Chapter 11

Graceful Restart

With standard implementations of routing protocols, any service interruption requires an affected router to recalculate adjacencies with neighboring routers, restore routing table entries, and update other protocol-specific information. As a result, an unprotected restart of a router can result in forwarding delays, route flapping, wait times stemming from protocol reconvergence, and even dropped packets.

In contrast, graceful restart allows a restarting router and its neighbors to continue forwarding packets without disrupting network performance. Because neighboring routers assist in the restart, the restarting router can quickly resume full operation without having to recalculate algorithms from scratch.

This feature guide covers these topics:

- Overview on page 450
- System Requirements on page 455
- Terms and Acronyms on page 455
- Configuring Routing Protocol and Summaries Graceful Restart on page 456
- Configuring Graceful Restart for an MPLS-Related Protocol on page 460
- Configuring VPN Graceful Restart on page 462
  - Example: Layer 3 VPN Graceful Restart Configuration on page 464
  - Checking Your Work on page 482
- Configuring Logical Router Graceful Restart on page 495
- For More Information on page 497
- Revision History on page 498

Overview

Three main types of graceful restart available on Juniper Networks routing platforms are:

- Graceful restart for routing protocols and summaries—Provides protection for aggregate routes, Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), next-generation RIP (RIPng), Protocol Independent Multicast (PIM) sparse mode, and static routes.

- Graceful restart for MPLS-related protocols—Provides protection for Label Distribution Protocol (LDP), Resource Reservation Protocol (RSVP), circuit cross-connect (CCC), and translational cross-connect (TCC).

- Graceful restart for virtual private networks (VPNs)—Provides protection for Layer 2 and Layer 3 VPNs.

Graceful restart works similarly for routing protocols and MPLS protocols and combines components of these protocol types to enable graceful restart in VPNs. The main benefits of graceful restart are uninterrupted packet forwarding and temporary suppression of all routing protocol updates. Graceful restart thus allows a router to pass through intermediate convergence states that are hidden from the rest of the network.

Most graceful restart implementations define two types of routers—the restarting router and the helper router. The restarting router requires rapid restoration of forwarding state information so it can resume the forwarding of network traffic. The helper router assists the restarting router in this process. Graceful restart configuration statements typically affect either the restarting router or the helper router. A brief description of graceful restart for each supported protocol follows:

Routing Protocols

BGP—When a router enabled for BGP graceful restart restarts, it retains BGP peer routes in its forwarding table and marks them as stale. However, it continues to forward traffic to other peers (or receiving peers) during the restart. To re-establish sessions, the restarting router sets the “restart state” bit in the BGP OPEN message and sends it to all participating peers. The receiving peers reply to the restarting router with messages containing end-of-routing-table markers. When the restarting router receives all replies from the receiving peers, the restarting router performs route selection, the forwarding table is updated, and the routes previously marked as stale are discarded. At this point, all BGP sessions are re-established and the restarting peer can receive and process BGP messages as usual.

While the restarting router does its processing, the receiving peers also temporarily retain routing information. Once a receiving peer detects a TCP transport reset, it retains the routes received and marks the routes as stale. After the session is re-established with the restarting router, the stale routes are replaced with updated route information.

IS-IS—Normally, IS-IS routers move neighbor adjacencies to the down state when changes occur. However, a router enabled for IS-IS graceful restart sends out Hello messages with the Restart Request (RR) bit set in a restart type length value (TLV) message. This indicates to neighboring routers that a graceful restart is in progress and that the IS-IS adjacency should be left intact. For this to work, the neighboring routers must understand and implement restart signaling themselves. Besides maintaining the adjacency, the neighbors send complete sequence number PDUs (CSNPs) to the restarting router and flood their entire database.

The restarting router never floods any of its own link-state PDUs (LSPs), including pseudonode LSPs, to IS-IS neighbors while undergoing graceful restart. This allows neighbors to re-establish their adjacencies without transitioning to the down state and allows the restarting router to re-initiate a smooth database synchronization.

OSPF—When a router enabled for OSPF graceful restart restarts, it retains routes learned prior to the restart in its forwarding table. The router does not allow new OSPF link-state advertisements (LSAs) to update the routing table. This router continues to forward traffic to other OSPF neighbors (or *helper routers*), and sends only a limited number of LSAs during the restart period. To re-establish OSPF adjacencies with neighbors, the restarting router must send a grace LSA to all neighbors. In response, the helper routers enter helper mode and send an acknowledgement back to the restarting router. Also, if there are no topology changes, the helper routers continue to advertise LSAs as if the restarting router had remained in continuous OSPF operation.

When the restarting router receives replies from all the helper routers, the restarting router selects routes, updates the forwarding table, and discards the old routes. At this point, full OSPF adjacencies are re-established and the restarting router receives and processes OSPF LSAs as usual. When the helper routers no longer receive grace LSAs from the restarting router or the topology of the network changes, the helper routers also resume normal operation.

Multicast Protocols

PIM sparse mode—This multicast protocol uses a mechanism called a *generation identifier* to indicate the need for graceful restart. Generation identifiers are included by default in PIM hello messages, as specified in the IETF Internet draft draft-ietf-pim-sm-v2-new-10.txt, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)* (expires January 2005). An initial generation identifier is created by each PIM neighbor to establish device capabilities. When one of the PIM neighbors restarts, it sends a new generation identifier to its neighbors. All neighbors that support graceful restart and are connected by point-to-point links assist by sending multicast updates to the restarting neighbor.

The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires. If the neighbors do not support graceful restart or connect to each other using multipoint interfaces, the restarting router uses the restart interval timer to define the restart period.

MPLS-related Protocols

RSVP—This protocol uses a field called the restart capabilities object. This object is sent in RSVP Hello messages to peers and is used to advertise a router's RSVP restart capabilities. When an RSVP-enabled router restarts, a Hello message is sent to neighbors (or helper routers) to indicate a restart is in progress. Helper routers reply to the restarting router with an RSVP PATH message that contains a recovery label. The recovery label contains the information from a previous label that was advertised by the restarting node before it restarted. After receiving the recovery label, the restarting router can restore its previous forwarding state and resume operation. In JUNOS Release 6.1 and later, RSVP graceful restart is also supported on ingress provider edge routers.

CCC and TCC—These two protocols rely on RSVP for all graceful restart functionality. As a result, there are no configuration statements unique to CCC and TCC. However, you can use the show connections and show route protocol ccc commands to verify that CCC and TCC graceful restart is operating. CCC and TCC graceful restart is supported on label-switched path (LSP) switch and remote interface switch connections.

LDP—This protocol uses the Fault Tolerant (FT) Session TLV as an optional parameter in the LDP Initialization message. Routers exchange this TLV during session initialization to advertise their capability to perform graceful restart or act as a helper router. When an LDP router restarts, it sends an initialization message to neighbors. The message advertises the length of time the helper routers are requested to assist the restarting router. During this recovery time, both routers maintain MPLS forwarding states. The restarting router marks all routes as stale and discards the routes when full neighborship is re-established and the restart is complete.

The neighbor (or helper router) of the restarting router marks all label bindings it received from the restarting router as stale and waits for them to be refreshed or to expire at the end of the recovery time. On the helper router, a local timer governs the maximum amount of time the helper router is willing to maintain forwarding states. LDP graceful restart can be configured in a master instance or in a routing instance and supports a carrier-of-carriers scenario.

Layer 2 and Layer 3 VPNs—VPN graceful restart uses three types of restart functionality:

1. BGP graceful restart functionality is used on all provider edge (PE) to PE BGP sessions. This affects sessions carrying any service signaling data for network layer reachability information (NLRI), for example, an IPv4 VPN or Layer 2 VPN NLRI.

2. OSPF, ISIS, LDP, or RSVP graceful restart functionality is used in all core routers. Routes added by these protocols are used to resolve Layer 2 and Layer 3 VPN NLRI.

3. Protocol restart functionality is used for any Layer 3 protocol (RIP, OSPF, LDP, and so on) used between the PE and customer edge (CE) routers. This does not apply to Layer 2 VPNs because Layer 2 protocols used between the CE and PE routers do not have graceful restart capabilities.

Before VPN graceful restart can work properly, all of the above components should restart gracefully. In other words, the routers should preserve their forwarding states and request neighbors to continue forwarding to the router in case of a restart. If all of the above conditions are satisfied, VPN graceful restart imposes the following rules on a restarting router:

- The router must wait to receive all BGP NLRI information from other PE routers before advertising routes to the CE routers.

- The router must wait for all protocols in all routing instances to converge (or complete the restart process) before it sends CE router information to other PE routers. In other words, the router must wait for all instance information (whether derived from local configuration or advertisements received from a remote peer) to be processed before it sends this information to other PE routers.

- The router must preserve all forwarding state in the *instance*.mpls.0 tables until the new labels and transit routes are allocated and announced to other PE routers (and CE routers in a carrier-of-carriers scenario).

If any condition is not met, VPN graceful restart will not succeed in providing uninterrupted forwarding between CE routers across the VPN infrastructure.

Logical routers—Graceful restart for a logical router functions much in the same way that graceful restart does in the main router. The only difference is the location of the graceful-restart statement. For a logical router, include the graceful-restart statement at the [edit logical-routers *logical-router-name* routing-options] hierarchy level. For a routing instance inside a logical router, include the graceful-restart statement at both the [edit logical-routers *logical-router-name* routing-options] and [edit logical-routers *logical-router-name* routing-instances *instance-name* routing-options] hierarchy levels.

System Requirements

To implement graceful restart, your system must meet these minimum requirements:

- JUNOS Release 6.4 or later for PIM graceful restart
- JUNOS Release 5.6 or later for the CCC, TCC, Layer 2 VPN, or Layer 3 VPN implementations of graceful restart
- JUNOS Release 5.5 or later for RSVP or LDP graceful restart
- JUNOS Release 5.3 or later for aggregate route, BGP, IS-IS, OSPF, RIP, RIPng, or static route graceful restart
- Two or more Juniper Networks M-series or T-series routing platforms

Terms and Acronyms

circuit cross-connect (CCC)—A Juniper Networks method of exchanging frames between one router interface running a Layer 2 protocol and another router interface using the same Layer 2 protocol. For more information about CCC, see the *JUNOS Network Interfaces and Class of Service Configuration Guide* or the *JUNOS MPLS Applications Configuration Guide*.

translational cross-connect (TCC)—A Juniper Networks method of exchanging frames between two router interfaces running different Layer 2 protocols, such as ATM, Cisco HDLC, Ethernet, Ethernet Extended VLAN, Frame Relay, and PPP. For more information about TCC, see the *JUNOS Network Interfaces and Class of Service Configuration Guide* or the *JUNOS MPLS Applications Configuration Guide*.

restarting router—A router that experiences protocol restart because of a routing protocol process (rpd) failure or similar event. In the case of dynamic protocols such as BGP, IS-IS, OSPF, RIP, and RIPng, the neighboring routers must also support graceful restart functionality.

helper router—A router that peers with a restarting router and cooperates to restore connections and adjacencies with the restarting router.

Configuring Routing Protocol and Summaries Graceful Restart

To implement graceful restart for a routing protocol or summary, you must configure the following:

Configuring Graceful Restart for All Routing Protocols on page 456

Configuring Graceful Restart Options for BGP on page 457

Configuring Graceful Restart Options for IS-IS on page 457

Configuring Graceful Restart Options for OSPF on page 458

Configuring Graceful Restart Options for RIP and RIPng on page 458

Configuring Graceful Restart Options for PIM Sparse Mode on page 459

To apply your knowledge, visit these sections:

Example: Layer 3 VPN Graceful Restart Configuration on page 464

Checking Your Work on page 482

Configuring Graceful Restart for All Routing Protocols

You enable graceful restart functionality with a global statement that applies to all routing protocols, including BGP, IS-IS, OSPF, PIM sparse mode, RIP, and RIPng. Helper mode is enabled by default when graceful restart is enabled. To configure graceful restart, include the graceful-restart statement at the [edit routing-options] hierarchy level. To disable graceful restart globally, include the disable statement at the [edit routing-options graceful-restart] hierarchy level. To set the amount of time the router should wait after a restart before it selects routing paths, include the restart-duration statement at the [edit routing-options graceful-restart] hierarchy level.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
```

Once graceful restart is enabled for all routing protocols at the [edit routing-options graceful-restart] hierarchy level, you can disable graceful restart on a per-protocol basis.

NOTE: If you configure graceful restart after a BGP session has been established, the BGP session restarts and the peers negotiate graceful restart capabilities.

Configuring Graceful Restart Options for BGP

BGP graceful restart can be disabled on a particular neighbor in a group, an entire group, or the entire BGP protocol itself. To disable BGP graceful restart capability for all BGP sessions, include the disable statement at the [edit protocols bgp graceful-restart] hierarchy level. There is no helper mode capability for BGP.

You can also set the amount of time a router will send graceful restart messages to peer routers or receive these messages from restarting neighbors before declaring them down. To configure, include the restart-time and stale-routes-time statements at the [edit protocols bgp graceful-restart] hierarchy level. The default value for both timers is 5 minutes.

```
[edit]
protocols {
    bgp {
        graceful-restart {
            disable;
            restart-time seconds;
            stale-routes-time seconds;
        }
    }
}
```

NOTE: To set BGP graceful restart properties or disable them for a group, configure the desired statements at the [edit protocols bgp group *group-name* graceful-restart] hierarchy level.

To set BGP graceful restart properties or disable them for a specific neighbor in a group, configure the desired statements at the [edit protocols bgp group *group-name* neighbor *ip-address* graceful-restart] hierarchy level.

Configuring Graceful Restart Options for IS-IS

To disable IS-IS graceful restart capability, include the disable statement at the [edit protocols isis graceful-restart] hierarchy level. To disable IS-IS graceful restart helper capability, include the helper-disable statement at the [edit protocols isis graceful-restart] hierarchy level.

You can also set the amount of time a helper router waits for a neighbor to restart before declaring the restarting router down. To configure, include the restart-duration statement at the [edit protocols isis graceful-restart] hierarchy level. The default value for the restart-duration statement is 30 seconds, but you can configure a manual value from 30 to 300 seconds.

```
[edit]
protocols {
    isis {
        graceful-restart {
            disable;
            helper-disable;
            restart-duration seconds;
        }
    }
}
```

Configuring Graceful Restart Options for OSPF

To disable OSPF graceful restart capability, include the disable statement at the [edit protocols ospf graceful-restart] hierarchy level. To disable OSPF graceful restart helper capability, include the helper-disable statement at the [edit protocols ospf graceful-restart] hierarchy level.

You can also set the amount of time a router will send graceful restart LSAs to helper routers or receive these LSAs from restarting neighbors before declaring them down. To configure, include the notify-duration and restart-duration statements at the [edit protocols ospf graceful-restart] hierarchy level. The default values for these statements are 30 seconds for the notify duration and 180 seconds for the restart duration.

```
[edit]
protocols {
    ospf {
      graceful-restart {
        disable;
        helper-disable;
        notify-duration seconds;
        restart-duration seconds;
      }
    }
}
```

Configuring Graceful Restart Options for RIP and RIPng

To disable RIP or RIPng graceful restart capability, include the disable statement at the [edit protocols (rip | ripng) graceful-restart] hierarchy level. You can also set the amount of time a router waits for restarting neighbors before declaring them down. To configure, include the restart-time statement at the [edit protocols (rip | ripng) graceful-restart] hierarchy level.

```
[edit]
protocols {
    (rip | ripng) {
      graceful-restart {
        disable;
        restart-time seconds;
      }
    }
}
```

Configuring Graceful Restart Options for PIM Sparse Mode

PIM sparse mode continues to forward existing multicast packet streams during a graceful restart, but does not forward new streams until after the restart is complete. After a restart, the routing platform updates the forwarding state with any updates that were received from neighbors and occurred during the restart period. For example, the routing platform relearns the join and prune states of neighbors during the restart, but does not apply the changes to the forwarding table until after the restart.

In JUNOS Release 6.4 and later, M-series and T-series PIM sparse mode-enabled routing platforms generate a unique 32-bit random number called a generation identifier. Generation identifiers are included by default in PIM hello messages, as specified in the IETF Internet draft *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*. When a routing platform receives PIM hellos containing generation identifiers on a point-to-point interface, the JUNOS software activates an algorithm that optimizes graceful restart.

Before PIM sparse mode graceful restart occurs, each routing platform creates a generation identifier and sends it to its multicast neighbors. If a PIM sparse mode-enabled routing platform restarts, it creates a new generation identifier and sends it to its neighbors. When a neighbor receives the new identifier, it resends multicast updates to the restarting router to allow it to exit graceful restart efficiently. The restart phase completes when either the PIM state becomes stable or when the restart interval timer expires.

To enable PIM sparse mode graceful restart, include the graceful-restart statement at the global [edit routing-options] hierarchy level. To disable PIM sparse mode graceful restart capability, include the disable statement at the [edit protocols pim graceful-restart] hierarchy level.

If a routing platform does not support generation identifiers or if PIM is enabled on multipoint interfaces, the PIM sparse mode graceful restart algorithm does not activate and a default restart timer is used as the restart mechanism. To configure the maximum amount of time a routing platform waits before exiting PIM graceful restart, include the restart-duration statement at the [edit protocols pim graceful-restart] hierarchy level. The default value for the restart duration is 60 seconds, with a minimum of 30 seconds and a maximum of 300 seconds.

```
[edit]
protocols {
    pim {
        graceful-restart {
            disable;
            restart-duration seconds;
        }
    }
}
```

NOTE: Multicast forwarding can be interrupted in two ways. First, if the underlying routing protocol is unstable, multicast reverse-path-forwarding (RPF) checks can fail and cause an interruption. Second, because the forwarding table is not updated during the graceful restart period, new multicast streams are not forwarded until graceful restart is complete.

Configuring Graceful Restart for an MPLS-Related Protocol

To implement graceful restart for an MPLS-related protocol, you must configure the following:

Configuring Graceful Restart for all MPLS-Related Protocols on page 460

Configuring Graceful Restart Options for RSVP, CCC, and TCC on page 461

Configuring Graceful Restart Options for LDP on page 461

To apply your knowledge, visit these sections:

Example: Layer 3 VPN Graceful Restart Configuration on page 464

Checking Your Work on page 482

Configuring Graceful Restart for all MPLS-Related Protocols

You enable graceful restart functionality with a global statement that applies to all MPLS-related protocols, including CCC, LDP, RSVP, and TCC. Helper mode is enabled by default when graceful restart is enabled. To configure graceful restart, include the graceful-restart statement at the [edit routing-options] hierarchy level. To disable graceful restart globally, include the disable statement at the [edit routing-options graceful-restart] hierarchy level. To set the amount of time the router should wait after a restart before it selects routing paths, include the restart-duration at the [edit routing-options graceful-restart] hierarchy level.

```
[edit]
routing-options {
    graceful-restart {
      disable;
      restart-duration seconds;
    }
}
```

Once graceful restart is enabled for all MPLS-related protocols at the [edit routing-options graceful-restart] hierarchy level, you can disable graceful restart for most protocols on a per-protocol basis.

Configuring Graceful Restart Options for RSVP, CCC, and TCC

Because CCC and TCC rely on RSVP, you must modify these three protocols as a single group. Note that RSVP graceful restart is supported on ingress provider edge routers in JUNOS Release 6.1 and later.

To disable RSVP, CCC, and TCC graceful restart capabilities, include the disable statement at the [edit protocols rsvp graceful-restart] hierarchy level. To disable RSVP, CCC, and TCC graceful restart helper capability, include the helper-disable statement at the [edit protocols rsvp graceful-restart] hierarchy level.

```
[edit]
protocols {
    rsvp {
      graceful-restart {
        disable;
        helper-disable;
      }
    }
}
```

Configuring Graceful Restart Options for LDP

To disable LDP graceful restart capability, include the disable statement at the [edit protocols ldp graceful-restart] hierarchy level. To disable LDP graceful restart helper capability, include the helper-disable statement at the [edit protocols ldp graceful-restart] hierarchy level.

On the restarting router, you can configure a statement that suggests to the helper routers the amount of time they are required to maintain the old forwarding state during a restart. To configure, include the recovery-time statement at the [edit protocols ldp graceful-restart] hierarchy level.

On the helper router, you can configure a statement that overrides the request from the restarting router and sets the maximum amount of time the helper router will maintain the old forwarding state. To configure, include the maximum-recovery-time statement at the [edit protocols ldp graceful-restart] hierarchy level. The default value for both timers is 140 seconds.

```
[edit]
protocols {
    ldp {
      graceful-restart {
        disable;
        helper-disable;
        maximum-recovery-time seconds;
        recovery-time seconds;
      }
    }
}
```

Configuring VPN Graceful Restart

To implement graceful restart for a Layer 2 VPN, virtual private LAN service (VPLS) instance, or Layer 3 VPN, configure the following:

Configuring Graceful Restart for All Routing and MPLS-Related Protocols on page 462

Enabling Graceful Restart in the Routing Instance on page 463

To apply your knowledge, visit these sections:

Example: Layer 3 VPN Graceful Restart Configuration on page 464

Checking Your Work on page 482

Configuring Graceful Restart for All Routing and MPLS-Related Protocols

To configure graceful restart for all routing and MPLS-related protocols, include the graceful-restart statement at the [edit routing-options] hierarchy level. To disable graceful restart globally, include the disable statement at the [edit routing-options graceful-restart] hierarchy level. To set the amount of time the router should wait after a restart before it selects routing paths, include the restart-duration statement at the [edit routing-options graceful-restart] hierarchy level.

```
[edit]
routing-options {
    graceful-restart {
        disable;
        restart-duration seconds;
    }
}
```

Enabling Graceful Restart in the Routing Instance

For Layer 3 VPNs only, you must also configure graceful restart for all routing and MPLS-related protocols within a routing instance. To configure, include the graceful-restart statement at the [edit routing-instances *instance-name* routing-options] hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. Also, you can disable graceful restart for individual protocols with the disable statement at the [edit routing-instances *instance-name* protocols *protocol-name* graceful-restart] hierarchy level. For more information on the disable statement, see "Configuring Routing Protocol and Summaries Graceful Restart" on page 456.

```
[edit]
routing-instances {
    instance-name {
        routing-options {
            graceful-restart {
                disable;
                restart-duration seconds;
            }
        }
    }
}
```

Example: Layer 3 VPN Graceful Restart Configuration

Figure 43: Layer 3 VPN Graceful Restart Topology Diagram

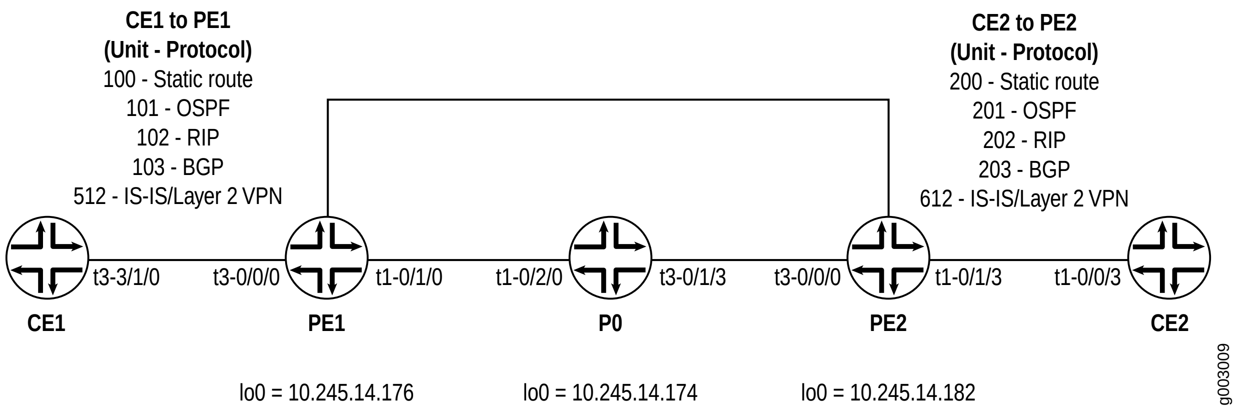

Figure 43 shows a standard MPLS VPN network. Routers CE1 and CE2 are customer edge routers, PE1 and PE2 are provider edge routers, and P0 is a provider core router. Several Layer 3 VPNs are configured across this network, as well as one Layer 2 VPN. Interfaces are shown in the diagram and are not included in the configuration example that follows.

On Router CE1, configure the following protocols on the logical interfaces of t3-3/1/0: OSPF on unit 101, RIP on unit 102, BGP on unit 103, and IS-IS on unit 512. Also configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on Router PE1.

Router CE1

```
[edit]
interfaces {
    t3-3/1/0 {
        encapsulation frame-relay;
        unit 100 {
            dlci 100;
            family inet {
                address 10.96.100.2/30;
            }
        }
        unit 101 {
            dlci 101;
            family inet {
                address 10.96.101.2/30;
            }
        }
        unit 102 {
            dlci 102;
            family inet {
                address 10.96.102.2/30;
            }
        }
```

```
        unit 103 {
            dlci 103;
            family inet {
                address 10.96.103.2/30;
            }
        }
        unit 512 {
            dlci 512;
            family inet {
                address 10.96.252.1/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.172/32;
                    primary;
                }
                address 10.96.110.1/32;
                address 10.96.111.1/32;
                address 10.96.112.1/32;
                address 10.96.113.1/32;
                address 10.96.116.1/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4172.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    autonomous-system 65100;
}
```

```
protocols {
    bgp {
        group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.103.1 {
                local-address 10.96.103.2;
                family inet {
                    unicast;
                }
                peer-as 65103;
            }
        }
    }
    isis {
        export ISIS_L2VPN_LB_DIRECT;
        interface t3-3/1/0.512;
    }
    ospf {
        export OSPF_LB_DIRECT;
        area 0.0.0.0 {
            interface t3-3/1/0.101;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t3-3/1/0.102;
        }
    }
}
```

```
policy-options {
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.101.0/30 exact;
                route-filter 10.96.111.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.102.0/30 exact;
                route-filter 10.96.112.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.103.0/30 exact;
                route-filter 10.96.113.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.116.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

On Router PE1, configure graceful restart in the master instance, along with BGP, OSPF, MPLS, and LDP. Next configure several protocol-specific instances of graceful restart. By including instances for BGP, OSPF, Layer 2 VPNs, RIP, and static routes, you can observe the wide range of options available to you when implementing graceful restart. Configure the following protocols in individual instances on the logical interfaces of t3-0/0/0: a static route on unit 100, OSPF on unit 101, RIP on unit 102, BGP on unit 103, and Frame Relay on unit 512 for the Layer 2 VPN instance.

Router PE1

```
[edit]
interfaces {
    t3-0/0/0 {
        dce;
        encapsulation frame-relay-ccc;
        unit 100 {
            dlci 100;
            family inet {
                address 10.96.100.1/30;
            }
            family mpls;
        }
        unit 101 {
            dlci 101;
            family inet {
                address 10.96.101.1/30;
            }
            family mpls;
        }
        unit 102 {
            dlci 102;
            family inet {
                address 10.96.102.1/30;
            }
            family mpls;
        }
        unit 103 {
            dlci 103;
            family inet {
                address 10.96.103.1/30;
            }
            family mpls;
        }
        unit 512 {
            encapsulation frame-relay-ccc;
            dlci 512;
        }
    }
    t1-0/1/0 {
        unit 0 {
            family inet {
                address 10.96.0.2/30;
            }
            family mpls;
        }
    }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.176/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4176.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    router-id 10.245.14.176;
    autonomous-system 69;
}
protocols {
    mpls {
        interface all;
    }
    bgp {
        group PEPE {
            type internal;
            neighbor 10.245.14.182 {
                local-address 10.245.14.176;
                family inet-vpn {
                    unicast;
                }
                family l2vpn {
                    unicast;
                }
            }
        }
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/1/0.0;
            interface fxp0.0 {
                disable;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface all;
    }
}
```

```
policy-options {
    policy-statement STATIC-import {
        from community STATIC;
        then accept;
    }
    policy-statement STATIC-export {
        then {
            community add STATIC;
            accept;
        }
    }
    policy-statement OSPF-import {
        from community OSPF;
        then accept;
    }
    policy-statement OSPF-export {
        then {
            community add OSPF;
            accept;
        }
    }
    policy-statement RIP-import {
        from community RIP;
        then accept;
    }
    policy-statement RIP-export {
        then {
            community add RIP;
            accept;
        }
    }
    policy-statement BGP-INET-import {
        from community BGP-INET;
        then accept;
    }
    policy-statement BGP-INET-export {
        then {
            community add BGP-INET;
            accept;
        }
    }
    policy-statement L2VPN-import {
        from community L2VPN;
        then accept;
    }
    policy-statement L2VPN-export {
        then {
            community add L2VPN;
            accept;
        }
    }
    community BGP-INET members target:69:103;
    community L2VPN members target:69:512;
    community OSPF members target:69:101;
    community RIP members target:69:102;
    community STATIC members target:69:100;
}
```

```
routing-instances {
    BGP-INET {
        instance-type vrf;
        interface t3-0/0/0.103;
        route-distinguisher 10.245.14.176:103;
        vrf-import BGP-INET-import;
        vrf-export BGP-INET-export;
        routing-options {
            graceful-restart;
            autonomous-system 65103;
        }
        protocols {
            bgp {
                group BGP-INET {
                    type external;
                    export BGP-INET-import;
                    neighbor 10.96.103.2 {
                        local-address 10.96.103.1;
                        family inet {
                            unicast;
                        }
                        peer-as 65100;
                    }
                }
            }
        }
    }
    L2VPN {
        instance-type l2vpn;
        interface t3-0/0/0.512;
        route-distinguisher 10.245.14.176:512;
        vrf-import L2VPN-import;
        vrf-export L2VPN-export;
        protocols {    # There is no graceful-restart statement for Layer 2 VPN instances.
            l2vpn {
                encapsulation-type frame-relay;
                site CE1-ISIS {
                    site-identifier 512;
                    interface t3-0/0/0.512 {
                        remote-site-id 612;
                    }
                }
            }
        }
    }
```

```
    OSPF {
        instance-type vrf;
        interface t3-0/0/0.101;
        route-distinguisher 10.245.14.176:101;
        vrf-import OSPF-import;
        vrf-export OSPF-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            ospf {
                export OSPF-import;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
    RIP {
        instance-type vrf;
        interface t3-0/0/0.102;
        route-distinguisher 10.245.14.176:102;
        vrf-import RIP-import;
        vrf-export RIP-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            rip {
                group RIP {
                    export RIP-import;
                    neighbor t3-0/0/0.102;
                }
            }
        }
    }
    STATIC {
        instance-type vrf;
        interface t3-0/0/0.100;
        route-distinguisher 10.245.14.176:100;
        vrf-import STATIC-import;
        vrf-export STATIC-export;
        routing-options {
            graceful-restart;
            static {
                route 10.96.110.1/32 next-hop t3-0/0/0.100;
            }
        }
    }
}
```

On Router P0, configure graceful restart in the main instance, along with OSPF, MPLS, and LDP. This allows the protocols on the PE routers to reach each other.

Router P0

```
[edit]
interfaces {
    t3-0/1/3 {
        unit 0 {
            family inet {
                address 10.96.0.5/30;
            }
            family mpls;
        }
    }
    t1-0/2/0 {
        unit 0 {
            family inet {
                address 10.96.0.1/30;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.174/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4174.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    router-id 10.245.14.174;
    autonomous-system 69;
}
protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface t1-0/2/0.0;
            interface t3-0/1/3.0;
            interface fxp0.0 {
                disable;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
    ldp {
        interface all;
    }
}
```

On Router PE2, configure BGP, OSPF, MPLS, LDP, and graceful restart in the master instance. Configure the following protocols in individual instances on the logical interfaces of t1-0/1/3: a static route on unit 200, OSPF on unit 201, RIP on unit 202, BGP on unit 203, and Frame Relay on unit 612 for the Layer 2 VPN instance. Also configure protocol-specific graceful restart in all routing instances, except the Layer 2 VPN instance.

Router PE2

```
[edit]
interfaces {
    t3-0/0/0 {
        unit 0 {
            family inet {
                address 10.96.0.6/30;
            }
            family mpls;
        }
    }
    t1-0/1/3 {
        dce;
        encapsulation frame-relay-ccc;
        unit 200 {
            dlci 200;
            family inet {
                address 10.96.200.1/30;
            }
            family mpls;
        }
        unit 201 {
            dlci 201;
            family inet {
                address 10.96.201.1/30;
            }
            family mpls;
        }
        unit 202 {
            dlci 202;
            family inet {
                address 10.96.202.1/30;
            }
            family mpls;
        }
        unit 203 {
            dlci 203;
            family inet {
                address 10.96.203.1/30;
            }
            family mpls;
        }
        unit 612 {
            encapsulation frame-relay-ccc;
            dlci 612;
        }
    }
```

```
        lo0 {
            unit 0 {
                family inet {
                    address 10.245.14.182/32;
                }
                family iso {
                    address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4182.00;
                }
            }
        }
}
routing-options {
        graceful-restart;
        router-id 10.245.14.182;
        autonomous-system 69;
}
protocols {
        mpls {
            interface all;
        }
        bgp {
            group PEPE {
                type internal;
                neighbor 10.245.14.176 {
                    local-address 10.245.14.182;
                    family inet-vpn {
                        unicast;
                    }
                    family l2vpn {
                        unicast;
                    }
                }
            }
        }
        ospf {
            area 0.0.0.0 {
                interface t3-0/0/0.0;
                interface fxp0.0 {
                    disable;
                }
                interface lo0.0 {
                    passive;
                }
            }
        }
        ldp {
            interface all;
        }
```

```
policy-options {
    policy-statement STATIC-import {
      from community STATIC;
      then accept;
    }
    policy-statement STATIC-export {
      then {
        community add STATIC;
        accept;
      }
    }
    policy-statement OSPF-import {
      from community OSPF;
      then accept;
    }
    policy-statement OSPF-export {
      then {
        community add OSPF;
        accept;
      }
    }
    policy-statement RIP-import {
      from community RIP;
      then accept;
    }
    policy-statement RIP-export {
      then {
        community add RIP;
        accept;
      }
    }
    policy-statement BGP-INET-import {
      from community BGP-INET;
      then accept;
    }
    policy-statement BGP-INET-export {
      then {
        community add BGP-INET;
        accept;
      }
    }
    policy-statement L2VPN-import {
      from community L2VPN;
      then accept;
    }
    policy-statement L2VPN-export {
      then {
        community add L2VPN;
        accept;
      }
    }
    community BGP-INET members target:69:103;
    community L2VPN members target:69:512;
    community OSPF members target:69:101;
    community RIP members target:69:102;
    community STATIC members target:69:100;
}
```

```
routing-instances {
    BGP-INET {
        instance-type vrf;
        interface t1-0/1/3.203;
        route-distinguisher 10.245.14.182:203;
        vrf-import BGP-INET-import;
        vrf-export BGP-INET-export;
        routing-options {
            graceful-restart;
            autonomous-system 65203;
        }
        protocols {
            bgp {
                group BGP-INET {
                    type external;
                    export BGP-INET-import;
                    neighbor 10.96.203.2 {
                        local-address 10.96.203.1;
                        family inet {
                            unicast;
                        }
                        peer-as 65200;
                    }
                }
            }
        }
    }
    L2VPN {
        instance-type l2vpn;
        interface t1-0/1/3.612;
        route-distinguisher 10.245.14.182:612;
        vrf-import L2VPN-import;
        vrf-export L2VPN-export;
        protocols {    # There is no graceful-restart statement for Layer 2 VPN instances.
            l2vpn {
                encapsulation-type frame-relay;
                site CE2-ISIS {
                    site-identifier 612;
                    interface t1-0/1/3.612 {
                        remote-site-id 512;
                    }
                }
            }
        }
    }
```

```
    OSPF {
        instance-type vrf;
        interface t1-0/1/3.201;
        route-distinguisher 10.245.14.182:201;
        vrf-import OSPF-import;
        vrf-export OSPF-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            ospf {
                export OSPF-import;
                area 0.0.0.0 {
                    interface all;
                }
            }
        }
    }
    RIP {
        instance-type vrf;
        interface t1-0/1/3.202;
        route-distinguisher 10.245.14.182:202;
        vrf-import RIP-import;
        vrf-export RIP-export;
        routing-options {
            graceful-restart;
        }
        protocols {
            rip {
                group RIP {
                    export RIP-import;
                    neighbor t1-0/1/3.202;
                }
            }
        }
    }
    STATIC {
        instance-type vrf;
        interface t1-0/1/3.200;
        route-distinguisher 10.245.14.182:200;
        vrf-import STATIC-import;
        vrf-export STATIC-export;
        routing-options {
            graceful-restart;
            static {
                route 10.96.210.1/32 next-hop t1-0/1/3.200;
            }
        }
    }
}
```

On Router CE2, complete the Layer 2 and Layer 3 VPN configuration by mirroring the protocols already set on PE2 and CE1. Specifically, configure the following on the logical interfaces of t1-0/0/3: OSPF on unit 201, RIP on unit 202, BGP on unit 203, and IS-IS on unit 612. Finally, configure graceful restart, BGP, IS-IS, OSPF, and RIP on the main instance to be able to connect to the routing instances on PE2.

Router CE2

```
[edit]
interfaces {
    t1-0/0/3 {
        encapsulation frame-relay;
        unit 200 {
            dlci 200;
            family inet {
                address 10.96.200.2/30;
            }
        }
        unit 201 {
            dlci 201;
            family inet {
                address 10.96.201.2/30;
            }
        }
        unit 202 {
            dlci 202;
            family inet {
                address 10.96.202.2/30;
            }
        }
        unit 203 {
            dlci 203;
            family inet {
                address 10.96.203.2/30;
            }
        }
        unit 512 {
            dlci 512;
            family inet {
                address 10.96.252.2/30;
            }
        }
    }
```

```
    lo0 {
        unit 0 {
            family inet {
                address 10.245.14.180/32 {
                    primary;
                }
                address 10.96.210.1/32;
                address 10.96.111.1/32;
                address 10.96.212.1/32;
                address 10.96.213.1/32;
                address 10.96.216.1/32;
            }
            family iso {
                address 47.0005.80ff.f800.0000.0108.0001.0102.4501.4180.00;
            }
        }
    }
}
routing-options {
    graceful-restart;
    autonomous-system 65200;
}
protocols {
    bgp {
        group CE-PE-INET {
            type external;
            export BGP_INET_LB_DIRECT;
            neighbor 10.96.203.1 {
                local-address 10.96.203.2;
                family inet {
                    unicast;
                }
                peer-as 65203;
            }
        }
    }
    isis {
        export ISIS_L2VPN_LB_DIRECT;
        interface t1-0/0/3.612;
    }
    ospf {
        export OSPF_LB_DIRECT;
        area 0.0.0.0 {
            interface t1-0/0/3.201;
        }
    }
    rip {
        group RIP {
            export RIP_LB_DIRECT;
            neighbor t1-0/0/3.202;
        }
    }
}
```

```
policy-options {
    policy-statement OSPF_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.201.0/30 exact;
                route-filter 10.96.211.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement RIP_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.202.0/30 exact;
                route-filter 10.96.212.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement BGP_INET_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.203.0/30 exact;
                route-filter 10.96.213.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
    policy-statement ISIS_L2VPN_LB_DIRECT {
        term direct {
            from {
                protocol direct;
                route-filter 10.96.216.1/32 exact;
            }
            then accept;
        }
        term final {
            then reject;
        }
    }
}
```

Checking Your Work

To verify proper operation of graceful restart, use the following commands:

- show bgp neighbor (for BGP graceful restart)
- traceoptions (for OSPF graceful restart)
- show rsvp neighbor detail (for RSVP graceful restart—helper router)
- show rsvp version (for RSVP graceful restart—restarting router)
- show ldp session detail (for LDP graceful restart)
- show connections (for CCC and TCC graceful restart)
- show route instance detail (for Layer 3 VPN graceful restart and for any protocols using graceful restart in a routing instance)
- show route protocol l2vpn (for Layer 2 VPN graceful restart)

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

- Router PE1 Status Prior to a Restart on page 483
- Router PE1 Status During a Restart on page 488

The remaining sections display output of some of these commands generated from other examples:

- Verifying OSPF Graceful Restart on page 493
- Verifying BGP Graceful Restart on page 493
- Verifying CCC and TCC Graceful Restart on page 494

Router PE1 Status Prior to a Restart

Before a restart happens, check your neighbor relationships on Router PE1:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2+3785 AS 65100 Local: 10.96.103.1+179 AS 65103
  Type: External    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.96.110.1      Local ID: 10.96.103.1      Active Holdtime: 90
  Keepalive Interval: 30
  Local Interface: t3-0/0/0.103
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-unicast
  NLRI peer can save forwarding state: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table BGP-INET.inet.0 Bit: 30001
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:            0
    Received prefixes:          0
    Suppressed due to damping:  0
  Last traffic (seconds): Received 8    Sent 3    Checked 3
  Input messages:  Total 15     Updates 0       Refreshes 0     Octets 321
  Output messages: Total 18     Updates 2       Refreshes 0     Octets 450
  Output Queue[2]: 0

Peer: 10.245.14.182+4701 AS 69    Local: 10.245.14.176+179 AS 69
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily
Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 1
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
```

```
NLRI of all end-of-rib markers sent: inet-vpn-unicast l2vpn
Table bgp.l3vpn.0 Bit: 10000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Suppressed due to damping:  0
Table bgp.l2vpn.0 Bit: 20000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        1
  Suppressed due to damping:  0
Table BGP-INET.inet.0 Bit: 30000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Suppressed due to damping:  0
Table OSPF.inet.0 Bit: 60000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Suppressed due to damping:  0
Table RIP.inet.0 Bit: 70000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Suppressed due to damping:  0
Table STATIC.inet.0 Bit: 80000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          0
  Received prefixes:        0
  Suppressed due to damping:  0
Table L2VPN.l2vpn.0 Bit: 90000
  RIB State: BGP restart is complete
  RIB State: VPN restart is complete
  Send state: in sync
  Active prefixes:          1
  Received prefixes:        1
  Suppressed due to damping:  0
Last traffic (seconds): Received 28   Sent 28   Checked 28
Input messages:  Total 2      Updates 0       Refreshes 0     Octets 86
Output messages: Total 13     Updates 10      Refreshes 0     Octets 1073
Output Queue[0]: 0
Output Queue[1]: 0
Output Queue[2]: 0
Output Queue[3]: 0
Output Queue[4]: 0
Output Queue[5]: 0
Output Queue[6]: 0
Output Queue[7]: 0
Output Queue[8]: 0
```

```
user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding      State: Active
  Restart State: Complete Path selection timeout: 300
  Tables:
    inet.0              : 17 routes (15 active, 0 holddown, 1 hidden)
    Restart Complete
    inet.3              : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    iso.0               : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0              : 19 routes (19 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l3vpn.0         : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Complete
    inet6.0             : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0         : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf             State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0     : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn           State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0       : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
OSPF:
  Router ID: 10.96.101.1
  Type: vrf             State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0         : 8 routes (7 active, 0 holddown, 0 hidden)
    Restart Complete
RIP:
  Router ID: 10.96.102.1
  Type: vrf             State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.245.14.176:102
```

```
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0           : 6 routes (6 active, 0 holddown, 0 hidden)
    Restart Complete
STATIC:
  Router ID: 10.96.100.1
  Type: vrf              State: Active
  Restart State: Complete Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0        : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Complete
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding       State: Active

user@PE1> show route protocol l2vpn

inet.0: 16 destinations, 17 routes (15 active, 0 holddown, 1 hidden)
Restart Complete

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Complete

OSPF.inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
Restart Complete

RIP.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
Restart Complete

STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

800003             *[L2VPN/7] 00:06:00
                    > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512       *[L2VPN/7] 00:06:00
                    > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4

bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Complete

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
```

```
10.245.14.176:512:512:611/96
                *[L2VPN/7] 00:06:01
                  Discard

bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete
```

Router PE1 Status During a Restart

Before you can verify that graceful restart is working, you must simulate a router restart. To cause the routing process to refresh and simulate a restart, use the restart routing operational mode command:

```
user@PE1> restart routing
Routing protocol daemon started, pid 3558
```

Next is some output captured during the router restart:

```
user@PE1> show bgp neighbor
Peer: 10.96.103.2     AS 65100 Local: 10.96.103.1     AS 65103
  Type: External    State: Active        Flags: <ImportEval>
  Last State: Idle         Last Event: Start
  Last Error: None
  Export: [ BGP-INET-import ]
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast
  Local Address: 10.96.103.1 Holdtime: 90 Preference: 170
  Number of flaps: 0

Peer: 10.245.14.182+179 AS 69    Local: 10.245.14.176+2131 AS 69
  Type: Internal    State: Established    Flags: <ImportEval>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress HoldTime GracefulRestart AddressFamily Rib-group Refresh>
  Address families configured: inet-vpn-unicast l2vpn
  Local Address: 10.245.14.176 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 10.245.14.182    Local ID: 10.245.14.176    Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-vpn-unicast l2vpn
  NLRI advertised by peer: inet-vpn-unicast l2vpn
  NLRI for this session: inet-vpn-unicast l2vpn
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
  NLRI that peer supports restart for: inet-vpn-unicast l2vpn
  NLRI peer can save forwarding state: inet-vpn-unicast l2vpn
  NLRI that peer saved forwarding for: inet-vpn-unicast l2vpn
  NLRI that restart is negotiated for: inet-vpn-unicast l2vpn
  NLRI of received end-of-rib markers: inet-vpn-unicast l2vpn
  Table bgp.l3vpn.0 Bit: 10000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
    Active prefixes:           10
    Received prefixes:         10
    Suppressed due to damping:  0
  Table bgp.l2vpn.0 Bit: 20000
    RIB State: BGP restart in progress
    RIB State: VPN restart in progress
    Send state: in sync
```

```
      Active prefixes:           1
      Received prefixes:         1
      Suppressed due to damping:  0
    Table BGP-INET.inet.0 Bit: 30000
      RIB State: BGP restart in progress
      RIB State: VPN restart in progress
      Send state: in sync
      Active prefixes:           2
      Received prefixes:         2
      Suppressed due to damping:  0
  Table OSPF.inet.0 Bit: 60000
      RIB State: BGP restart is complete
      RIB State: VPN restart in progress
      Send state: in sync
      Active prefixes:           2
      Received prefixes:         2
      Suppressed due to damping:  0
    Table RIP.inet.0 Bit: 70000
      RIB State: BGP restart is complete
      RIB State: VPN restart in progress
      Send state: in sync
      Active prefixes:           2
      Received prefixes:         2
      Suppressed due to damping:  0
    Table STATIC.inet.0 Bit: 80000
      RIB State: BGP restart is complete
      RIB State: VPN restart in progress
      Send state: in sync
      Active prefixes:           1
      Received prefixes:         1
      Suppressed due to damping:  0
    Table L2VPN.l2vpn.0 Bit: 90000
      RIB State: BGP restart is complete
      RIB State: VPN restart in progress
      Send state: in sync
      Active prefixes:           1
      Received prefixes:         1
      Suppressed due to damping:  0
    Last traffic (seconds): Received 0    Sent 0    Checked 0
    Input messages:  Total 14     Updates 13      Refreshes 0     Octets 1053
    Output messages: Total 3      Updates 0       Refreshes 0     Octets 105
    Output Queue[0]: 0
    Output Queue[1]: 0
    Output Queue[2]: 0
    Output Queue[3]: 0
    Output Queue[4]: 0
    Output Queue[5]: 0
    Output Queue[6]: 0
    Output Queue[7]: 0
    Output Queue[8]: 0
```

```
user@PE1> show route instance detail
master:
  Router ID: 10.245.14.176
  Type: forwarding        State: Active
  Restart State: Pending  Path selection timeout: 300
  Tables:
    inet.0                 : 17 routes (15 active, 1 holddown, 1 hidden)
    Restart Pending: OSPF LDP
    inet.3                 : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: OSPF LDP
    iso.0                  : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Complete
    mpls.0                 : 23 routes (23 active, 0 holddown, 0 hidden)
    Restart Pending: LDP VPN
    bgp.l3vpn.0            : 10 routes (10 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
    inet6.0                : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Complete
    bgp.l2vpn.0            : 1 routes (1 active, 0 holddown, 0 hidden)
    Restart Pending: BGP VPN
BGP-INET:
  Router ID: 10.96.103.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.103
  Route-distinguisher: 10.245.14.176:103
  Vrf-import: [ BGP-INET-import ]
  Vrf-export: [ BGP-INET-export ]
  Tables:
    BGP-INET.inet.0        : 6 routes (5 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
L2VPN:
  Router ID: 0.0.0.0
  Type: l2vpn             State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.512
  Route-distinguisher: 10.245.14.176:512
  Vrf-import: [ L2VPN-import ]
  Vrf-export: [ L2VPN-export ]
  Tables:
    L2VPN.l2vpn.0          : 2 routes (2 active, 0 holddown, 0 hidden)
    Restart Pending: VPN L2VPN
OSPF:
  Router ID: 10.96.101.1
  Type: vrf               State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.101
  Route-distinguisher: 10.245.14.176:101
  Vrf-import: [ OSPF-import ]
  Vrf-export: [ OSPF-export ]
  Tables:
    OSPF.inet.0            : 8 routes (7 active, 1 holddown, 0 hidden)
    Restart Pending: OSPF VPN
```

```
RIP:
  Router ID: 10.96.102.1
  Type: vrf            State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.102
  Route-distinguisher: 10.245.14.176:102
  Vrf-import: [ RIP-import ]
  Vrf-export: [ RIP-export ]
  Tables:
    RIP.inet.0           : 8 routes (6 active, 2 holddown, 0 hidden)
    Restart Pending: RIP VPN
STATIC:
  Router ID: 10.96.100.1
  Type: vrf            State: Active
  Restart State: Pending  Path selection timeout: 300
  Interfaces:
    t3-0/0/0.100
  Route-distinguisher: 10.245.14.176:100
  Vrf-import: [ STATIC-import ]
  Vrf-export: [ STATIC-export ]
  Tables:
    STATIC.inet.0        : 4 routes (4 active, 0 holddown, 0 hidden)
    Restart Pending: VPN
__juniper_private1__:
  Router ID: 0.0.0.0
  Type: forwarding       State: Active
```

You might notice there was no significant difference between the first set of pre-restart output and the second set of mid-restart output. To verify the restart, you need more details.

```
user@PE1> show route instance summary
Instance         Type         Primary rib          Active/holddown/hidden
master           forwarding
                              inet.0               15/0/1
                              iso.0                1/0/0
                              mpls.0               35/0/0
                              l3vpn.0              0/0/0
                              inet6.0              2/0/0
                              l2vpn.0              0/0/0
                              l2circuit.0          0/0/0
BGP-INET         vrf
                              BGP-INET.inet.0      5/0/0
                              BGP-INET.iso.0       0/0/0
                              BGP-INET.inet6.0     0/0/0
L2VPN            l2vpn
                              L2VPN.inet.0         0/0/0
                              L2VPN.iso.0          0/0/0
                              L2VPN.inet6.0        0/0/0
                              L2VPN.l2vpn.0        2/0/0
OSPF             vrf
                              OSPF.inet.0          7/0/0
                              OSPF.iso.0           0/0/0
                              OSPF.inet6.0         0/0/0
RIP              vrf
                              RIP.inet.0           6/0/0
                              RIP.iso.0            0/0/0
                              RIP.inet6.0          0/0/0
```

```
STATIC           vrf
                          STATIC.inet.0         4/0/0
                          STATIC.iso.0          0/0/0
                          STATIC.inet6.0        0/0/0
__juniper_private1__ forwarding
                          __juniper_priva.inet.0 0/0/0
                          __juniper_privat.iso.0 0/0/0
                          __juniper_priv.inet6.0 0/0/0
```

user@PE1> **show route protocol l2vpn**

```
inet.0: 16 destinations, 17 routes (15 active, 1 holddown, 1 hidden)
Restart Pending: OSPF LDP

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: OSPF LDP

BGP-INET.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
Restart Pending: VPN

OSPF.inet.0: 7 destinations, 8 routes (7 active, 1 holddown, 0 hidden)
Restart Pending: OSPF VPN

RIP.inet.0: 6 destinations, 8 routes (6 active, 2 holddown, 0 hidden)
Restart Pending: RIP VPN

STATIC.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Pending: VPN

iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

mpls.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
Restart Pending: LDP VPN
+ = Active Route, - = Last Active, * = Both

800001             *[L2VPN/7] 00:00:13
                  > via t3-0/0/0.512, Pop       Offset: 4
t3-0/0/0.512       *[L2VPN/7] 00:00:13
                  > via t1-0/1/0.0, Push 800003, Push 100004(top) Offset: -4

bgp.l3vpn.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

L2VPN.l2vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Pending: VPN L2VPN
+ = Active Route, - = Last Active, * = Both

10.245.14.176:512:512:611/96
                  *[L2VPN/7] 00:00:13
                    Discard

bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Pending: BGP VPN
```

Verifying OSPF Graceful Restart

To view graceful restart information for OSPF, configure OSPF traceoptions. You can configure traceoptions for OSPF with the file and flag statements at the [edit protocols ospf traceoptions] hierarchy level.

Here is the output of a traceoptions log from an OSPF restarting router:

```
Oct  8 05:20:12 Restart mode - sending grace lsas
Oct  8 05:20:12 Restart mode - estimated restart duration timer triggered
Oct  8 05:20:13 Restart mode - Sending more grace lsas
```

Here is the output of a traceoptions log from an OSPF helper router:

```
Oct  8 05:20:14 Helper mode for neighbor 192.255.5.1
Oct  8 05:20:14  Received multiple grace lsa from 192.255.5.1
```

Verifying BGP Graceful Restart

To view graceful restart information for BGP sessions, use the show bgp neighbor command:

```
user@PE1> show bgp neighbor 192.255.10.1
Peer: 192.255.10.1+179 AS 64595 Local: 192.255.5.1+1106 AS 64595
  Type: Internal    State: Established    Flags: <>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ static ]
  Options: <Preference LocalAddress HoldTime GracefulRestart Damping PeerAS
Refresh>
  Local Address: 192.255.5.1 Holdtime: 90 Preference: 170
  IPSec SA Name: hope
  Number of flaps: 0
  Peer ID: 192.255.10.1     Local ID: 192.255.5.1      Active Holdtime: 90
  Keepalive Interval: 30
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 180
  Stale routes from peer are kept for: 180
  Restart time requested by this peer: 300
  NLRI that peer supports restart for: inet-unicast
  NLRI that peer saved forwarding for: inet-unicast
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Table inet.0 Bit: 10000
    RIB State: restart is complete
    Send state: in sync
    Active prefixes: 0
    Received prefixes: 0
    Suppressed due to damping: 0
  Last traffic (seconds): Received 19   Sent 19   Checked 19
  Input messages:  Total 2      Updates 1       Refreshes 0     Octets 42
  Output messages: Total 3      Updates 0       Refreshes 0     Octets 116
  Output Queue[0]: 0
```

Verifying CCC and TCC Graceful Restart

To view graceful restart information for CCC and TCC connections, use the show connections command. This sample assumes four remote interface CCC connections between CE1 and CE2:

```
user@PE1> show connections
CCC and TCC connections [Link Monitoring On]
  Legend for status (St)            Legend for connection types
  UN -- uninitialized               if-sw:  interface switching
  NP -- not present                 rmt-if: remote interface switching
  WE -- wrong encapsulation         lsp-sw: LSP switching
  DS -- disabled
  Dn -- down                        Legend for circuit types
  -> -- only outbound conn is up    intf -- interface
  <- -- only inbound  conn is up    tlsp -- transmit LSP
  Up -- operational                 rlsp -- receive LSP
  RmtDn -- remote CCC down
  Restart -- restarting

CCC Graceful restart : Restarting

Connection/Circuit              Type    St      Time last up     # Up trans
CE1-CE2-0                       rmt-if  Restart -----                0
  fe-1/1/0.0                      intf  Up
  PE1-PE2-0                       tlsp  Up
  PE2-PE1-0                       rlsp  Up
CE1-CE2-1                       rmt-if  Restart -----                0
  fe-1/1/0.1                      intf  Up
  PE1-PE2-1                       tlsp  Up
  PE2-PE1-1                       rlsp  Up
CE1-CE2-2                       rmt-if  Restart -----                0
  fe-1/1/0.2                      intf  Up
  PE1-PE2-2                       tlsp  Up
  PE2-PE1-2                       rlsp  Up
CE1-CE2-3                       rmt-if  Restart -----                0
  fe-1/1/0.3                      intf  Up
  PE1-PE2-3                       tlsp  Up
  PE2-PE1-3                       rlsp  Up
```

Configuring Logical Router Graceful Restart

Graceful restart for a logical router functions much in the same way that graceful restart does in the main router. The only difference is the location of the graceful-restart statement.

To implement graceful restart in a logical router, configure the following:

Configuring Graceful Restart for All Routing and MPLS-Related Protocols in a Logical Router on page 495

Enabling Graceful Restart in the Routing Instance in a Logical Router on page 496

Option: Enabling Graceful Routing Engine Switchover on page 496

Configuring Graceful Restart for All Routing and MPLS-Related Protocols in a Logical Router

To configure graceful restart for all routing and MPLS-related protocols in a logical router, include the graceful-restart statement at the [edit logical-routers *logical-router-name* routing-options] hierarchy level. To disable graceful restart globally, include the disable statement at the [edit logical-routers *logical-router-name* routing-options graceful-restart] hierarchy level. To set the amount of time the router should wait after a restart before it selects routing paths, include the restart-duration statement at the [edit logical-routers *logical-router-name* routing-options graceful-restart] hierarchy level.

```
[edit]
logical-routers {
    logical-router-name {
        routing-options {
            graceful-restart {
                disable;
                restart-duration seconds;
            }
        }
    }
}
```

Enabling Graceful Restart in the Routing Instance in a Logical Router

For Layer 3 VPNs only, you must also configure graceful restart for all routing and MPLS-related protocols within a routing instance inside a logical router. To configure, include the graceful-restart statement at the [edit logical-routers *logical-router-name* routing-instances *instance-name* routing-options] hierarchy level. Because you can configure multi-instance BGP and multi-instance LDP, graceful restart for a carrier-of-carriers scenario is supported. Also, you can disable graceful restart for individual protocols with the disable statement at the [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols *protocol-name* graceful-restart] hierarchy level. For more information on the disable statement, see "Configuring Routing Protocol and Summaries Graceful Restart" on page 456.

```
[edit]
logical-routers {
    logical-router-name {
        routing-instances {
            instance-name {
                routing-options {
                    graceful-restart {
                        disable;
                        restart-duration seconds;
                    }
                }
            }
        }
    }
}
```

Option: Enabling Graceful Routing Engine Switchover

If you wish to implement graceful Routing Engine switchover for a logical router, graceful restart must be configured within the logical router. To configure graceful restart for the logical router, see "Configuring Graceful Restart for All Routing and MPLS-Related Protocols in a Logical Router" on page 495 and "Enabling Graceful Restart in the Routing Instance in a Logical Router" on page 496. To configure graceful Routing Engine switchover, include the graceful-switchover statement at the [edit chassis redundancy] hierarchy level.

```
[edit]
chassis {
    redundancy {
        graceful-switchover {
            enable;
        }
    }
}
```

For more information about graceful Routing Engine switchover, see the *JUNOS System Basics Configur ation Guide* .

For More Information

To view additional information about graceful restart, see the following:

JUNOS Routing Protocols Configuration Guide

JUNOS MPLS Applications Configuration Guide

JUNOS VPNs Configuration Guide

R. Braden, Editor, *Resource ReSerVation Protocol (RSVP)*, RFC 2205, September 1997

L. Berger, editor, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*, RFC 3473, January 2003. (only Section 9, "Fault Handling")

M. Leelanivas, et. al., *Graceful Restart Mechanism for LDP*, RFC 3478, February 2003

J. Moy, et. al., *Graceful OSPF Restart*, RFC 3623, November 2003

M. Shand, *Restart signaling for IS-IS*, RFC 3847, July 2004

S. Sangli, et. al., Internet draft draft-ietf-idr-restart-10.txt, *Graceful Restart Mechanism for BGP* (expires December 2004)

Y. Rekhter, et. al., Internet draft draft-ietf-mpls-bgp-mpls-restart-03.txt, *Graceful Restart Mechanism for BGP with MPLS* (expires August 2004)

B. Fenner, et. al., Internet draft draft-ietf-pim-sm-v2-new-10.txt, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)* (expires January 2005)

Revision History

2 February 2005—7.1R1 Release. Richard Hendricks.

6 October 2004—7.0R1 Release. Richard Hendricks.

6 July 2004—Added PIM graceful restart, 6.4R1 Release. Richard Hendricks.

5 April 2004—Added information about graceful restart in a logical router, 6.3R1 Release. Richard Hendricks.

22 December 2003—6.2R1 Release. Richard Hendricks.

22 September 2003—Added support for RSVP ingress graceful restart, 6.1R1 Release. Richard Hendricks.

30 June 2003—6.0R1 Release. Richard Hendricks.

2 April 2003—5.7R1 Release. Richard Hendricks.

27 December 2002—5.6R1 Release. Richard Hendricks.

22 October 2002—Added LDP, RSVP, CCC, TCC, Layer 2 VPN, and Layer 3 VPN graceful restart information and reformatted the document as a Feature Guide for JUNOS Release 5.6. Richard Hendricks.

2002 February 17—Initial Quick Start Guide document written. Tony Sinopoli.