

Chapter 20

Configuration Examples for Interprovider and Carrier-of-Carriers VPNs

This chapter contains examples that illustrate how to configure interprovider and carrier-of-carriers virtual private networks (VPNs). It includes the following sections:

Example Terminology on page 410

Interprovider VPN Examples on page 411

Carrier-of-Carriers VPN Examples on page 422

Multiple Instances for LDP and Carrier-of-Carriers VPNs on page 447

Example Terminology

The following terminology is used in these examples and is specific to Juniper Networks:

`bgp.l3vpn.0`—The table on the provider edge (PE) router in which the VPN-IPv4 routes that are received from another PE router are stored. Incoming routes are checked against the `vrf-import` statements from all the VPNs configured on the PE router. If there is a match, the VPN-Internet Protocol version 4 (IPv4) route is added to the `bgp.l3vpn.0` table. To view the `bgp.l3vpn.0` table, issue the `show route table bgp.l3vpn.0` command.

`routing-instance-name.inet.0`—The routing table for a specific routing instance. For example, a routing instance called `VPN-A` has a routing table called `VPN-A.inet.0`. Routes are added to this table in the following ways:

They are sent from a customer edge (CE) router configured within the `VPN-A` routing instance.

They are advertised from a remote PE router that passes the `vrf-import` policy configured within `VPN-A` (to view the route, run the `show route` command). IPv4 (not VPN-IPv4) routes are stored in this table.

`vrf-import policy-name`—An import policy configured on a particular routing instance on a PE router. This policy is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes learned from another PE router or a route reflector.

`vrf-export policy-name`—An export policy configured on a particular routing instance on a PE router. It is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes (originally learned from locally connected CE routers as IPv4 routes), which are advertised to another PE router or route reflector.

MP-EBGP—The multiprotocol external Border Gateway Protocol (MP-EBGP) mechanism is used to export VPN-IPv4 routes across an autonomous system (AS) boundary. To apply this mechanism, use the `labeled-unicast` statement at the `[edit protocols bgp group group-name family inet]` hierarchy level.

Interprovider VPN Examples

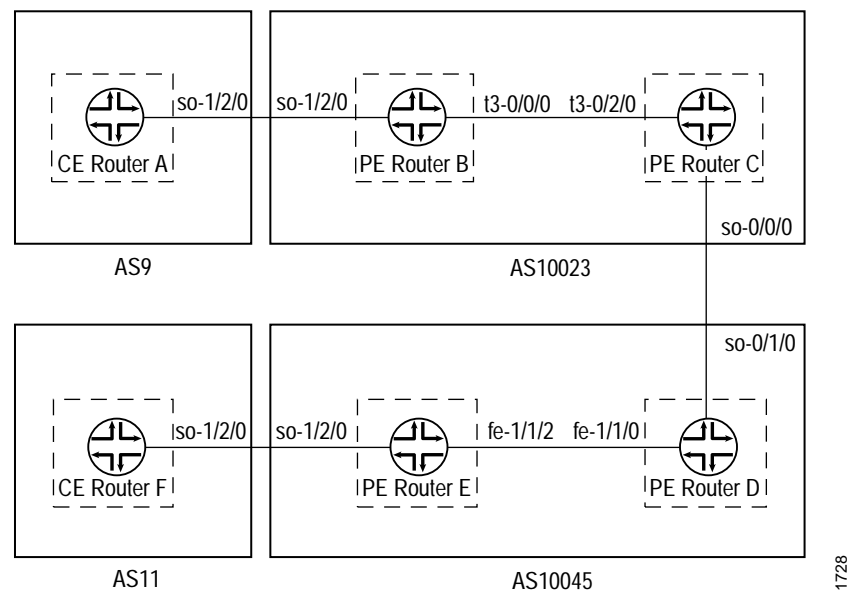
The following examples illustrate how to configure interprovider VPNs:

Interprovider VPN Example—MP-EBGP Between ISP Peer Routers on page 411

Interprovider VPN Example—Multihop MP-EBGP on page 418

Figure 46 illustrates the network topology used in both VPN examples.

Figure 46: Network Topology of Interprovider VPN Examples



Interprovider VPN Example—MP-EBGP Between ISP Peer Routers

In this example, all routes learned from the CE routers are sent over both service provider networks as VPN-IPv4 routes. The routes are initially learned by the PE routers (Router B and Router E) from the CE routers (Router A and Router F) and are announced by the PE routers to the AS border routers (Router C and Router D). The AS border routers are then configured with an MP-EBGP session, enabling them to pass the VPN-IPv4 routes with each other. When an AS border router—Router C for example—learns VPN-IPv4 routes from an internal Border Gateway Protocol (IBGP) PE, the following occurs:

1. Router C sets itself as the next hop for the route and creates a label for that route.
2. Router C advertises the VPN-IPv4 route to PE Router D in AS 10045.
3. Router D sets the next hop to itself, creates another label, and then forwards the label and the route to its IBGP PE router (Router E).

This example has scaling limitations because of restrictions on the number of labels each PE router needs to allocate at the AS border.

Configuration for Router A

Configure a family inet EBGP session with Router B and export the direct routes:

```
[edit]
protocols {
  bgp {
    group to-provider {
      export attached;
      peer-as 10023;
      neighbor 192.168.198.2;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Router A is configured as a CE router (using the routing-instances statement) in the configuration for Router B. Because they exchange VPN-IPv4 routes, Router D and Router C are configured as PE routers.

Configure Router B:

```
[edit]
protocols {
  rsvp {
    interface t3-0/0/0.0;
  }
  mpls {
    label-switched-path to-routerC {
      to 10.255.14.171;
      description "to-routerC for use with VPNs";
    }
    interface t3-0/0/0.0;
    interface so-1/2/0.0;
  }
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.175;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.171;
    }
  }
}
```

```

ospf {
  traffic-engineering;
  reference-bandwidth 4g;
  area 0.0.0.0 {
    interface t3-0/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
}
}
routing-instances {
  vpn {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpn-import;
    vrf-export vpn-export;
    protocols {
      bgp {
        group to-ce {
          peer-as 9;
          neighbor 192.168.198.1;
        }
      }
    }
  }
}
}
policy-options {
  policy-statement vpn-import {
    term 1 {
      from {
        protocol bgp;
        community vpn-comm;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement vpn-export {
    term 1 {
      from protocol bgp;
      then {
        community add vpn-comm;
        accept;
      }
    }
    term 2 {
      then reject;
    }
  }
  community vpn-comm members target:100:1001;
}
}

```

Configuration for Router C

In the BGP protocol configuration for Router C, include the keep all statement. When this statement is included, BGP must store every route learned through BGP. Configure two BGP sessions (configure family inet-vpn on both sessions):

IBGP session to Router D (group to-ibgp in this example)

EBGP session to Router B (group to-ebgp-pe in this example)

Interface t3-0/2/0 is added at the [edit protocols mpls] hierarchy level, allowing BGP to announce routes with labels over the EBGP session.

Configure Router C:

```
[edit]
protocols {
  rsvp {
    interface t3-0/2/0.0;
  }
  mpls {
    label-switched-path to-routerB {
      to 10.255.14.175;
      description "to-routerB for use with vpns";
    }
    interface t3-0/2/0.0;
    interface so-0/0/0.0;
  }
  bgp {
    keep all;
    group to-ibgp {
      type internal;
      local-address 10.255.14.171;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.175;
    }
    group to-ebgp-pe {
      type external;
      family inet-vpn {
        unicast;
      }
      neighbor 192.168.197.22 {
        peer-as 10045;
      }
    }
  }
}
```

```

ospf {
  traffic-engineering;
  reference-bandwidth 4g;
  area 0.0.0.0 {
    interface t3-0/2/0.0;
    interface lo0.0 {
      passive;
    }
  }
}
}

```

Configuration for Router D

The configuration for Router D is almost identical to that of Router C:

```

[edit]
protocols {
  rsvp {
    interface fe-1/1/0.0;
  }
  mpls {
    label-switched-path to-E {
      to 10.255.14.177;
      description "to-routerE for vpna";
    }
    interface fe-1/1/0.0;
    interface so-0/1/0.0;
  }
  bgp {
    keep all;
    group to-ibgp-pe {
      type internal;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.177;
    }
    group to-ebgp-pe {
      type external;
      family inet-vpn {
        unicast;
      }
      peer-as 10023;
      neighbor 192.168.197.21;
    }
  }
}

```

```

ospf {
  traffic-engineering;
  reference-bandwidth 4g;
  area 0.0.0.0 {
    interface fe-1/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
}

```

Configuration for Router E

The configuration for Router E is very similar to the configuration for Router B:

```

[edit]
protocols {
  rsvp {
    interface fe-1/1/2.0;
  }
  mpls {
    label-switched-path to-routerD {
      to 10.255.14.173;
      description "to-routerD for use with VPNa";
    }
    interface fe-1/1/2.0;
    interface so-1/2/0.0;
  }
  bgp {
    group to-ibgp-pe {
      type internal;
      local-address 10.255.14.177;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.173;
    }
  }
  ospf {
    traffic-engineering;
    reference-bandwidth 4g;
    area 0.0.0.0 {
      interface fe-1/1/2.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}

```

```

routing-instances {
  vpna {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 10.255.14.177:11;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      bgp {
        group to-routerF-ce {
          neighbor 192.168.198.14 {
            peer-as 11;
          }
        }
      }
    }
  }
}
policy-options {
  policy-statement vpna-import {
    term 1 {
      from {
        protocol bgp;
        community vpna-comm;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement vpna-export {
    term 1 {
      from protocol bgp;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term 2 {
      then reject;
    }
  }
  community vpna-comm members target:100:1001;
}

```

Configuration for Router F

Configure Router F as a CE router; the configuration is similar to that for Router A:

```
[edit]
protocols {
  bgp {
    group to-provider {
      type external;
      export attached;
      neighbor 192.168.198.13 {
        peer-as 10045;
      }
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Interprovider VPN Example—Multihop MP-EBGP

In this example, labeled IPv4 (not VPN-IPv4), routes are exchanged by the AS border routers (Router C and Router D) to provide Multiprotocol Label Switching (MPLS) connectivity between the PE routers. Only routes internal to the service provider networks should be announced between Router C and Router D. Configure this by including the family inet labeled-unicast statement in the IBGP and EBGP configuration on the PE routers. When you set family inet labeled-unicast, the local router announces internal routes from inet.0 in the following manner:

If a label exists for the route, the local router creates a label, performs a swap, and announces the route from inet.0 with the label.

If a label does not exist for the route, the local router creates a label, performs a pop, and announces the route from inet.0 with the label.

Routes learned from the labeled-unicast session are placed into the inet.0 routing table.

In addition, you configure a multihop MP-EBGP session between the end PE routers (Router B and Router E). This additional MP-EBGP session allows the announcement of VPN-IPv4 routes, and allows you to maintain VPN connectivity while keeping VPN-IPv4 routes out of the core of the network.



NOTE: The configurations for the routers in the section “Interprovider VPN Example—Multihop MP-EBGP” on page 418 are similar to those for the routers in the section “Interprovider VPN Example—MP-EBGP Between ISP Peer Routers” on page 411. In the sections that follow, only the differences in these configurations are shown. The configurations for Router A and Router F are the same, so they are not repeated.

Configuration for Router B

In group to-ibgp, include the family inet labeled-unicast statement to pass labeled IPv4 routes, and configure an EBGP multihop session to pass VPN-IPv4 routes:

```
[edit]
protocols {
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.175;
      family inet {
        labeled-unicast {
          resolve-vpn;
        }
      }
      neighbor 10.255.14.171;
    }
    group to-remote-pe {
      multihop {
        ttl 10;
      }
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.177 {
        peer-as 10045;
      }
    }
  }
}
```

Configuration for Router C

Configure Router C:

```
[edit]
protocols {
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.171;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.175;
    }
    group to-ebgp-pe {
      type external;
      family inet {
        labeled-unicast;
      }
      export internal;
      neighbor 192.168.197.22 {
        peer-as 10045;
      }
    }
  }
}
```

```

policy-options {
  policy-statement internal {
    term 1 {
      from protocol [ ospf direct ];
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

Configuration for Router D

Configure Router D:

```

[edit]
protocols {
  bgp {
    group to-ibgp-pe {
      type internal;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.177;
    }
    group to-ebgp-pe {
      type external;
      family inet {
        labeled-unicast;
      }
      export internal;
      peer-as 10023;
      neighbor 192.168.197.21;
    }
  }
}
policy-options {
  policy-statement internal {
    term 1 {
      from protocol [ direct ospf ];
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

Configuration for Router E

Configure Router E:

```
[edit]
protocols {
  bgp {
    group to-ibgp-pe {
      type internal;
      local-address 10.255.14.177;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.173;
    }
    group to-remote-pe {
      multihop {
        ttl 10;
      }
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.175 {
        peer-as 10023;
      }
    }
  }
}
```

Carrier-of-Carriers VPN Examples

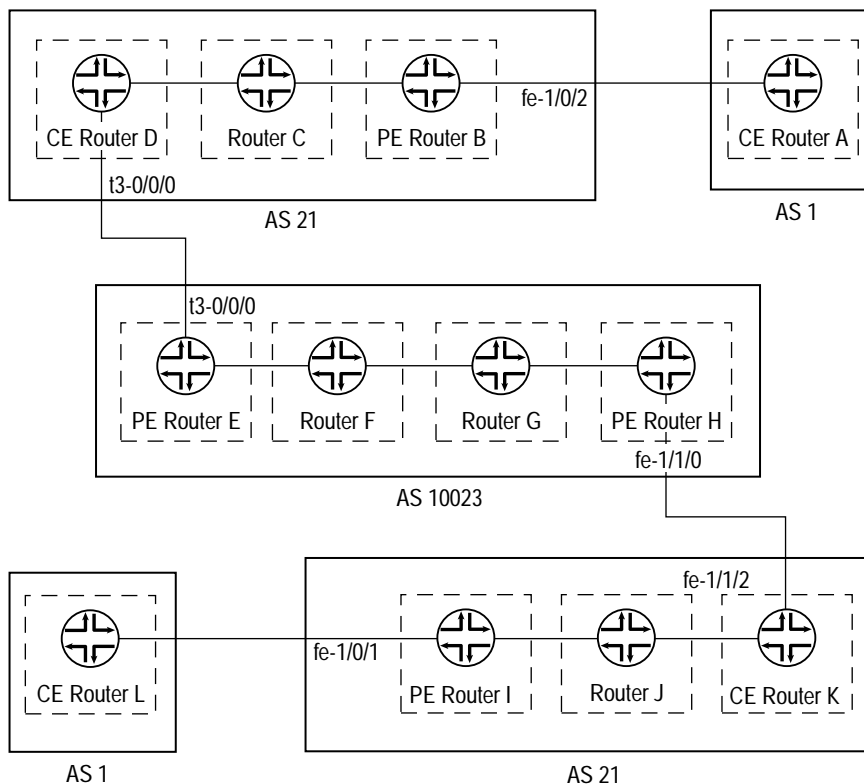
A carrier-of-carriers service allows an Internet service provider (ISP) to connect to a transparent outsourced backbone at multiple locations. There are two variations of this example:

Carrier-of-Carriers VPN Example—Customer Provides Internet Service on page 423

Carrier-of-Carriers VPN Example—Customer Provides VPN Service on page 434

Figure 47 shows the network topology in both carrier-of-carriers examples.

Figure 47: Carrier-of-Carriers VPN Example Network Topology



1729

Carrier-of-Carriers VPN Example—Customer Provides Internet Service

In this example, the carrier customer is not required to configure MPLS and Label Distribution Protocol (LDP) on its network. However, the carrier provider must configure MPLS and LDP on its network.

Configuration for Router A

In this example, Router A represents an end customer. You configure this router as a CE device.

```
[edit]
protocols {
  bgp {
    group to-routerB {
      export attached;
      peer-as 21;
      neighbor 192.168.197.169;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Router B can act as the gateway router, responsible for aggregating end customers and connecting them to the network. If a full-mesh IBGP session is configured, you can use route reflectors.

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.181;
      neighbor 10.255.14.176;
      neighbor 10.255.14.178;
      neighbor 10.255.14.177;
    }
    group to-vpn-blue {
      peer-as 1;
      neighbor 192.168.197.170;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/3.0;
      interface fe-1/0/2.0 {
        passive;
      }
    }
  }
}
```

Configuration for Router C

Configure Router C:

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.176;
      neighbor 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.177;
      neighbor 10.255.14.178;
      neighbor 10.255.14.181;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/3.0;
      interface fe-0/3/0.0;
    }
  }
}
```

Configuration for Router D

Router D is the CE router with respect to AS 10023. In a carrier-of-carriers VPN, the CE router must be able to send labels to the carrier provider; this is done with the labeled-unicast statement in group to-isp-red.

```
[edit]
protocols {
  mpls {
    interface t3-0/0/0.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.175;
      neighbor 10.255.14.179;
      neighbor 10.255.14.176;
      neighbor 10.255.14.177;
      neighbor 10.255.14.178;
      neighbor 10.255.14.181;
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.13 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/0.0;
      interface t3-0/0/0.0 {
        passive;
      }
    }
  }
}
policy options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

Configuration for Router E

This configuration sets up the inet-vpn IBGP session with Router H and the PE router portion of the VPN with Router D. Because Router D is required to send labels in this example, configure the BGP session with the labeled-unicast statement within the VPN routing and forwarding (VRF) table.

```
[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-0/1/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.173;
    }
  }
  isis {
    interface at-0/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface at-0/1/0.0;
  }
}
routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.14 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
}
```

```

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router F

Configure Router F to act as a label-swapping router:

```

[edit]
protocols {
  isis {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
  }
}

```

Configuration for Router G

Configure Router G to act as a label-swapping router:

```
[edit]
protocols {
  isis {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
  }
}
```

Configuration for Router H

Router H acts as the PE router for AS 10023. The configuration that follows is similar to that for Router F:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/0.0;
    interface so-1/0/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.173;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.171;
    }
  }
  isis {
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-1/0/0.0;
  }
}
```

```

routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.173:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.94 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router I

Configure Router I to connect to the basic Internet service customer (Router L):

```
[edit]
protocols {
  mpls {
    interface fe-1/0/1.0;
    interface fe-1/1/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.181;
      neighbor 10.255.14.177;
      neighbor 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.178;
    }
    group to-vpn-green {
      peer-as 1;
      neighbor 192.168.197.198;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/1.0 {
        passive;
      }
      interface fe-1/1/3.0;
    }
  }
}
```

Configuration for Router J

Configure Router J as a label-swapping router:

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.178;
      neighbor 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.179;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
}
```

Configuration for Router K

Router K acts as the CE router at the end of the connection to the carrier provider. As in the configuration for Router D, include the labeled-unicast statement for the EBGP session:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/2.0;
    interface fe-1/0/2.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.178;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.179;
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.93 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/2.0;
      interface fe-1/1/2.0 {
        passive;
      }
    }
  }
}
policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

Configuration for Router L

Configure Router L to act as the end customer for the carrier-of-carriers VPN service:

```
[edit]
protocols {
  bgp {
    group to-routerl {
      export attached;
      peer-as 21;
      neighbor 192.168.197.197;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Carrier-of-Carriers VPN Example—Customer Provides VPN Service

In this example, the carrier customer *must* run some form of MPLS (Resource Reservation Protocol [RSVP] or LDP) on its network to provide VPN services to the end customer. In the example below, Router B and Router I act as PE routers, and a functioning MPLS path is required between these routers if they exchange VPN-IPv4 routes.

Configuration for Router A

In this example, Router A acts as the CE router for the end customer. Configure a default family inet BGP session on Router A:

```
[edit]
protocols {
  bgp {
    group to-routerB {
      export attached;
      peer-as 21;
      neighbor 192.168.197.169;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Because Router B is the PE router for the end customer CE router (Router A), you need to configure a routing instance (vpna). Configure the labeled-unicast statement on the IBGP session to Router D, and configure family-inet-vpn for the IBGP session to the other side of the network (see Figure 47 on page 422) with Router I:

```
[edit]
protocols {
  mpls {
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.179;
      neighbor 10.255.14.175 {
        family inet {
          labeled-unicast;
          resolve-vpn;
        }
      }
    }
    neighbor 10.255.14.181 {
      family inet-vpn {
        any;
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/3.0;
    }
  }
  ldp {
    interface fe-1/0/3.0;
  }
}
```

```

routing-instances {
  vpna {
    instance-type vrf;
    interface fe-1/0/2.0;
    route-distinguisher 10.255.14.179:21;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      bgp {
        group vpna-06 {
          peer-as 1;
          neighbor 192.168.197.170;
        }
      }
    }
  }
}
policy-options {
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community vpna-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpna-export {
    term a {
      from protocol bgp;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpna-comm members target:100:1001;
}

```

Configuration for Router C

Configure Router C as a label-swapping router within the local AS:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/3.0;
      interface fe-0/3/0.0;
    }
  }
  ldp {
    interface fe-0/3/0.0;
    interface fe-0/3/3.0;
  }
}
```

Configuration for Router D

Router D acts as the CE router for the VPN services provided by the AS 10023 network. In the BGP group configuration for group int, which handles traffic to Router B (10.255.14.179), you include the labeled-unicast statement. You also need to configure the BGP group to-isp-red to send labeled internal routes to the PE router (Router E).

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface fe-0/3/0.0;
    interface t3-0/0/0.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.175;
      neighbor 10.255.14.179 {
        family inet {
          labeled-unicast;
        }
      }
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.13 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/0.0;
    }
  }
  ldp {
    interface fe-0/3/0.0;
  }
}
```

```

policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}

```

Configuration for Router E

Router E and Router H are PE routers. Configure a PE-router-to-PE-router BGP session to allow VPN-IPv4 routes to pass between these two PE routers. Configure the routing instance on Router E to send labeled routes to the CE router (Router D).

Configure Router E:

```

[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-0/1/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.173;
    }
  }
  isis {
    interface at-0/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface at-0/1/0.0;
  }
}

```

```

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}
routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.14 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}

```

Configuration for Router F

Configure Router F to swap labels for routes running through its interfaces:

```
[edit]
protocols {
  isis {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
  }
}
```

Configuration for Router G

Configure Router G:

```
[edit]
protocols {
  isis {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
  }
}
```

Configuration for Router H

The configuration for Router H is similar to the configuration for Router E:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/0.0;
    interface so-1/0/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.173;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.171;
    }
  }
  isis {
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-1/0/0.0;
  }
}
routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.173:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.94 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
}
```

```

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router I

Router I acts as the PE router for the end customer. The configuration that follows is similar to the configuration for Router B:

```

[edit]
protocols {
  mpls {
    interface fe-1/0/1.0;
    interface fe-1/1/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.181;
      neighbor 10.255.14.177 {
        family inet {
          labeled-unicast {
            resolve-vpn;
          }
        }
      }
    }
    neighbor 10.255.14.179 {
      family inet-vpn {
        any;
      }
    }
  }
}

```

```

ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/1/3.0;
  }
}
ldp {
  interface fe-1/1/3.0;
}
}
routing-instances {
  vpna {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.181:21;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      bgp {
        group vpna-0 {
          peer-as 1;
          neighbor 192.168.197.198;
        }
      }
    }
  }
}
}
policy-options {
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community vpna-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpna-export {
    term a {
      from protocol bgp;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpna-comm members target:100:1001;
}
}

```

Configuration for Router J

Configure Router J to swap labels for routes running through its interfaces:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/2.0;
      interface fe-1/0/3.0;
    }
  }
  ldp {
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
}
```

Configuration for Router K

The configuration for Router K is similar to the configuration for Router D:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface fe-1/1/2.0;
    interface fe-1/0/2.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.177;
      neighbor 10.255.14.181 {
        family inet {
          labeled-unicast;
        }
      }
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.93 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
}
```

```

ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/0/2.0;
  }
}
ldp {
  interface fe-1/0/2.0;
}
}
policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
}

```

Configuration for Router L

In this example, Router L is the end customer's CE router. Configure Router L:

```

[edit]
protocols {
  bgp {
    group to-l {
      export attached;
      peer-as 21;
      neighbor 192.168.197.197;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
}

```

Multiple Instances for LDP and Carrier-of-Carriers VPNs

By configuring multiple LDP routing instances, you can use LDP to advertise labels in a carrier-of-carriers VPN from a core provider PE router to a customer carrier CE router. Having LDP advertise labels in this manner is especially useful when the carrier customer is a basic ISP and wants to restrict full Internet routes to its PE routers. By using LDP instead of BGP, the carrier customer shields its other internal routers from the Internet at large. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 3 VPN or Layer 2 VPN services to its customers.

For an example of how to configure multiple LDP routing instances for carrier-of-carriers VPNs, see the *JUNOS Feature Guide* on the product documentation page of the Juniper Networks Web site, located at <http://www.juniper.net/>.

