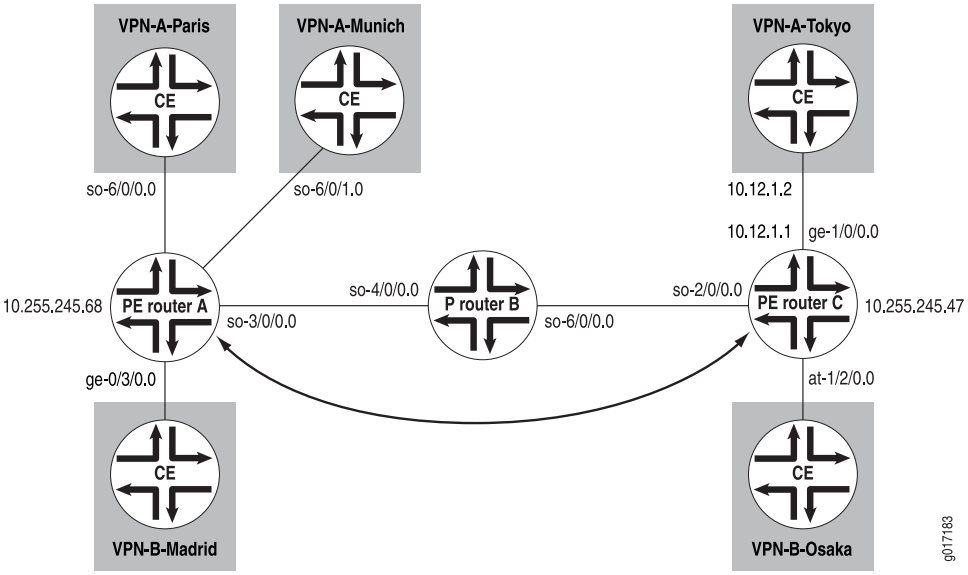
Configuring a Simple Full-Mesh VPN Topology

This example shows how to set up a simple full-mesh service provider VPN configuration, which consists of the following components (see Figure 1):

- Two separate VPNs (VPN-A and VPN-B)
- Two provider edge (PE) routers, both of which service VPN-A and VPN-B
- RSVP as the signaling protocol
- One RSVP label-switched path (LSP) that tunnels between the two PE routers through one provider (P) router

Figure 1: Example of a Simple VPN Topology



In this configuration, route distribution in VPN A from Router VPN-A-Paris to Router VPN-A-Tokyo occurs as follows:

1. The customer edge (CE) router VPN-A-Paris announces routes to the PE router Router A.
2. Router A installs the received announced routes into its VPN routing and forwarding (VRF) table, VPN-A.inet.0.
3. Router A creates an MPLS label for the interface between it and Router VPN-A-Paris.
4. Router A checks its VRF export policy.
5. Router A converts the Internet Protocol version 4 (IPv4) routes from Router VPN-A-Paris into VPN IPv4 format using its route distinguisher and announces these routes to PE Router C over the IBGP between the two PE routers.
6. Router C checks its VRF import policy and installs all routes that match the policy into its bgp.l3vpn.0 routing table. (Any routes that do not match are discarded.)

7. Router C checks its VRF import policy and installs all routes that match into its VPN-A.inet.0 routing table. The routes are installed in IPv4 format.
8. Router C announces its routes to the CE router Router VPN-A-Tokyo, which installs them into its master routing table. (For routing platforms running JUNOS Software, the master routing table is inet.0.)
9. Router C uses the LSP between it and Router A to route all packets from Router VPN-A-Tokyo that are destined for Router VPN-A-Paris.

The final section in this example, "Configuring a Simple Full-Mesh VPN Topology" on page 1, consolidates the statements needed to configure VPN functionality on each of the service P routers shown in Figure 1.

NOTE: In this example, a private autonomous system (AS) number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

The following sections explain how to configure the VPN functionality on the PE and P routers. The CE routers have no information about the VPN, so you configure them normally.

Enabling an IGP on the PE and P Routers

To allow the PE and P routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enabling RSVP and MPLS on the P Router

On the P router, Router B, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the two PE routers, Router A and Router C:

```
[edit]
protocols {
```

```
    rsvp {
        interface so-4/0/0.0;
        interface so-6/0/0.0;
    }
    mpls {
        interface so-4/0/0.0;
        interface so-6/0/0.0;
    }
}
```

Configuring the MPLS LSP Tunnel Between the PE Routers

In this configuration example, RSVP is used for VPN signaling. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the VPN traffic.

On PE Router A, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF. When configuring the MPLS LSP, include **interface** statements for all interfaces participating in MPLS, including the interfaces to the PE and CE routers. The statements for the interfaces between the PE and CE routers are needed so that the PE router can create an MPLS label for the private interface. In this example, the first **interface** statement configures MPLS on the interface connected to the LSP, and the remaining three configure MPLS on the interfaces that connect the PE router to the CE routers.

```
[edit]
protocols {
    rsvp {
        interface so-3/0/0.0;
    }
    mpls {
        label-switched-path RouterA-to-RouterC {
            to 10.255.245.47;
        }
        interface so-3/0/0.0;
        interface so-6/0/0.0;
        interface so-6/0/1.0;
        interface ge-0/3/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-3/0/0.0;
        }
    }
}
```

On PE Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and the CE routers.

```
[edit]
protocols {
    rsvp {
        interface so-2/0/0.0;
```

```
    }
    mpls {
        label-switched-path RouterC-to-RouterA {
            to 10.255.245.68;
        }
        interface so-2/0/0.0;
        interface ge-1/0/0.0;
        interface at-1/2/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-2/0/0.0;
        }
    }
}
```

Configuring IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- VPN family—To indicate that the IBGP session is for the VPN, include the **family inet-vpn** statement.
- Loopback address—Include the **local-address** statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. You must also configure the **lo0** interface at the **[edit interfaces]** hierarchy level. The example does not include this part of the router's configuration.
- Neighbor address—Include the **neighbor** statement, specifying the IP address of the neighboring PE router, which is its loopback (**lo0**) address.

On PE Router A, configure IBGP:

```
[edit]
protocols {
    bgp {
        group PE-RouterA-to-PE-RouterC {
            type internal;
            local-address 10.255.245.68;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.245.47;
        }
    }
}
```

On PE Router C, configure IBGP:

```
[edit]
protocols {
    bgp {
        group PE-RouterC-to-PE-RouterA {
            type internal;
```

```
        local-address 10.255.245.47;
        family inet-vpn {
          unicast;
        }
        neighbor 10.255.245.68;
      }
    }
}
```

Configuring Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A and VPN-B, so you must configure two routing instances on each router, one for each VPN. For each VPN, you must define the following in the routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router.
- It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless an import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.

NOTE: In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

On PE Router A, configure the following routing instance for VPN-A. In this example, Router A uses static routes to distribute routes to and from the two CE routers to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    routing-options {
      static {
        route 172.16.0.0/16 next-hop so-0/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
```

```
        }
      }
    }
  }
```

On PE Router C, configure the following routing instance for VPN-A. In this example, Router C uses BGP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-A-Tokyo {
      instance-type vrf;
      interface ge-1/0/0.0;
      route-distinguisher 65535:1;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
      protocols {
         bgp {
            group VPN-A-Site2 {
               peer-as 1;
               neighbor 10.12.1.2;
            }
         }
      }
   }
}
```

On PE Router A, configure the following routing instance for VPN-B. In this example, Router A uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
   VPN-B-Madrid {
      instance-type vrf;
      interface ge-0/3/0.0;
      route-distinguisher 65535:2;
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
      protocols {
         ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
               interface ge-0/3/0;
            }
         }
      }
   }
}
```

On PE Router C, configure the following routing instance for VPN-B. In this example, Router C uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
```

```
routing-instance {
   VPN-B-Osaka {
      instance-type vrf;
      interface at-1/2/0.0;
      route-distinguisher 65535:3;
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
      protocols {
         rip {
            group PE-C-to-VPN-B {
               export bgp-to-rip;
               neighbor at-1/2/0;
            }
         }
      }
   }
}
```

Configuring VPN Policy on the PE Routers

Configure the VPN import and export policies on each PE router so that the appropriate routes are installed in the PE router's VRF tables. The VRF table is used to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0, and for VPN-B it is VPN-B.inet.0.

In the VPN policy, you also configure VPN target communities.

In the following example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number. The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, for any policies that you configure.

On PE Router A, configure the following VPN import and export policies:

```
[edit]
policy-options {
   policy-statement VPN-A-import {
      term a {
         from {
            protocol bgp;
            community VPN-A;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement VPN-A-export {
      term a {
         from protocol static;
         then {
            community add VPN-A;
            accept;
```

```
        }
      }
      term b {
        then reject;
      }
    }
    policy-statement VPN-B-import {
      term a {
        from {
          protocol bgp;
          community VPN-B;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement VPN-B-export {
      term a {
        from protocol ospf;
        then {
          community add VPN-B;
          accept;
        }
      }
      term b {
        then reject;
      }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

On PE Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
    policy-statement VPN-A-import {
      term a {
        from {
          protocol bgp;
          community VPN-A;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement VPN-A-export {
      term a {
        from protocol bgp;
        then {
          community add VPN-A;
          accept;
```

```
        }
      }
      term b {
        then reject;
      }
    }
    policy-statement VPN-B-import {
      term a {
        from {
          protocol bgp;
          community VPN-B;
        }
        then accept;
      }
      term b {
        then reject;
      }
    }
    policy-statement VPN-B-export {
      term a {
        from protocol rip;
        then {
          community add VPN-B;
          accept;
        }
      }
      term b {
        then reject;
      }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
  }
```

To apply the VPN policies on the routers, include the **vrf-export** and **vrf-import** statements when you configure the routing instance. For both VPNs, the VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on PE Router A, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Madrid {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

To apply the VPN policies on PE Router C, include the following statements:

```
[edit]
```

```
routing-instance {
   VPN-A-Tokyo {
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
   }
   VPN-B-Osaka {
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
   }
}
```

Simple VPN Configuration Summarized by Router

Router A (PE Router)

Routing Instance for VPN-A

```
routing-instance {
   VPN-A-Paris-Munich {
      instance-type vrf;
      interface so-6/0/0.0;
      interface so-6/0/1.0;
      route-distinguisher 65535:0;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
   }
}
```

Instance Routing Protocol

```
routing-options {
   static {
      route 172.16.0.0/16 next-hop so-6/0/0.0;
      route 172.17.0.0/16 next-hop so-6/0/1.0;
   }
}
```

Routing Instance for VPN-B

```
routing-instance {
   VPN-B-Madrid {
      instance-type vrf;
      interface ge-0/3/0.0;
      route-distinguisher 65535:2;
      vrf-import VPN-B-import;
      vrf-export VPN-B-export;
   }
}
```

Instance Routing Protocol

```
protocols {
   ospf {
      area 0.0.0.0 {
         interface ge-0/3/0;
      }
   }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
   interface so-3/0/0.0;
}
```

Configure an MPLS LSP

```
mpls {
   label-switched-path RouterA-to-RouterC {
      to 10.255.245.47;
   }
   interface so-3/0/0.0;
   interface so-6/0/0.0;
   interface so-6/0/1.0;
   interface ge-0/3/0.0;
}
```

Configure IBGP

```
bgp {
   group PE-RouterA-to-PE-RouterC {
      type internal;
      local-address 10.255.245.68;
      family inet-vpn {
         unicast;
      }
      neighbor 10.255.245.47;
   }
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface so-3/0/0.0;
   }
}
```

Configure VPN Policy

```
policy-options {
   policy-statement VPN-A-import {
      term a {
         from {
            protocol bgp;
            community VPN-A;
         }
         then accept;
      }
      term b {
         then reject;
      }
   }
   policy-statement VPN-A-export {
      term a {
         from protocol static;
```

```
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol ospf;
            then {
                community add VPN-B;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

Router B (P Router)

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
}
```

Enable MPLS

```
mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
}
```

Router C (PE Router)

Routing Instance for VPN-A

```
routing-instance {
   VPN-A-Tokyo {
      instance-type vrf;
      interface ge-1/0/0.0;
      route-distinguisher 65535:1;
      vrf-import VPN-A-import;
      vrf-export VPN-A-export;
   }
}
```

Instance Routing Protocol

```
protocols {
   bgp {
      group VPN-A-Site2 {
         peer-as 1;
         neighbor 10.12.1.2;
      }
   }
}
```

Routing Instance for VPN-B

```
VPN-B-Osaka {
   instance-type vrf;
   interface at-1/2/0.0;
   route-distinguisher 65535:3;
   vrf-import VPN-B-import;
   vrf-export VPN-B-export;
}
```

Instance Routing Protocol

```
protocols {
   rip {
      group PE-C-to-VPN-B {
         neighbor at-1/2/0;
      }
   }
}
```

Master Protocol Instance

```
protocols {
}
```

Enable RSVP

```
rsvp {
   interface so-2/0/0.0;
}
```

Configure an MPLS LSP

```
mpls {
   label-switched-path RouterC-to-RouterA {
      to 10.255.245.68;
   }
   interface so-2/0/0.0;
```

```
    interface ge-1/0/0.0;
    interface at-1/2/0.0;
}
```

Configure IBGP

```
bgp {
    group PE-RouterC-to-PE-RouterA {
        type internal;
        local-address 10.255.245.47;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.245.68;
    }
}
```

Configure OSPF for Traffic Engineering Support

```
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface so-2/0/0.0;
    }
}
```

Configure VPN Policy

```
policy-options {
    policy-statement VPN-A-import {
        term a {
            from {
                protocol bgp;
                community VPN-A;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-A-export {
        term a {
            from protocol bgp;
            then {
                community add VPN-A;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-import {
        term a {
            from {
                protocol bgp;
                community VPN-B;
            }
```

```
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement VPN-B-export {
        term a {
            from protocol rip;
            then {
                community add VPN-B;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community VPN-A members target:65535:4;
    community VPN-B members target:65535:5;
}
```

Published: 2010-04-27