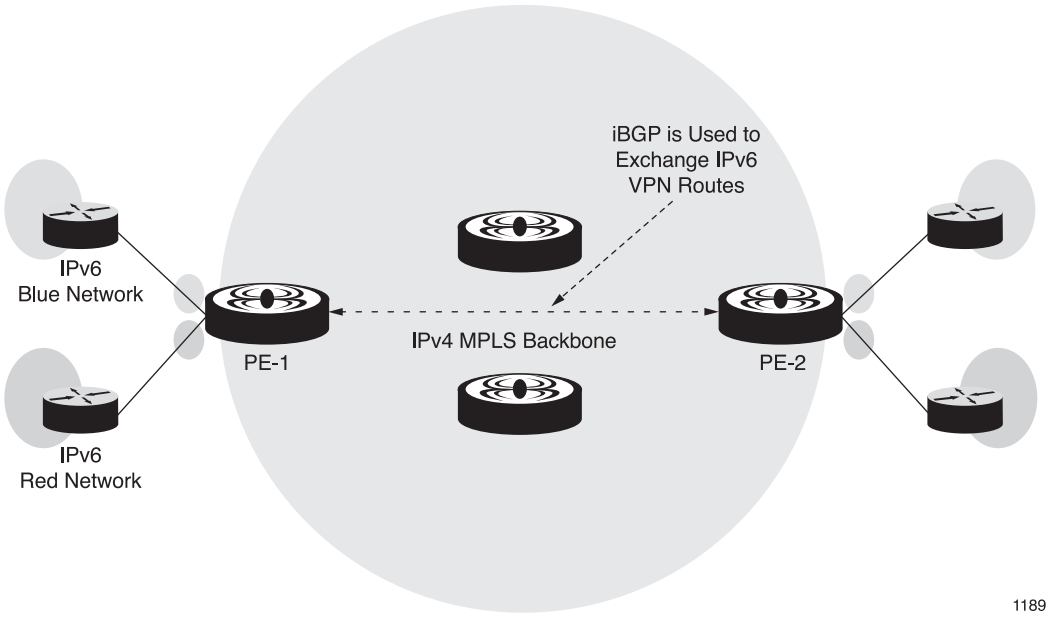

Figure 8 IPv6 Routes Over an MPLS Core (VPN)

The following example enables router PE-1 to exchange routes from the IPv6 networks (called **blue** and **red**) over an IPv4 MPLS network.

First, enable OSPF routing on a the interface called **trunk 1**:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#router ospf 10
[local]PE1(config-ospf)#area 10.10.10.2
[local]PE1(config-ospf-area)#interface trunk1
```

Next, enable LDP on the interface called **trunk1**, so that the interface can be used to exchange Hello messages with neighbors and establish an LSP:

```
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface trunk1
```

Specify the use of standard IPv6 unicast address prefixes for the neighbors in the BGP address family:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 10.10.10.2 internal
[local]PE1(config-bgp-neighbor)#address-family ipv6 vpn
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
```

Enable MPLS on the interface called **trunk1**:

```
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)# interface trunk1
```

Specify the use of IPv6 unicast address prefixes for the BGP routing instance in a VPN context called **blue**. Use the **export route-target** and **import route-target** commands to add the route target extended community with the value **100:100** to the export and import target lists. Use the **redistribute** command to redistribute routes learned from OSPF protocols into the BGP VPN routing instance:

```
[local]PE1(config)#context blue vpn-rd 10.10.10.1:10
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv6 unicast
[local]PE1(config-bgp-af)#export route-target 100:100
[local]PE1(config-bgp-af)#import route-target 100:100
[local]PE1(config-bgp-af)#redistribute ospf 1
```

Enable OSPFv3 on the interface called **blue-ce-pe**. This is the interface that connects the **blue** CE network to PE1:

```
[local]PE1(config-ctx)#router ospf3 1
[local]PE1(config-ospf)#area 10.10.10.2
[local]PE1(config-ospf-area)#interface blue-ce-pe
```

Assign a primary IPv6 address (2001:24:32::1/48) to the interface called **blue-ce**:

```
[local]PE1(config-ctx)#interface blue-ce
[local]PE1(config-if)#ipv6 address 2001:24:32::1/48
```

Specify the use of IPv6 unicast address prefixes for the BGP routing instance in a VPN context called **red**. Use the **export route-target** and **import route-target** commands to add the route target extended community with the value **200:200** to the export and import target lists. Use the **redistribute** command to redistribute routes learned from other protocols into the BGP VPN routing instance:

```
[local]PE1(config)#context red vpn-rd 2.2.2.1:10
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv6 unicast
[local]PE1(config-bgp-af)#import route-target 200:200
[local]PE1(config-bgp-af)#export route-target 200:200
[local]PE1(config-bgp-af)#redistribute ospf 100
```

Enable OSPFv3 on the interface called **red-ce-pe**. This is the interface that connects the **red** CE network to PE1:

```
[local]PE1(config-ctx)#router ospf3 100
[local]PE1(config-ospf)#area 2.2.2.1
[local]PE1(config-ospf-area)#interface red-ce-pe
```

Assign a primary IPv6 address (2001:24:32::1/48) to the interface called **red-ce**:

```
[local]PE1(config-ctx)#interface red-ce
[local]PE1(config-if)#ipv6 address 2001:24:32::1/48
```

Bind the relevant ports to the appropriate interfaces to bring up the connections and enable router CE-1 to transport IPv6 routes over the IPv4 MPLS network:

```
[local]PE1(config)#port ethernet 1/1
[local]PE1(config-port)#description trunk link
[local]PE1(config-port)#bind interface 1 local
[local]PE1(config)#port ethernet 2/2
[local]PE1(config-port)#description link-to-customer-blue-pe
[local]PE1(config-port)#bind interface blue-ce-pe
[local]PE1(config)#port ethernet 3/3
[local]PE1(config-port)#description link-to-customer-red-pe
[local]PE1(config-port)#bind interface red-pe-ce
```

3.6.2 IPv6 on PE (6PE)

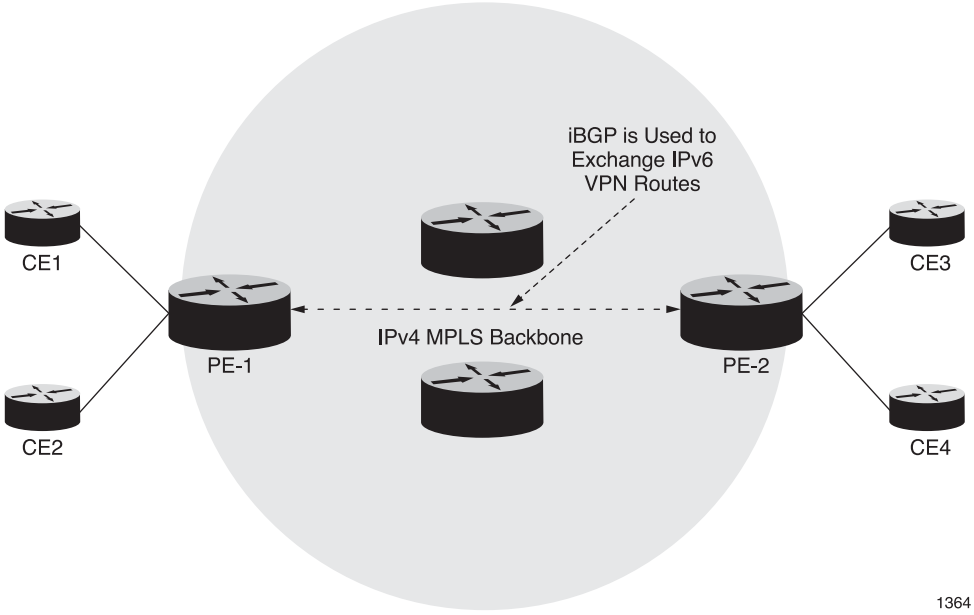

Figure 9 IPv6 Routes Over an MPLS Core (6PE)

This example configures two BGP routers (PE1 and PE2) to transport IPv6 routes through an IPv4 MPLS network over an 802.1q PVC connected to port 3 on slot 5 of PE1 and port 9 on slot 10 of PE2.

On PE1:

Configure a loopback interface to be the BGP endpoint. In this example, the IPv4 gateway is **`3.3.3.3/32`**, and the IPv6 gateway is **`3:3:3::3/128`**:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loopPE loopback
[local]PE1(config-if)#ip address 3.3.3.3/32
[local]PE1(config-if)#ipv6 address 3:3:3::3/128
[local]PE1(config-if)#commit
```

Enable LDP on the loopback interface **`loopPE`**:

```
[local]PE#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface loopPE
[local]PE1(config-ldp)#commit
```

Create a second interface to host the LSP that connects the two PE routers:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config)#interface to-r2
[local]PE1(config)#ip address 12.1.1.1/24
[local]PE1(config)#commit
```

Enable MPLS on the interface **to-r2**:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface to-r2
[local]PE1(config-mpls)#commit
```

Enable LDP on the interface **to-r2**:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface to-r2
[local]PE1(config-ldp)#commit
```

Enable OSPF on the interfaces **loopPE** and **to-r2**:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface loopPE
[local]PE1(config-ospf-area)#interface to-r2
[local]PE1(config-ospf-area)#commit
```

Access router BGP configuration mode, specify the use of both IPv4 and IPv6 unicast address prefixes for the BGP routing instance, and redistribute the desired routes from directly attached networks into the BGP routing domain:

```
[local]PE1#configure
[local]PE1(config)#context local
[local]PE1(config-ctx)#router bgp 1
[local]PE1(config-bgp)#address-family ipv6 unicast
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#
```

Configure the IP address for the internal BGP neighbor (PE2) and configure the **loopPE** interface for BGP peering:

```
[local]PE1(config-bgp)#neighbor 2.2.2.2 internal
[local]PE1(config-bgp-neighbor)#update-source loopPE
```

Specify the use of both IPv4 and IPv6 unicast address prefixes for the BGP routing instance and enable the PE1 router to send MPLS labels with BGP IPv6 routes to the peer router (PE2):

```
[local]PE1(config-bgp-neighbor)#address-fami
ly ipv4 unicast
[local]PE1(config-bgp-peer-af)#exit
[local]PE1(config-bgp-neighbor)#address-family ipv6 unicast
[local]PE1(config-bgp-peer-af)#send label
[local]PE1(config-bgp-peer-af)#commit
```

Bind the interface `to-r2` to a dot1Q PVC on port 5/3. This creates the first endpoint of the LSP between PE1 and PE2:

```
[local]PE1#configure
[local]PE1(config)#port ethernet 5/3
[local]PE1(config-port)#encapsulation dot1q
[local]PE1(config-port)#dot1q pvc 1
[local]PE1(config-dot1q-pvc)# bind interface to-r2 local
[local]PE1(config-dot1q-pvc)#end
```

On PE2:

Configure a loopback interface to be the BGP endpoint. In this example, the IPv4 gateway is **`2.2.2.2/32`**, and the IPv6 gateway is **`2:2:2::2/128`**:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loopPE loopback
[local]PE2(config-if)#ip address 2.2.2.2/32
[local]PE2(config-if)#ipv6 address 2:2:2::2/128
[local]PE2(config-if)#commit
```

Enable LDP on the loopback interface **`loopPE`**:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface loopPE
[local]PE2(config-ldp)#commit
```

Create a second interface to host the LSP that connects the two PE routers. Note that the interface name (**`to-r2`**) and IP address (**`12.1.1.1/24`**) match the configuration for the LSP interface on PE1:

```
[local]PE2#configure
[config]PE2#context local
[local]PE2(config-ctx)#interface to-r2
[local]PE2(config-if)#ip address 12.1.1.1/24
[local]PE2(config-if)#commit
```

Enable MPLS on the interface **to-r2**:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface to-r2
[local]PE2(config-mpls)#commit
```

Enable LDP on the interface **to-r2**:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface to-r2
[local]PE2(config-ldp)#commit
```

Enable OSPF on the interfaces **loopPE** and **to-r2**:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#router ospf 1
[local]PE2(config-ospf)#area 0.0.0.0
[local]PE2(config-ospf-area)#interface loopPE
[local]PE2(config-ospf-if)#interface to-r2
[local]PE2(config-ospf-if)#commit
```

Access router BGP configuration mode, specify the use of both IPv4 and IPv6 unicast address prefixes for the BGP routing instance, and redistribute routes from directly attached networks into the BGP routing domain:

```
[local]PE2#configure
[local]PE2(config)#context local
[local]PE2(config-ctx)#router bgp 1
[local]PE2(config-bgp)#address-family ipv6 unicast
[local]PE2(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#
```

Configure the IP address for the internal BGP neighbor (PE1) and access BGP neighbor configuration mode. Specify the interface used for BGP peering:

```
[local]PE2(config-bgp)#neighbor 3.3.3.3 internal
[local]PE2(config-bgp-neighbor)#update-source loopPE
```

Specify the use of standard IPv4 and IPv6 unicast address prefixes for the BGP routing instance and enable the PE2 router to send MPLS labels with BGP IPv4 or IPv6 routes to the peer (PE1):

```
[local]PE2(config-bgp-neighbor)#address-fami
ly ipv4 unicast
[local]PE2(config-bgp-peer-af)#exit
[local]PE2(config-bgp-neighbor)#address-family ipv6 unicast
[local]PE2(config-bgp-peer-af)#send label
[local]PE2(config-bgp-peer-af)#commit
```

Bind the interface `to-r2` to a dot1Q PVC on port **`9/10`**. This activates the connection between PE1 and PE2:

```
[local]PE2#configure
[local]PE2(config)#port ethernet 9/10
[local]PE2(config-port)#encapsulation dot1q
[local]PE2(config-port)#dot1q pvc 1
[local]PE2(config-dot1q-pvc)#bind interface to-r2 local
[local]PE2(config-dot1q-pvc)#end
```

Use the **`ping ipv6`** command to verify your connection and ensure PE1 is reachable. If you have successfully configured 6PE between two PEs, there will be no (0%) packet loss:

```
[local]PE2#ping ipv6 2:2:2::2

PING6 2:2:2::2 : 8 data bytes
timeout is 1 second, source 2:2:2::2
!!!!!

--- 2:2:2::2 ping6 statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 0.816/1.012/1.389/0.197 ms
```

Use the **`show ipv6 route`** command to verify your configuration on both PE routers:

```
[local]PE1#show ipv6 route
Codes: C - connected, S - static, S dv - dvsr, R - RIP, e B - EBGP, i B - IBGP
       O   - OSPF, O3  - OSPFv3, IA - OSPF(v3) inter-area,
       N1  - OSPF(v3) NSSA external type 1, N2  - OSPF(v3) NSSA external type 2
       E1  - OSPF(v3) external type 1, E2  - OSPF(v3) external type 2
       i   - IS-IS, L1 - IS-IS level-1,  L2  - IS-IS level-2, N - NAT
       IPH - IP Host, SUB A - Subscriber address, SUB S - Subscriber static
       SUB P - AAA downloaded aggregate subscriber routes
       SUB N - Subscriber ND, SUB D - Subscriber DHCP-PD
       M F - Mobile Sub Foreign Agent, M H - Mobile Sub Home Agent
       M G - Mobile Sub GTP
       A - Derived Default, MeH - Media Nexthop
       >   - Active Route, * - LSP

Type    Network             Next Hop        Dist  Metric    UpTime  Interface

> i B   2:2:2::2/128        2.2.2.2          200       0     1d05h
> C     3:3:3::3/128        3:3:3::3           0       0     1d05h  loopPE
[local]jazz#

[local]PE1#show ipv6 route 3:3:3::3/128

    Best match Routing entry for 3:3:3::3/128 is 3:3:3::3/128 , version 3
    Route Uptime 1d05h
    Paths: total 1, best path count 1
        Route redistributed to bgp 1

    Route has been downloaded to following slots
     01/0, 04/0, 05/0, 06/0,X-EP-NAME, 12/0, 14/0

    Path information :

      Active path :
      Known via connected, distance 0, metric 0,
      Tag 0, Next-hop 3:3:3::3, NH-ID 0x31D00004, Interface loopPE
      Circuit 255/2:1:1/1/1/31
```

3.7 GRE over MPLS

GRE over MPLS provides a way to establish a GRE tunnel over an MPLS LSP, allowing you to run applications, such as multicast, over the GRE tunnel. The following example configures BGP/MPLS VPNs on routers **PE1** and **PE2**. The GRE tunnel, **tun1**, is created over MPLS by specifying the GRE peer relationship on both ends of the tunnel, which are represented by routers **PE1** and **PE2**. For each GRE peer relationship specified, the remote IP address must be an IP address in the remote VPN context.

The configuration for the **PE1** router is:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 2.2.2.2/32
[local]PE1(config-ctx)#interface toP
[local]PE1(config-if)#ip address 10.1.1.2/30
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface lo1
[local]PE1(config-ospf-interface)#passive
[local]PE1(config-ospf-area)#interface toP
[local]PE1(config-ospf-area)#exit
[local]PE1(config-ospf)#exit
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#no propagate ttl ip-to-mpls
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router rsvp
[local]PE1(config-rsvp)#interface toP
[local]PE1(config-rsvp-if)#lsp lsp1
[local]PE1(config-rsvp-lsp)#ingress 2.2.2.2
[local]PE1(config-rsvp-lsp)#egress 3.3.3.3
[local]PE1(config-rsvp-lsp)#exit
[local]PE1(config-rsvp-if)#exit
[local]PE1(config-rsvp)#exit
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 3.3.3.3 internal
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn1 vpn-rd 2.2.2.2:1
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#interface gre1
[local]PE1(config-if)#ip address 30.1.1.1/30
[local]PE1(config-ctx)#interface toCE1
[local]PE1(config-if)#ip address 100.1.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 100:1
[local]PE1(config-bgp-af)#import route-target 100:1
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#tunnel gre tun1
[local]PE1(config-tunnel)#peer-end-point local 100.2.1.1 remote 100.1.1.1 context local
[local]PE1(config-tunnel)#end
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the **PE2** router is:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop loopback
[local]PE2(config-if)#ip address 3.3.3.3/32
[local]PE2(config-ctx)#interface toP
[local]PE2(config-if)#ip address 10.1.2.2/30
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ospf 1
[local]PE2(config-ospf)#area 0.0.0.0
[local]PE2(config-ospf-area)#interface loop
[local]PE2(config-ospf-interface)#passive
[local]PE2(config-ospf-area)#interface toP
[local]PE2(config-ospf-area)#exit
[local]PE2(config-ospf)#exit
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#no propagate ttl ip-to-mpls
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router rsvp
[local]PE2(config-rsvp)#interface toP
[local]PE2(config-rsvp-if)#lsp lsp1 signaled
[local]PE2(config-rsvp-lsp)#ingress 3.3.3.3
[local]PE2(config-rsvp-lsp)#egress 2.2.2.2
[local]PE2(config-rsvp-lsp)#exit
[local]PE2(config-rsvp-if)#exit
[local]PE2(config-rsvp)#exit
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 2.2.2.2 internal
[local]PE2(config-bgp-neighbor)#update-source loop
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#context vpn1 vpn-rd 3.3.3.3:1
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface gre1
[local]PE2(config-if)#ip address 30.1.1.2/30
[local]PE2(config-ctx)#interface toCE1
[local]PE2(config-if)#ip address 100.2.1.1/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 100:1
[local]PE2(config-bgp-af)#import route-target 100:1
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#tunnel gre tun1
[local]PE2(config-tunnel)#peer-end-point local 100.2.1.1 remote 100.1.1.1 context local
[local]PE2(config-tunnel)#end
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

3.8 BGP/MPLS VPN over GRE

BGP/MPLS VPN over GRE provides a way to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. For BGP/MPLS VPN over GRE to work, the PE routers must know how to handle GRE and label packets, and they must have MPLS enabled on the interface that receives GRE and label packets from the backbone.

Figure 10 shows the network topology for this BGP/MPLS VPN over GRE configuration example where both PE routers are within the same AS.

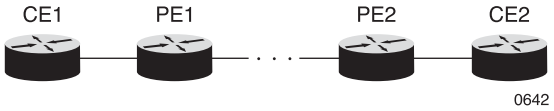

Figure 10 Basic BGP/MPLS VPN over GRE Network Topology

The configuration for the **PE1** router is:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop loopback
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface to_backbone
[local]PE1(config-if)#ip address 15.3.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface t0
[local]PE1(config-if)#ip address 50.50.51.2/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface to_backbone
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 2.2.2.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#ip soft-gre source 1.1.1.1
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn0 vpn-rd 100:200
[local]PE1(config-ctx)#interface to_ce1
[local]PE1(config-if)#ip address 10.31.0.2/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 4134:4000
[local]PE1(config-bgp-af)#import route-target 4134:4000
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 10.31.0.1 external
[local]PE1(config-bgp-neighbor)#remote-as 4001
[local]PE1(config-bgp-neighbor)#update-source to_ce1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
```

The configuration for the **PE2** router is:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop loopback
[local]PE2(config-if)#ip address 2.2.2.2/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface to_backbone
[local]PE2(config-if)#ip address 16.3.1.1/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface to_backbone
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#ip soft-gre source 2.2.2.2
[local]PE2(config-ctx)#exit
[local]PE2(config)#context vpn0 vpn-rd 100:300
[local]PE2(config-ctx)#interface to_ce2
[local]PE2(config-if)#ip address 10.11.0.2/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 4134:4000
[local]PE2(config-bgp-af)#import route-target 4134:4000
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#neighbor 10.11.0.1 external
[local]PE2(config-bgp-neighbor)#remote-as 4001
[local]PE2(config-bgp-neighbor)#update-source to_ce2
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
```

If BGP/MPLS VPN service spans multiple autonomous systems, there are two ways to exchange VPN routes between the VPN sites across the autonomous systems:

1. Configure eBGP peering between the ASBRs, enable a VPN address family between the PE router and ASBR, and enable a VPN address family between the ASBRs. That is, within each AS, both IPv4 unicast and VPN routes are exchanged, and ASBRs are used to exchange VPN routes for interdomain routing.

2. Configure multihop eBGP peering between the PE routers, and enable VPN address family between the PE routers to exchange VPN routes. The ASBR and PE routers on the backbone exchange only IPv4 unicast routes.

For both methods, the next-hop-unchanged option must be configured on the ASBRs in the VPN address family for the peer that is peering with the other ASBR to preserve the (next-hop, label) pair.

3.9 BGP Commands for BGP/MPLS VPN

Some BGP/MPLS VPN-related commands should only be used for specific situations. The following sections provide configuration examples that illustrate the correct use of the VPN-related commands, **`asloop-in`**, **`as-override`**, and **`route-origin`**:

3.9.1 Using the asloop-in Command

The **`asloop-in`** command is used to disable the AS_PATH loop detection by accepting a route advertisement which contains the local AS number in AS_PATH.

This command is useful for Hub-and-Spoke network topologies where routes containing a hub PE router's ASN can be advertised to the same hub PE router as route advertisements are forwarded from one spoke to another.

This command should be configured for the hub CE neighbor in the export context on the hub PE router.

The configuration for the hub **PE** router is:

```
[local]PE#config
[local]PE(config)#context HUB-export vpn-rd 1.1.1.1:2
[local]PE(config-ctx)#interface 10/2
[local]PE(config-if)#ip address 9.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 2:2
[local]PE(config-bgp)#neighbor 9.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#asloop-in 2
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 10/2
[local]PE(config-port)#bind interface 10/2 HUB-export
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

3.9.2 Using the as-override Command

The **`as-override`** command is used to replace all occurrences of the peer's ASN in the AS_PATH attribute with the local ASN when advertising the route to the peer.

Assuming that both VPN sites for the **CE1** and **CE2** routers use the ASN **200**, the **as-override** command must be configured for the CE peers on the PE routers before the route advertisements can be accepted by the CE routers at both sites.

Note: Backbone connectivity in the local context is not shown in the following example.

The configuration for the **CE1** router is:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-neighor)#remote-as 100
[local]CE1(configneighor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the **PE1** router is:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#as-override
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 VPN1
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the **PE2** router is:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#interface 12/1
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 200
[local]PE2(config-bgp-neighbor)#as-override
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/1
[local]PE2(config-port)#bind interface 12/1 VPN1
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

The configuration for the **CE2** router is:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 3/1
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 200
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 3/1
[local]CE2(config-port)#bind interface 3/1 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

3.9.3 Using the route-origin Command

In the case of multiple sites sharing the same ASN, using an ASN alone is no longer adequate for AS loop detection. To prevent the readvertisement of routes back to the originating site, use the **route-origin** command to identify the site from where the routes originated.

The configuration for the **PE1** router is:

```
[local]PE1#config
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#route-origin 100:300
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#as-override
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#end
```

The configuration for the **PE2** router is:

```
[local]PE2#config
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#route-origin 100:400
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 200
[local]PE2(config-bgp-neighbor)#as-override
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
```