Chapter 7
Logical Routers

For many years, engineers have combined power supplies, routing hardware and software, forwarding hardware and software, and physical interfaces into a networking device known as a router. Networking vendors have created large routers and small routers, but all routers have been placed into service as individual devices. As a result, the router has been considered a single physical device for most of its history.

The concept of logical routers breaks with this tradition. With JUNOS software, you can now partition a single physical router into multiple logical devices that perform independent routing tasks. Because logical routers perform a subset of the tasks once handled by the physical router, logical routers offer an effective way to maximize the use of a single router platform.

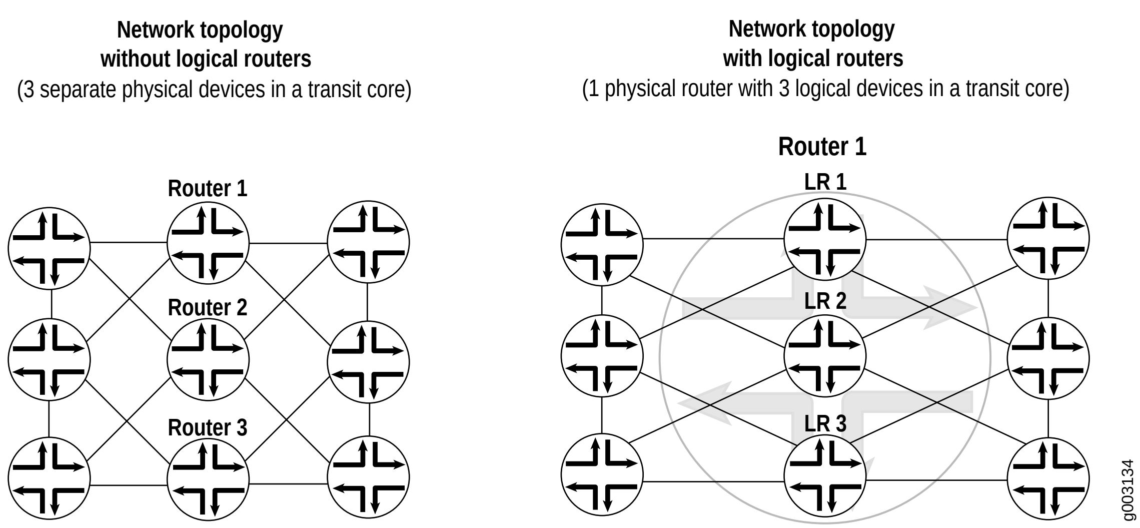
This feature guide covers these topics:

Overview

Traditionally, service provider network design requires multiple layers of switches and routers. These devices transport packet traffic between customers. As seen on the left side of Figure 31, access devices are connected to edge devices, which are in turn connected to core devices.

However, this complexity can lead to challenges in maintenance, configuration, and operation. To reduce such complexity, Juniper Networks now supports logical routers. Logical routers perform a subset of the actions of a physical router and have their own unique routing tables, interfaces, policies, and routing instances. As shown on the right side of Figure 31, a set of logical routers within a single router can handle the functions previously performed by several small routers.

Figure 31: Logical Routers Concept



The following protocols and functions are supported on logical routers:

Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), Routing Information Protocol (RIP), RIP Next-Generation (RIPng), Border Gateway Protocol (BGP), Resource Reservation Protocol (RSVP), Label Distribution Protocol (LDP), static routes, and Internet Protocol Version 4 (IPv4) and Version 6 (IPv6) are supported at the [edit logical-routers protocols] hierarchy level.

Basic Multiprotocol Label Switching (MPLS) for core provider router functionality is supported at the [edit logical-routers protocols mpls] hierarchy level.

All policy-related statements available at the [edit policy-options] hierarchy level are supported at the [edit logical-routers policy-options] hierarchy level.

Most routing options statements available at the [edit routing-options] hierarchy level are supported at the [edit logical-routers routing-options] hierarchy level. Only the route-record statement is not supported at the [edit logical-routers routing-options] hierarchy level.

- You can assign most interface types to a logical router, including SONET interfaces, Ethernet interfaces, Asynchronous Transfer Mode (ATM) interfaces, ATM2 interfaces, channelized Q Performance Processor (QPP) interfaces, aggregated interfaces, Link Services interfaces, and Multilink Services interfaces.
- Source class usage, destination class usage, unicast reverse-path forwarding, class of service, firewall filters, class-based forwarding, and policy-based accounting work with logical routers when you configure these features on the physical router.

The following restrictions apply to logical routers:

- You can configure a maximum of three logical routers on one physical router.
- The router has only one configuration file which contains configuration information for the physical router and all associated logical routers.
- If a logical router experiences an interruption of its routing protocol process (rpd), the core dump output will be placed in a file in the following location: /var/tmp/rpd_*logical-router-name*.core-tarball.*number*.tgz. Likewise, if you issue the restart routing command in a logical router, only the routing protocol process (rpd) for the logical router is restarted.
- If you configure trace options for a logical router, the output log file is stored in the following location: /var/tmp/*logical-router-name.*
- MPLS provider edge (PE) router functions, such as Layer 2 virtual private networks (VPNs), Layer 3 VPNs, circuit cross-connect (CCC), Layer 2 circuits, Virtual Private LAN Service (VPLS), and Generalized MPLS (GMPLS) are not supported.
- The following Physical Interface Cards (PICs) are not supported with logical routers: Adaptive Services PIC, ES PIC, Monitoring Services PIC, and the Monitoring Services II PIC.
- Graceful Routing Engine switchover, sampling, port mirroring, IP Security (IPSec), and multicast protocols are not supported.

It is important to note that a virtual router does not have the same capabilities as a logical router. A virtual router is a type of simplified routing instance that has a single routing table. A logical router is a partition of a physical router and can contain multiple routing instances and routing tables. For example, a logical router can contain multiple virtual router routing instances. As a result, these two entities are not equivalent.

System Requirements

To implement logical routers, your system must meet these minimum requirements:

- JUNOS software version 6.0 or later
- One or more Juniper Networks M-series or T-series platforms with an Internet Processor II ASIC
- A variety of PICs to assign interfaces to each logical router

Terms and Acronyms

logical router—Segmentation of a physical router into multiple logical routing devices. Logical router configuration statements are found at the [edit logical-routers] hierarchy level.

physical router—The standard concept of a router. Physical router configuration statements are found at the [edit] hierarchy level.

Configure Logical Routers

To implement logical routers, you must configure the following:

Configure Interfaces and Assign Logical Interfaces to the Logical Router on page 252

Configure Protocols, Routing, and Policy Statements for the Logical Router on page 253

Option: Configure Other Logical Router Statements on page 253

To apply your knowledge, visit these sections:

Example: Logical Router Configuration on page 254

Check Your Work on page 265

Configure Interfaces and Assign Logical Interfaces to the Logical Router

To add interfaces to a logical router, you must configure the interfaces and assign logical interfaces to each logical router. To configure interfaces, include the *interface-name* statement at the [edit interfaces] hierarchy level, including all subsequent interface-related statements.

After you configure the interfaces, you must assign the logical interfaces to a logical router. To configure, include the unit statement at the [edit logical-routers *logical-router-name* interfaces *interface-name*] hierarchy level. Once you assign logical interfaces to a logical router, they are considered part of the logical router. If you do not assign a logical interface, it remains part of the physical router. You can assign logical interfaces only to one logical router at a time.

```
[edit]
logical-routers logical-router-name {
   interfaces {
      interface-name {
         unit unit-number;
      }
   }
}
interfaces {
   interface-name {
      encapsulation encapsulation-type;
      unit unit-number {
         family inet {
            address ip-address;
         }
      }
   }
}
```

Configure Protocols, Routing, and Policy Statements for the Logical Router

You can configure routing protocols (such as OSPF, BGP, and MPLS), policies (such as next-hop or load-balancing policies), routing options, and routing instances for a logical router.

To configure routing protocols, include the protocols statement at the [edit logical-routers *logical-router-name*] hierarchy level. To configure policies, include the policy-options statement at the [edit logical-routers *logical-router-name*] hierarchy level. To configure routing options, include the routing-options statement at the [edit logical-routers *logical-router-name*] hierarchy level. To configure routing instances, include the routing-instances statement at the [edit logical-routers *logical-router-name*] hierarchy level.

```
[edit]
logical-routers logical-router-name {
   protocols {
      ...
   policy-options {
      ...
   routing-options {
      ...
   routing-instances {
      ...
   }
}
```

Option: Configure Other Logical Router Statements

You can configure a variety of additional statements in conjunction with a logical router:

Filter-based forwarding—You can configure filter-based forwarding for a logical router. To configure, include the logical-router statement at the [edit firewall filter *filter-name* term *term-name* then] hierarchy level. For more information about filter-based forwarding, see the *JUNOS Internet Softw are Configur ation Guide: P olicy Frame work.*

Dynamic Host Control Protocol (DHCP) relay, Bootstrap Protocol (BOOTP), Trivial File Transfer Protocol (TFTP) and Domain Name Service (DNS)—In a logical router, you can configure a DHCP or BOOTP server, and allow TFTP and DNS packets to be forwarded. To configure a DHCP or BOOTP server in a logical router, include the logical-router statement at the [edit forwarding-options helpers bootp interface *interface-name* server *ip-address*] hierarchy level. To configure TFTP packet forwarding in a logical router, include the logical-router statement at the [edit forwarding-options helpers tftp interface *interface-name* server *ip-address*] hierarchy level. To configure DNS packet forwarding in a logical router, include the logical-router statement at the [edit forwarding-options helpers domain interface *interface-name* server *ip-address*] hierarchy level. For more information about DHCP relay, BOOTP, TFTP, or DNS, see the *JUNOS Internet Software Configur ation Guide: P olicy Frame work.*

You can select to view only the operational mode and configuration mode hierarchies for a specific logical router. To configure a logical router view, issue the set cli logical-router *logical-router-name* command. To clear the logical router view and return to a full router view, issue the clear cli logical-router command. For an example of these commands, see “Router P0 Status: Logical Router” on page 268.

Example: Logical Router Configuration

Figure 32: Logical Router Topology Diagram

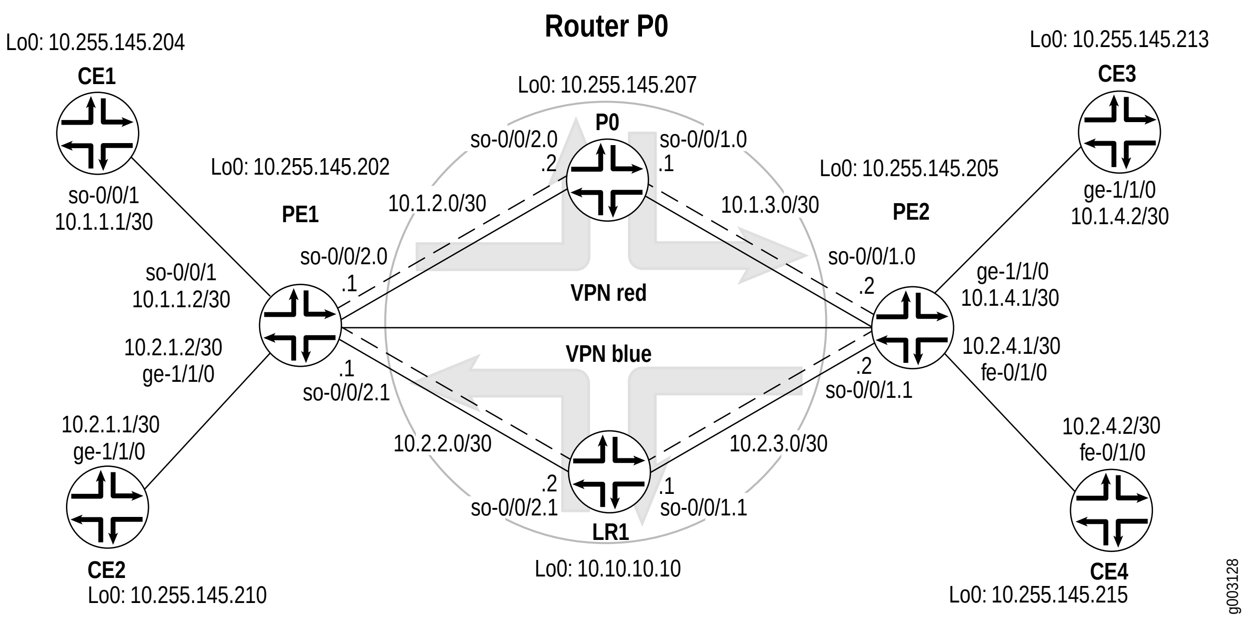


Figure 32 shows two pairs of customer edge (CE) routers that are connected across an MPLS backbone. Routers CE1 and CE3 are part of the red VPN, while routers CE2 and CE4 are in the blue VPN. Both VPNs are configured on provider edge (PE) routers PE1 and PE2. Provider core router P0 has one logical router LR1. To illustrate the concept of a logical router, the red VPN traffic travels through the physical part of router P0, while the blue VPN traffic traverses the logical router LR1 in router P0.

On router CE1, configure external BGP to connect to router PE1. Define a policy to export only the routes that originate from the loopback interface and apply the policy to your external BGP configuration.

CE1

```
[edit]
interfaces {
    so-0/0/1 {
        description "to router PE1";
        unit 0 {
            family inet {
                address 10.1.1.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.255.145.204/32;
            }
        }
    }
}
```

```
protocols {
  bgp {                              # Use BGP as the routing protocol to connect to router PE1.
    group to_PE1 {
      type external;
      local-address 10.1.1.1;
      export policy1;                # Apply the export policy for your locally originated BGP routes.
      peer-as 100;
      neighbor 10.1.1.2;
    }
  }
}
routing-options {
  autonomous-system 200;
}
policy-options {
  policy-statement policy1 {         # Define your BGP export policy here.
    term 1 {
      from {
        route-filter 10.255.145.204/32 exact;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}
```

On router CE2, configure OSPF to connect to router PE1.

CE2

```
[edit]
interfaces {
  ge-1/1/0 {
    unit 0 {
      family inet {
        address 10.2.1.1/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.145.210/32;
      }
    }
  }
}
protocols {
  ospf {                             # Use OSPF as the routing protocol to connect to router PE1.
    area 0.0.0.0 {
      interface ge-1/1/0.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
```

On router PE1, create two VPN routing and forwarding (VRF) routing instances: red and blue. Configure the CE-facing interfaces so that traffic from router CE1 is placed in the red VPN and traffic from router CE2 is placed in the blue VPN. Next, create two logical interfaces on the so-0/0/2 connection to router P0: data-link connection identifier (DLCI) 100 and DLCI 200. Configure DLCI 100 (so-0/0/2.0) to connect to the physical part of router P0 and DLCI 200 (so-0/0/2.1) to connect to the logical router at router P0.

To send traffic from the two VPNs over different label-switched paths (LSPs), create two different LSPs: one to travel through the physical router at P0, the other to travel through the logical router at P0. Associate the VRF targets of the two VPNs with two separate communities: blue-com and red-com. Then, create two policies: one that directs traffic from the red-com community across the physical router LSP and another that points blue-com community traffic across the logical router LSP. Finally, apply these policies to the forwarding table as an export routing option.

PE1

```
[edit]
interfaces {
  so-0/0/1 {
    description "to router CE1";
    unit 0 {
      family inet {
        address 10.1.1.2/30;
      }
      family mpls;
    }
  }
  so-0/0/2 {
    description "to router P0";
    encapsulation frame-relay;
    unit 0 {                # This connects to the physical router on router P0.
      dlci 100;
      family inet {
        address 10.1.2.1/30;
      }
      family iso;
      family mpls;
    }
    unit 1 {                # This connects to logical router LR1 on router P0.
      dlci 200;
      family inet {
        address 10.2.2.1/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-1/1/0 {
    description "to router CE2";
    unit 0 {
      family inet {
        address 10.2.1.2/30;
      }
    }
  }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.255.145.202/32;
            }
        }
    }
}
routing-options {
    autonomous-system 100;
    forwarding-table {
        export [ blue-policy red-policy ];      # These policies match route communities with LSPs.
    }
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path to_PE2_main {     # This LSP travels through P0's physical router.
            to 10.255.145.205;
            primary main;
        }
        label-switched-path to_PE2_lr {       # This LSP travels through P0's logical router LR1.
            to 10.255.145.205;
            primary logical;
        }
        path main {
            10.1.2.2 strict;
            10.1.3.2 strict;
        }
        path logical {
            10.2.2.2 strict;
            10.2.3.2 strict;
        }
        interface all;
    }
    bgp {
        group to_PE2 {                 # Use internal BGP to connect to router PE2.
            type internal;
            local-address 10.255.145.202;
            family inet-vpn {
                any;
            }
            neighbor 10.255.145.205;
        }
    }
    isis {                             # Use IS-IS as the IGP for the connection to router PE2.
        interface so-0/0/2.0;
        interface so-0/0/2.1;
        interface lo0.0 {
            passive;
        }
    }
}
```

```
routing-instances {
  blue {
    instance-type vrf;
    interface ge-1/1/0.0;
    route-distinguisher 10.255.145.202:200;
    vrf-target target:10:10;    # Use this value to add the blue VPN into a blue-com community.
    protocols {
      ospf {            # Use OSPF as the routing protocol to connect the blue VPN to router CE2.
        export advertise_all_routes;
        area 0.0.0.0 {
          interface ge-1/1/0.0;
        }
      }
    }
  }
  red {
    instance-type vrf;
    interface so-0/0/1.0;
    route-distinguisher 10.255.145.202:100;
    vrf-target target:20:20;    # Use this value to add the red VPN into a red-com community.
    protocols {
      bgp {             # Use BGP as the routing protocol to connect the red VPN to router CE1.
        group to_CE1 {
          local-address 10.1.1.2;
          peer-as 200;
          neighbor 10.1.1.1;
        }
      }
    }
  }
}
policy-options {
  policy-statement advertise_all_routes {
    then accept;
  }
  policy-statement blue-policy {  # This policy sends blue VPN traffic across P0's logical router.
    from community blue-com;
    then {
      install-nexthop lsp to_PE2_lr;
    }
  }
  policy-statement red-policy {   # This policy sends red VPN traffic across P0's physical router.
    from community red-com;
    then {
      install-nexthop lsp to_PE2_main;
    }
  }
  community blue-com members target:10:10;# blue traffic joins the blue-com community.
  community red-com members target:20:20;  # red traffic joins the red-com community.
}
```

On router P0, configure the logical router LR1 and the physical router. For the logical router, you must configure interfaces on the physical router and assign the logical interfaces to the logical router. Remember to include all interface-related statements as part of the [edit interfaces] hierarchy level, such as encapsulation types, logical interfaces, and IP addresses. Next, you must configure protocols (such as RSVP, MPLS, and IS-IS), routing options, and policy options for the logical router. In this example, the logical router transports traffic for the blue VPN that exists between routers PE1 and PE2.

For the physical router on router P0, you can configure the router as usual. In this example, the physical router transports traffic for the red VPN that exists between routers PE1 and PE2. As a result, configure the interfaces and routing protocols (RSVP, MPLS, and IS-IS) to transport red VPN traffic.

P0

```
[edit]
logical-routers {      # Here is the hierarchy level that is used to define logical router properties.
  lr1 {
    interfaces {       # This is where you assign logical interfaces to the logical router.
      so-0/0/1 {
        unit 1;        # Use the unit statement to assign logical interfaces to the logical router.
      }
      so-0/0/2 {
        unit 1;        # Use the unit statement to assign logical interfaces to the logical router.
      }
      lo0 {
        unit 1;        # Use the unit statement to assign logical interfaces to the logical router.
      }
    }
    protocols { # To transport the blue VPN, enable RSVP, MPLS, and IS-IS on the logical router.
      rsvp {
        interface all;
      }
      mpls {
        interface all;
      }
      isis {
        interface so-0/0/1.1;
        interface so-0/0/2.1;
        interface lo0.1 {
          passive;
        }
      }
    }
  }
}
interfaces {
  so-0/0/1 {
    description "to router PE2";
    dce;
    encapsulation frame-relay;
    unit 0 {              # This connects the physical router to router PE2 and the red VPN.
      dlci 100;
      family inet {
        address 10.1.3.1/30;
      }
      family iso;
      family mpls;
    }
    unit 1 {              # This connects the logical router to router PE2 and the blue VPN.
      dlci 200;
      family inet {
        address 10.2.3.1/30;
      }
      family iso;
      family mpls;
    }
  }
```

```
    so-0/0/2 {
        description "to router PE1";
        dce;
        encapsulation frame-relay;
        unit 0 {    # This connects the physical router to router PE1 and the red VPN.
            dlci 100;
            family inet {
                address 10.1.2.2/30;
            }
            family iso;
            family mpls;
        }
        unit 1 {    # This connects the logical router to router PE1 and the blue VPN.
            dlci 200;
            family inet {
                address 10.2.2.2/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {    # The IS-IS AFI and IP address of this logical interface are part of the red VPN.
            family inet {
                address 10.255.145.207/32;
            }
            family iso {
                address 47.1111.1111.1111.1111.00;
            }
        }
        unit 1 {    # The IS-IS AFI and IP address of this logical interface are part of the blue VPN.
            family inet {
                address 10.10.10.10/32;
            }
            family iso {
                address 47.2222.2222.2222.2222.00;
            }
        }
    }
}
protocols {    To transport red VPN traffic, enable RSVP, MPLS, and IS-IS for the physical router.
    rsvp {
        interface all;
    }
    mpls {
        interface all;
    }
    isis {
        interface so-0/0/1.0;
        interface so-0/0/2.0;
        interface lo0.0 {
            passive;
        }
    }
}
```

On router PE2, create two VRF routing instances: red and blue. Configure the CE-facing interfaces so that traffic from router CE3 is placed into the red VPN and traffic from router CE4 is placed into the blue VPN. Next, create two logical interfaces on the so-0/0/1 connection to router P0: DLCI 100 and DLCI 200. Configure DLCI 100 (so-0/0/1.0) to connect to the physical part of router P0 and DLCI 200 (so-0/0/1.1) to connect to the logical router LR1 at router P0.

To send traffic from the two VPNs over different LSPs, create two different LSPs: one to travel through the physical router at P0, the other to travel through the logical router at P0. Associate the VRF targets of the two VPNs with two separate communities: blue-com and red-com. Then, create two policies: one that directs traffic from the red-com community across the physical router LSP and another that points blue-com community traffic across the logical router LSP. Finally, apply these policies to the forwarding table as an export routing option.

PE2

```
[edit]
interfaces {
  so-0/0/1 {
    description "to router P0";
    encapsulation frame-relay;
    unit 0 {          # This connects to the physical router on router P0.
      dlci 100;
      family inet {
        address 10.1.3.2/30;
      }
      family iso;
      family mpls;
    }
    unit 1 {          # This connects to logical router LR1 on router P0.
      dlci 200;
      family inet {
        address 10.2.3.2/30;
      }
      family iso;
      family mpls;
    }
  }
  fe-0/1/0 {
    description "to router CE4";
    unit 0 {
      family inet {
        address 10.2.4.1/30;
      }
    }
  }
  ge-1/1/0 {
    description "to router CE3";
    unit 0 {
      family inet {
        address 10.1.4.1/30;
      }
      family mpls;
    }
  }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.255.145.205/32;
            }
        }
    }
}
routing-options {
    autonomous-system 100;
    forwarding-table {
        export [ blue-policy red-policy ];    # These policies match route communities with LSPs.
    }
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        label-switched-path to_PE1_main {    # This LSP travels through P0's physical router.
            to 10.255.145.202;
            primary main;
        }
        label-switched-path to_PE1_lr {      # This LSP travels through P0's logical router LR1.
            to 10.255.145.202;
            primary logical;
        }
        path main {
            10.1.3.1 strict;
            10.1.2.1 strict;
        }
        path logical {
            10.2.3.1 strict;
            10.2.2.1 strict;
        }
        interface all;
    }
    bgp {
        group to_PE1 {                # Use internal BGP to connect to router PE1.
            type internal;
            local-address 10.255.145.205;
            family inet-vpn {
                any;
            }
            neighbor 10.255.145.202;
        }
    }
isis {                                # Use IS-IS as the IGP for the connection to router PE1.
        interface so-0/0/1.0;
        interface so-0/0/1.1;
        interface lo0.0 {
            passive;
        }
    }
}
```

```
routing-instances {
   blue {
      instance-type vrf;
      interface fe-0/1/0.0;
      route-distinguisher 10.255.145.205:200;
      vrf-target target:10:10;    # Use this value to add the blue VPN into a blue-com community.
      protocols {
         ospf {          # Use OSPF as the routing protocol to connect the blue VPN to router CE4.
            export advertise_all_routes;
            area 0.0.0.0 {
               interface fe-0/1/0.0;
            }
         }
      }
   }
   red {
      instance-type vrf;
      interface ge-1/1/0.0;
      route-distinguisher 10.255.145.205:100;
      vrf-target target:20:20;    # Use this value to add the red VPN into a red-com community.
      protocols {
         bgp {           # Use BGP as the routing protocol to connect the red VPN to router CE3.
            group to_CE3 {
               local-address 10.1.4.1;
               peer-as 300;
               neighbor 10.1.4.2;
            }
         }
      }
   }
}
policy-options {
   policy-statement advertise_all_routes {
      then accept;
   }
   policy-statement red-policy {     # This policy sends red VPN traffic across P0's physical router.
      from community red-com;
      then {
         install-nexthop lsp to_PE1_main;
      }
   }
   policy-statement blue-policy {  # This policy sends blue VPN traffic across P0's logical router.
      from community blue-com;
      then {
         install-nexthop lsp to_PE1_lr;
      }
   }
   community blue-com members target:10:10;# blue traffic joins the blue-com community.
   community red-com members target:20:20;  # red traffic joins the red-com community.
}
```

On router CE3, configure external BGP to connect to router PE2. Define a policy to export only the routes that originate from the loopback interface and apply the policy to your external BGP configuration. The red VPN at router PE2 will complete the path from router CE3 to router CE1.

CE3

```
[edit]
interfaces {
   ge-1/1/0 {
      unit 0 {
         family inet {
            address 10.1.4.2/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.255.145.213/32;
         }
      }
   }
}
protocols
   bgp {                          # Use BGP as the routing protocol to connect to router PE2.
      group to_PE2 {
         type external;
         local-address 10.1.4.2;
         export policy1;          # Apply the export policy for your locally originated BGP routes.
         peer-as 100;
         neighbor 10.1.4.1;
      }
   }
}
policy-options {
   policy-statement policy1 {     # Define your BGP export policy here.
      term 1 {
         from {
            route-filter 10.255.145.213/32 exact;
         }
         then accept;
      }
      term 2 {
         then reject;
      }
   }
}
```

On router CE4, configure OSPF to connect to router PE2. The blue VPN at router PE2 will complete the path from router CE4 to router CE2.

CE4
```
[edit]
interfaces {
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.2.4.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.255.145.215/32;
      }
    }
  }
}
protocols {                # Use OSPF as the routing protocol to connect to router PE2.
  ospf {
    area 0.0.0.0 {
      interface fe-0/1/0.0;
      interface lo0.0;
        passive;
      }
    }
  }
}
```

Check Your Work

To verify the proper operation of logical routers, use the following commands:

show isis adjacency (logical-router *logical-router-name*)

show mpls lsp (logical-router *logical-router-name*)

show route (logical-router *logical-router-name*)

show rsvp session (logical-router *logical-router-name*)

The following sections show the output of commands used with the configuration example:

Router PE1 Status on page 266

Router P0 Status: Physical Router on page 266

Router P0 Status: Logical Router on page 268

Verify that the Different VPNs Use Different LSPs on page 269

Router PE1 Status

At router PE1, the red VPN route to router CE3 (10.255.145.213) is reachable by the LSP named to_PE2_main. This LSP travels through router P0's main physical router.

The blue VPN route to router CE4 (10.255.145.215) is reachable by the LSP named to_PE2_lr. This LSP travels through router P0's logical router LR1.

```
user@PE1> show route 10.255.145.213 table red

red.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.145.213/32  *[BGP/170] 00:14:03, localpref 100, from 10.255.145.205
                      AS path: 300 I
                    > via so-0/0/2.0, label-switched-path to_PE2_main

user@PE1> show route 10.255.145.215 table blue

blue.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.145.215/32  *[BGP/170] 00:14:09, MED 1, localpref 100, from
10.255.145.205
                      AS path: I
                    > via so-0/0/2.1, label-switched-path to_PE2_lr

user@PE1> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
so-0/0/2.0            P0             3 Up                   20
so-0/0/2.1            1111.1111.1111 2 Up                   24

user@PE1> show mpls lsp
Ingress LSP: 2 sessions
To               From             State Rt ActivePath       P     LSPname
10.255.145.205  10.255.145.202  Up     0 main             *     to_PE2_main
10.255.145.205  10.255.145.202  Up     0 logical          *     to_PE2_lr
Total 2 displayed, Up 2, Down 0

Egress LSP: 2 sessions
To               From             State Rt Style Labelin Labelout LSPname
10.255.145.202  10.255.145.205  Up     0 1 FF       3        - to_PE1_main
10.255.145.202  10.255.145.205  Up     0 1 FF       3        - to_PE1_lr
Total 2 displayed, Up 2, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Router P0 Status: Physical Router

P0 consists of the physical router itself and logical router LR1. IS-IS is configured in both the main physical router and the logical router. Notice that only the physical router interfaces appear in the physical router IS-IS adjacencies and routing tables. Likewise, only the physical router LSPs and RSVP sessions appear in the physical router operational mode command output.

P0

```
user@P0> show interfaces terse lo0.0
Interface               Admin Link Proto Local                 Remote
lo0                     up    up
lo0.0                   up    up   inet  10.255.145.207      --> 0/0
                                         127.0.0.1           --> 0/0
                                   iso   47.1111.1111.1111.1111.00
                                   inet6 feee::10:255:145:207
lo0.16383               up    up   inet

user@P0> show route protocol direct

inet.0: 19 destinations, 19 routes (18 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.255.145.207/32  *[Direct/0] 2d 14:56:26
                    > via lo0.0
10.1.2.0/30        *[Direct/0] 06:33:08
                    > via so-0/0/2.0
10.1.3.0/30        *[Direct/0] 00:54:34
                    > via so-0/0/1.0

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.1111.1111.1111.1111.00/80
                   *[Direct/0] 2d 14:56:26
                    > via lo0.0

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

user@P0> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
so-0/0/1.0            PE2              3 Up                   24
so-0/0/2.0            PE1              3 Up                   19

user@P0> show mpls lsp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.145.202  10.255.145.205  Up     1 1 FF  100016       3 to_PE1_main
10.255.145.205  10.255.145.202  Up     1 1 FF  100000       3 to_PE2_main
Total 2 displayed, Up 2, Down 0

user@P0> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.145.202  10.255.145.205  Up     1 1 FF  100016       3 to_PE1_main
10.255.145.205  10.255.145.202  Up     1 1 FF  100000       3 to_PE2_main
Total 2 displayed, Up 2, Down 0
```

Router P0 Status: Logical Router

For logical router LR1 on router P0, only the logical router interfaces appear in the logical router IS-IS adjacencies. Likewise, only the logical router LSPs and RSVP sessions appear in the logical router operational mode command output.

```
user@P0> show interfaces terse lo0.1
Interface               Admin Link Proto Local                 Remote
lo0                     up    up
lo0.1                   up    up   inet  10.10.10.10          --> 0/0
                                   iso   47.2222.2222.2222.2222.00
lo0.16383               up    up   inet

user@P0> show isis adjacency logical-router lr1
Interface             System         L State        Hold (secs) SNPA
so-0/0/1.1            PE2              2 Up                   20
so-0/0/2.1            PE1              2 Up                   24

user@P0> show rsvp session logical-router lr1
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.145.202  10.255.145.205  Up     1 1 FF  100016       3 to_PE1_lr
10.255.145.205  10.255.145.202  Up     1 1 FF  100000       3 to_PE2_lr
Total 2 displayed, Up 2, Down 0

user@P0> show mpls lsp logical-router lr1
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To              From            State Rt Style Labelin Labelout LSPname
10.255.145.202  10.255.145.205  Up     1 1 FF  100016       3 to_PE1_lr
10.255.145.205  10.255.145.202  Up     1 1 FF  100000       3 to_PE2_lr
Total 2 displayed, Up 2, Down 0
```

You can select to view only the operational mode and configuration mode hierarchies for a specific logical router. To configure a logical router view, issue the set cli logical-router *logical-router-name* command.

When you enter logical router view mode and enter an operational mode command, the output of the command displays information related to the logical router only. For example, when you issue the show route command, the output shows only the routes that are assigned to the logical router.

```
user@P0> set cli logical-router lr1
Logical router: lr1

user@P0:lr1>         # Note that the user is now restricted to a logical router view.
```

```
user@P0:lr1> show route protocol direct

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.10.10/32     *[Direct/0] 00:55:30
                    > via lo0.1
10.2.2.0/30        *[Direct/0] 00:55:30
                    > via so-0/0/2.1
10.2.3.0/30        *[Direct/0] 00:55:30
                    > via so-0/0/1.1

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

47.2222.2222.2222.2222.00/80
                   *[Direct/0] 00:55:30
                    > via lo0.1

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
```

To clear the logical router view and return to a full router view, issue the clear cli logical-router command.

```
user@P0:lr1> clear cli logical-router
Cleared default logical router

user@P0>                # Note that the user can now view the entire router again.
```

Verify that the Different VPNs Use Different LSPs

To verify that traffic for the red VPN travels from router CE1 to CE3 through router P0's physical router, issue the ping command on router CE1 to send 200 packets to router CE3. Then, issue the show mpls lsp statistics and show mpls lsp statistics logical-router lr1 commands on router P0 to confirm that red VPN traffic traverses the physical router LSPs.

CE1

```
user@CE1> ping 10.255.145.213 rapid count 200
PING 10.255.145.213 (10.255.145.213): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 10.255.145.213 ping statistics ---
200 packets transmitted, 200 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.720/0.846/9.881/0.972 ms
```

P0

```
user@P0> show mpls lsp statistics
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To               From             State     Packets            Bytes LSPname
10.255.145.202   10.255.145.205   Up            200            18400 to_PE1_main
10.255.145.205   10.255.145.202   Up            200            18400 to_PE2_main
Total 2 displayed, Up 2, Down 0
```

LR1

```
user@P0> show mpls lsp statistics logical-router lr1
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To              From            State     Packets            Bytes LSPname
10.255.145.202  10.255.145.205  Up              0                0 to_PE1_lr
10.255.145.205  10.255.145.202  Up              0                0 to_PE2_lr
Total 2 displayed, Up 2, Down 0
```

To verify that traffic for the blue VPN travels from router CE2 to CE4 through router P0's logical router LR1, clear the counters on router P0 and logical router LR1, send a ping with 300 packets from router CE2 to CE4, and check the results on router P0 and logical router LR1 to confirm that blue VPN traffic traverses the logical router LSPs.

P0

```
user@P0> clear mpls lsp statistics
user@P0> clear mpls lsp statistics logical-router lr1
```

CE2

```
user@CE2> ping 10.255.145.215 rapid count 300
PING 10.255.145.215 (10.255.145.215): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 10.255.145.215 ping statistics ---
300 packets transmitted, 300 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.739/0.864/9.920/0.953 ms
```

P0

```
user@P0> show mpls lsp statistics
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To              From            State     Packets            Bytes LSPname
10.255.145.202  10.255.145.205  Up              0                0 to_PE1_main
10.255.145.205  10.255.145.202  Up              0                0 to_PE2_main
Total 2 displayed, Up 2, Down 0
```

LR1

```
user@P0> show mpls lsp statistics logical-router lr1
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 2 sessions
To              From            State     Packets            Bytes LSPname
10.255.145.202  10.255.145.205  Up            300            27600 to_PE1_lr
10.255.145.205  10.255.145.202  Up            300            27600 to_PE2_lr
```

For More Information

Because the concepts that comprise logical routers cut across the entire JUNOS software documentation set, you will find the following manuals to be useful references:

For additional information about routing protocols, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

For additional information about policies, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

For additional information about interface configuration, see the *JUNOS Internet Software Configuration Guide: Network Interfaces and Class of Service*.

For additional information about MPLS and related protocols, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

For additional information about VPN protocols, see the *JUNOS Internet Software Configuration Guide: VPNs*.

For additional information about user access privileges and router operations, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

For additional information about operational mode commands and output, see the *JUNOS Internet Software Command Reference: Interfaces* and the *JUNOS Internet Software Command Reference: Protocols, Class of Service, Chassis, and Management*.

Revision History

30 June 2003—Initial document written, Release 6.0R1. Richard Hendricks.