

Chapter 2

VPN Configuration Guidelines

Layer 2 virtual private networks (VPNs), Layer 3 VPNs, and virtual private LAN service (VPLS) use a common infrastructure within JUNOS and common configuration procedures. This chapter describes the configuration steps that are common to Layer 2 VPNs, Layer 3 VPNs, and VPLS. It is best to complete the configuration steps outlined in this chapter regardless of which type of VPN you are configuring before proceeding to the more specific configuration steps described elsewhere in this manual.

The following sections describe the general procedures you need to follow to configure Layer 2 VPNs, Layer 3 VPNs, and VPLS:

- Enable a Signaling Protocol on the PE Routers

- Configure an IGP on the PE and Provider Routers on page 13

- Configure an IBGP Session between PE Routers on page 14

- Configure a VPN Routing Instance on the PE Routers

- Configure a Virtual-Router Routing Instance on page 25

- Configure Graceful Restart on page 27

- Rewrite Markers and VPNs

For information on the configuration procedures specific to Layer 2 VPNs, Layer 3 VPNs, and VPLS, see the following configuration chapters:

- Layer 2 VPN Configuration Guidelines on page 39

- Layer 3 VPN Configuration Guidelines on page 101

- VPLS Configuration Guidelines on page 283

Enable a Signaling Protocol on the PE Routers

For VPNs to function, you must enable a signaling protocol on the PE routers. You can do one of the following:

Use LDP for VPN Signaling on page 10

Use RSVP for VPN Signaling on page 12



Note

As with any configuration involving MPLS, you cannot configure any of the core-facing interfaces on the PE routers over Fast Ethernet PICs.

Use LDP for VPN Signaling

To use Label Distribution Protocol (LDP) for VPN signaling, perform the following steps on the PE and provider routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the `ldp` statement at the [edit protocols] hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and provider routers. You can think of these as the "core-facing" interfaces. You do not need to configure LDP on the interface between the PE and CE routers.

```
[edit]
protocols {
  ldp {
    interface type-fpc/pic/port;
  }
}
```

2. Configure the Multiprotocol Label Switching (MPLS) address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1) by including the `family mpls` statement at the [edit interfaces *type-fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit]
interfaces {
  type-fpc/pic/port {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

3. Configure Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) on each PE and provider router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

To configure OSPF, include the `ospf` statement at the [edit protocols] hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface type-fpc/pic/port;
    }
  }
}
```

To configure IS-IS, include the `isis` statement at the [edit protocols] hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the [edit interfaces] hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, lo0), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the address statement, *address* is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For more information about configuring OSPF and IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Use RSVP for VPN Signaling

To use the Resource Reservation Protocol (RSVP) for VPN signaling, perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an interior gateway protocol (IGP) that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

To enable OSPF traffic engineering support, include the traffic-engineering statement at the [edit protocols ospf] hierarchy level:

```
[edit protocols ospf]
traffic-engineering {
  shortcuts;
}
```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and provider router, enable RSVP on the router interfaces that participate in the label-switched path (LSP). On the PE router, these interfaces are the ingress and egress points to the LSP. On the provider router, these interfaces connect the LSP between the PE routers. Do not enable RSVP on the interface between the PE and the CE routers, because this interface is not part of the LSP.

To configure RSVP on the PE and provider routers, include the interface statement at the [edit protocols rsvp] hierarchy level. Include one interface statement for each interface on which you are enabling RSVP.

```
[edit protocols]
rsvp {
  interface interface-name;
  interface interface-name;
}
```

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the label-switched-path and interface statements at the [edit protocols mpls] hierarchy level.

```
[edit protocols]
mpls {
  label-switched-path path-name {
    to ip-address;
  }
  interface interface-name;
}
```

In the to statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

In the interface statement, specify the name of the interface (both the physical and logical portions). Include one interface statement for the interface associated with the LSP.

When you configure the logical portion of the same interface at the [edit interfaces] hierarchy level, you must also configure the family mpls and family inet statements:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

4. On all provider routers that participate in the LSP, enable MPLS by including the interface statement at the [edit mpls] hierarchy level. Include one interface statement for each connection to the LSP.

```
[edit]
mpls {
  interface interface-name;
  interface interface-name;
}
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

```
[edit]
mpls {
  interface interface-name;
}
```

For information about configuring MPLS, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

Configure an IGP on the PE and Provider Routers

For Layer 2 VPNs, Layer 3 VPNs, and VPLS to function properly, the service provider's provider edge (PE) and provider (P) routers need to be able to exchange routing information. To allow them to do this, you must configure either an interior gateway protocol (IGP) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process at the [edit protocols] hierarchy level, not within the routing instance used for the VPN—that is, not at the [edit routing-instances] hierarchy level.

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

For information about configuring IGPs and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure an IBGP Session between PE Routers

You must configure an IBGP session between the PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites.

Configure an IBGP session for the VPN at the [edit protocols bgp group *group-name*] hierarchy level as follows:

```
[edit protocols]
  bgp {
    group group-name {
      type internal;
      local-address ip-address;
      family family-type {
        unicast;
      }
      neighbor ip-address;
    }
  }
```

The IP address in the local-address statement is the address of the loopback interface (lo0) on the local PE router. The IBGP session for the VPN runs through the loopback address. (You must also configure the lo0 interface at the [edit interfaces] hierarchy level.)

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path *lsp-path-name*] hierarchy level when you configure the MPLS LSP.

The family statement allows you to configure the IBGP session for either Layer 2 VPNs and VPLS or for Layer 3 VPNs. To configure the IBGP session for Layer 2 VPNs and VPLS, configure the family statement at the [edit protocols bgp group *group-name*] hierarchy level as follows:

```
[edit protocols bgp group group-name]
  family l2vpn {
    unicast;
  }
```

To configure the IBGP session for Layer 3 VPNs, configure the family statement at the [edit protocols bgp group *group-name*] hierarchy level as follows:

```
[edit protocols bgp group group-name]
  family inet-vpn {
    unicast;
  }
```

Configure a VPN Routing Instance on the PE Routers

You need to configure a routing instance for each VPN on each of the PE routers participating in the VPN. The configuration procedures outlined in this section are applicable to Layer 2 VPNs, Layer 3 VPNs, and VPLS. The configuration procedures specific to each type of VPN are described in the corresponding sections in the other configuration chapters.

To configure routing instances for VPNs, include the routing-instances statement at the [edit] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    instance-type type;
    interface interface-name;
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-name ];
    vrf-export [ policy-name ];
    vrf-target {
      export community-name;
      import community-name;
    }
  }
}
```

The following sections describe how to configure VPN routing instances:

Configure the Description on page 15

Configure the Instance Type on page 16

Configure Interfaces for VPN Routing on page 16

Configure the Route Distinguisher on page 18

Configure Policy for the PE Router's VRF Table on page 19

Enable Outbound Route Filtering for VPNs on page 24

Configure the Description

To provide a text description for the routing instance, include the description statement at the [edit routing-instances *routing-instance-name*] hierarchy level. Enclose any descriptive text that includes spaces in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.

```
[edit routing-instances routing-instance-name]
description text;
```

Configure the Instance Type

The instance type you configure varies depending on whether you are configuring Layer 2 VPNs, Layer 3 VPNs, VPLS, or virtual routers. Specify the instance type by configuring the instance-type statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

To enable Layer 2 VPN routing on a PE router, configure the instance-type statement as l2vpn at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
instance-type l2vpn;
```

To enable VPLS routing on a PE router, configure the instance-type statement as vpls at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
instance-type vpls;
```

Layer 3 VPNs require that each PE router have a VPN routing and forwarding (VRF) table for distributing routes within the VPN. To create the VRF table on the PE router, configure the instance-type statement as vrf at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
instance-type vrf;
```

To enable the virtual router routing instance, configure the instance-type statement as virtual-router at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
instance-type virtual-router;
```

Configure Interfaces for VPN Routing

On each PE router, you must configure an interface over which the VPN traffic travels between the PE and CE routers. The configuration described in this section is applicable to all types of VPNs. However, Layer 3 VPNs and carrier-of-carriers VPNs require some additional configuration described in the following sections:

Configure Interfaces for Layer 3 VPNs on page 17

Configure Interfaces for Carrier-of-Carriers VPNs on page 17

To configure interfaces for VPN routing, include the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
interface interface-name;
```

Specify both the physical and logical portions of the interface name, in the following format:

```
physical.logical
```

For example, in at-1/2/1.2, at-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

A logical interface can be associated with only one routing instance. If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level, and if you configure a specific interface for VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the VPN.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels, when you try to commit the configuration, it will fail.

Configure Interfaces for Layer 3 VPNs

When you configure the Layer 3 VPN interfaces at the [edit interfaces] hierarchy level, you must also configure family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
  }
}
```



Note

If you downgrade JUNOS to version 5.5 or earlier, the Layer 3 VPN configuration might become invalid. Layer 3 VPN interfaces between PE and CE routers formerly required the family mpls statement.

Configure Interfaces for Carrier-of-Carriers VPNs

When you configure carrier-of-carriers VPNs, you need to configure the family mpls statement in addition to the family inet statement for the interfaces between the PE and CE routers. For carrier-of-carriers VPNs, configure the logical interface as follows:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

If you configure family mpls on the logical interface and then configure this interface for a non-carrier-of-carriers vrf routing instance, the family mpls statement is automatically removed from the configuration for the logical interface, since it is not needed.

Configure the Route Distinguisher

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. VPN routing instances need a route distinguisher to help Border Gateway Protocol (BGP) distinguish overlapping network layer reachability information (NLRIs) from different VPNs.

We recommend that you use unique route distinguishers for each routing instance that you configure. Although you can use the same route distinguisher on all PE routers in the same VPN, if you use a unique route distinguisher, you can determine the PE router from which a route originated.

To configure a route distinguisher on a PE router, include the `route-distinguisher` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]
route-distinguisher (as-number:number | ip-address:number);
```

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

as-number:number, where *as-number* is an autonomous system (AS) number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.

ip-address:number, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the `router-id` statement, which is a nonprivate address in your assigned prefix range.

If you configure the `route-distinguisher-id` statement at the `[edit routing-options]` hierarchy level, a route distinguisher is automatically assigned to the routing instance. If you configure the `route-distinguisher` statement in addition to the `route-distinguisher-id` statement, the value configured for `route-distinguisher` supersedes the value generated from `route-distinguisher-id`.

To assign a route distinguisher automatically, include the `route-distinguisher-id` statement at the `[edit routing-options]` hierarchy level:

```
[edit]
routing-options {
  route-distinguisher-id ip-address;
}
```

A type 1 route distinguisher is automatically assigned to the routing instance using the format *ip-address:number*. The IP address is specified by the `route-distinguisher-id` statement and the number is unique for the routing instance.

Configure Policy for the PE Router's VRF Table

On each PE router, you must define policies that define how routes are imported into and exported from the router's VRF table. In these policies, you must define the route target and you can optionally define the route origin.

The following sections describe how to configure policy for the VRF tables:

Configure the Route Target on page 19

Configure the Route Origin on page 20

Configure Import Policy for the PE Router's VRF Table on page 21

Configure Export Policy for the PE Router's VRF Table on page 22

Apply Both the VRF Export and the BGP Export Policies on page 23

Configure a VRF Target on page 23

Configure the Route Target

As part of the policy configuration for the VPN routing table, you must define a route target which defines which VPN the route is a part of. When you configure different types of VPN services (Layer 2 VPNs, Layer 3 VPNs, or VPLS) on the same PE router, be sure to assign unique route target values to avoid the possibility of adding route and signaling information to the wrong VPN routing table.

Include the target option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members target: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following formats:

as-number:number, where *as-number* is an autonomous system (AS) number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.

ip-address:number, where *ip-address* is an Internet Protocol Version 4 (IPv4) address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65,535.

Configure the Route Origin

In the import and export policies for the PE router's VRF table, you can optionally define the route origin (also known as the site of origin), which identifies the set of routes learned from a particular CE site. This attribute ensures that a route learned from a particular site through a particular PE-CE connection is not distributed back to the site through a different PE-CE connection. It is particularly useful if you are using the Border Gateway Protocol (BGP) as the routing protocol between the PE and CE routers and if different sites in the VPN have been assigned the same AS numbers.

To configure a route origin, complete the following steps:

1. Include the origin option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members origin: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following format:

as-number:number, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65,535. We recommend that you use an IANA-assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.

ip-address:number, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65,535.

2. Include the community in the import policy for the PE router's VRF table by configuring the community statement with the *community-id* identifier defined in Step 1 at the [edit policy-options policy-statement *import-policy-name* term *import-term-name* from] hierarchy level. See "Configure Import Policy for the PE Router's VRF Table" on page 21.
3. Include the community in the export policy for the PE router's VRF table by configuring the community statement with the *community-id* defined in Step 1 at the [edit policy-options policy-statement *export-policy-name* term *export-term-name* then] hierarchy level. See "Configure Export Policy for the PE Router's VRF Table" on page 22.

Configure Import Policy for the PE Router's VRF Table

Each VPN can have a policy that defines how routes are imported into the PE router's VRF table. An import policy is applied to routes received from other PE routers in the VPN. A policy must evaluate all routes received over the IBGP session with the peer PE router. If the routes match the conditions, the route is installed in the PE router's *routing-instance-name.inet.0* VRF table. An import policy must contain a second term that rejects all other routes.

Unless an import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you try to commit the configuration, the commit fails. You can configure multiple import policies.

An import policy determines what to import to a specified VRF table based on the VPN routes learned from the remote PE routers through IBGP. The IBGP session is configured at the [edit protocols bgp] hierarchy level. If you also configure an import policy at the [edit protocols bgp] hierarchy level, the import policies at the [edit policy-options] hierarchy level and the [edit protocols bgp] hierarchy level are combined through a logical AND operation. This allows you to filter traffic as a group.

To configure an import policy for the PE router's VRF table, follow these steps:

1. To define an import policy, include the policy-statement statement at the [edit policy-options] hierarchy level. For all PE routers, an import policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement import-policy-name {
    term import-term-name {
      from {
        protocol bgp;
        community community-id;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

The *import-policy-name* policy evaluates all routes received over the IBGP session with the other PE router. If the routes match the conditions in the from statement, the route is installed in the PE router's *routing-instance-name.inet.0* VRF table. The second term in the policy rejects all other routes.

For more information about creating policies, see the *JUNOS Internet Softw are Configuration Guide: Policy Framework*.

2. To configure an import policy, include the vrf-import statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-import [ import-policy-name ];
```

Configure Export Policy for the PE Router's VRF Table

Each VPN can have a policy that defines how routes are exported from the PE router's VRF table. An export policy is applied to routes sent to other PE routers in the VPN. An export policy must evaluate all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or Routing Information Protocol (RIP) routing protocols, or static routes.) If the routes match the conditions, the specified community target (which is the route target) is added to them and they are exported to the remote PE routers. An export policy must contain a second term that rejects all other routes.

Export policies defined within the VPN routing instance are the only export policies that apply to the VRF table. Any export policy that you define on the IBGP session between the PE routers has no effect on the VRF table. You can configure multiple export policies.

To configure an export policy for the PE router's VRF table, follow these steps:

1. To define an export policy, include the policy-statement statement at the [edit policy-options] hierarchy level. For all PE routers, an export policy must distribute VPN routes to and from the connected CE routers in accordance with the type of routing protocol that you configure between the CE and PE routers within the routing instance. An export policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement export-policy-name {
    term export-term-name {
      from protocol (bgp | ospf | rip | static);
      then {
        community add community-id;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
}
```

The *export-policy-name* policy evaluates all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or RIP routing protocol, or static routes.) If the routes match the conditions in the from statement, the community target specified in the then community add statement is added to them and they are exported to the remote PE routers. The second term in the policy rejects all other routes.

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

2. To apply the policy, include the vrf-export statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-export export-policy-name;
```

Apply Both the VRF Export and the BGP Export Policies

When you apply a VRF export policy as described in “Configure Export Policy for the PE Router’s VRF Table” on page 22, routes from VPN routing instances are advertised to other PE routers based on this policy, while the BGP export policy is ignored.

If you configure the `vpn-apply-export` statement, both the VRF export and BGP group or neighbor export policies are applied (VRF first, then BGP) before routes are advertised in the VPN routing tables to other PE routers.

Include the `vpn-apply-export` statement at the `[edit protocols bgp]`, `[edit protocols bgp group group-name]`, or `[edit protocols bgp group group-name neighbor neighbor]` hierarchy level:

```
[edit]
vpn-apply-export;
```

Configure a VRF Target

Before JUNOS 5.5, you needed to configure VRF import and export policies for each VPN routing instance on a PE router. These policies control redistribution of routes between the VRF table and BGP.

In the current JUNOS release, the `vrf-target` statement simplifies this configuration. Configuring a VRF target community using the `vrf-target` statement causes default VRF import and export policies to be generated that accept and tag routes with the specified target community. You can still create more complex policies by explicitly configuring VRF import and export policies. These policies override the default policies generated when you configure the `vrf-target` statement.

If you do not configure the import and export options of the `vrf-target` statement, the specified community string is applied in both directions. The import and export keywords give you more flexibility, allowing you to specify a different community for each direction.

An example of how you might configure the `vrf-target` statement follows:

```
[edit routing-instances sample]
vrf-target target:69:102;
```

The syntax for the VRF target community is not a name. You must specify it in the format `target:x.y`. A community name cannot be specified because this would also require you to configure the community members for that community using the `policy-options` statement. If you define the `policy-options` statements, then you can just configure VRF import and export policies as usual. The purpose of the `vrf-target` statement is to simplify the configuration by allowing you to configure most statements at the `[edit routing-instances]` hierarchy level.

To configure a VRF target, include the `vrf-target` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    vrf-target community;
  }
}
```

To configure the `vrf-target` statement with the `export` and `import` options, include the following statements at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    vrf-target {
      export community-name;
      import community-name;
    }
  }
}
```

Enable Outbound Route Filtering for VPNs

Outbound route filtering allows you to filter BGP route advertisements for a particular BGP peer or set of peers. Peers in the same BGP group can have different routing table entries by filtering so that only a select peer or set of peers receives the route advertisements. For VPNs, this allows you to configure the PE routers to accept only a subset of the total number of VPN routes based on the configured VRF route targets.

You can enable outbound route filtering for a VPN either with or without a route reflector:

route reflector—The router acting as a route reflector receives all the VPN routes from its clients, but only sends to each VPN client the PE routes that have route targets that the PE router registered for using the extended community-based outbound route filtering.

no route reflector—Each PE router accepts all extended community outbound route filters from its peer PE routers. It requests outbound route filtering from its peer PE routers based on the route targets it is interested in.

You can enable outbound route filtering for routing instances by including the `outbound-route-filtering` statement at the [edit routing-instances *routing-instance-name* protocols bgp group *group-name*] hierarchy level or at the [edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*] hierarchy level:

```
[edit]
outbound-route-filtering {
  extended-community {
    accept;
    no-accept;
    vrf-filter;
  }
}
```

To accept a peer's request for filtering on Network Layer Reachability Information (NLRI) route advertisements, include the `accept` option. To deny a peer's request for filtering on Network Layer Reachability Information (NLRI) route advertisements, include the `no-accept` option. To request filtering from a remote peer, include the `vrf-filter` option.

Configure a Virtual-Router Routing Instance

A virtual-router routing instance like VRF routing instances maintains separate routing and forwarding tables for each instance. However, many of the configuration steps required for VRF routing instances are not required for virtual-router routing instances. Specifically, you do not need to configure a route distinguisher, a routing table policy (the vrf-export, vrf-import, and route-distinguisher statements), or MPLS between the provider routers.

Configure a virtual-router routing instance as follows:

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    instance-type virtual-router;
    interface interface-name;
    protocols { ... }
  }
}
```

The sections that follow outline the configuration procedures for a virtual-router routing instance:

Configure a Routing Protocol Between the Service Provider Routers on page 25

Configure Logical Interfaces Between Participating Routers on page 26

Configure a Routing Protocol Between the Service Provider Routers

The service provider routers need to be able to exchange routing information. You can configure the following protocols for the virtual-router routing instance protocols statement configuration at the [routing-instances routing-instance-name] hierarchy level:

BGP

ISIS

LDP

OSPF

RIP

You can also configure static routes.

The following are not supported for virtual-router routing instances:

IBGP route reflection

Multicast routing

If you configure LDP under a virtual-router instance, by default LDP routes are placed in both the routing instance's inet.0 and inet.3 routing tables (for example, sample.inet.0 and sample.inet.3). To restrict LDP routes to only the routing instance's inet.3 table, you must include the no-forwarding statement at the [routing-instances *routing-instance-name* protocols ldp] hierarchy level:

```
[edit routing-instances routing-instance-name protocols ldp]
no-forwarding;
```

This places the LDP routes only in to the inet.3 routing table, so the corresponding IGP route in the inet.0 routing table can be redistributed and advertised into other routing protocols.

For information on how to configure routing protocols, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure Logical Interfaces Between Participating Routers

You must configure an interface to each customer router participating in the routing instance and to each provider router participating in the routing instance. Each virtual-router routing instance requires its own separate logical interfaces to all routers participating in the instance. To configure interfaces for virtual-router instances, include the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
interface interface-name;
```

Specify both the physical and logical portions of the interface name, in the following format:

```
physical.logical
```

For example, in at-1/2/1.2, at-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

You must also configure the interfaces at the [edit interfaces] hierarchy level.

One method of providing this logical interface between the provider routers is by configuring tunnels between them. You can configure IPSec, GRE, or IP-IP tunnels between the provider routers, terminating the tunnels at the virtual-router instance.

For information on how to configure tunnels and interfaces, see the *JUNOS Internet Software Configuration Guide: Services Interfaces*.

Configure Graceful Restart

Graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any VPN services provided by the router. Graceful restart is supported on Layer 2 VPNs, Layer 3 VPNs, and virtual-router routing instances. It is not available for VPLS.

To enable VPN graceful restart, include the graceful-restart statement at the [edit routing-options] hierarchy level on the PE router:

```
[edit routing-options]
graceful-restart {
  disable;
  path-selection-defer-time-limit time-limit;
}
```

Also include the graceful-restart statement at the [edit routing-instances *routing-instance-name* routing-options] hierarchy level on the PE router:

```
[edit routing-instances routing-instance-name routing-options]
graceful-restart {
  disable;
  path-selection-defer-time-limit time-limit;
}
```

Rewrite Markers and VPNs

A marker reads the current forwarding class and loss priority information associated with a packet and finds the chosen code point from a table. It then writes the code point information into the packet header. Entries in a marker configuration represent the mapping of the current forwarding class into a new forwarding class, to be written into the header.

You define markers in the rewrite rules section of the CoS configuration hierarchy and reference them in the logical interface configuration. You can configure different rewrite rules to handle VPN traffic and non-VPN traffic. The rewrite rule can be applied to MPLS and IPv4 packet headers simultaneously, making it possible to initialize MPLS EXP and IP precedence bits at LSP ingress.

For a detailed example of how to configure rewrite rules for MPLS and IPv4 packets and for more information on how to configure statements at the [edit class-of-service] hierarchy level, see the *JUNOS Internet Software Configuration Guide: Network Interfaces and Class of Service*.

