

Chapter 2

Configuring VPNs

Layer 2 virtual private networks (VPNs), Layer 3 VPNs, virtual-router routing instances, and virtual private LAN service (VPLS) use a common infrastructure within JUNOS and common configuration procedures. This chapter describes the common configuration steps. Complete these configuration steps, regardless of which type of VPN you are configuring, before proceeding to the more specific configuration steps described in other chapters.

This chapter describes how to configure Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS, discussing the following topics:

- Enabling a Signaling Protocol on the PE Routers on page 12
- Configuring an IGP on the PE and P Routers on page 15
- Configuring an IBGP Session Between PE Routers on page 16
- Configuring a VPN Routing Instance on the PE Routers on page 17
- Configuring a Virtual-Router Routing Instance on page 31
- Configuring Graceful Restart on page 33
- Configuring Aggregate Labels for VPNs on page 34
- Rewriting Markers and VPNs on page 34
- Transmitting Nonstandard BPDUs on page 35
- Pinging VPNs and Layer 2 Circuits on page 35
- Configuring a Path MTU Check for VPNs on page 37

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

- Configuring Layer 2 VPNs on page 71
- Configuring Layer 3 VPNs on page 145
- Configuring VPLS on page 383

Enabling a Signaling Protocol on the PE Routers

For VPNs to function, you must enable a signaling protocol on the provider edge (PE) routers. To enable a signaling protocol, perform the steps in one of the following sections:

- Using LDP for VPN Signaling on page 12
- Using RSVP for VPN Signaling on page 14



NOTE: As with any configuration involving Multiprotocol Label Switching (MPLS), you cannot configure any of the core-facing interfaces on the PE routers over dense Fast Ethernet Physical Interface Cards (PICs).

Using LDP for VPN Signaling

To use Label Distribution Protocol (LDP) for VPN signaling, perform the following steps on the PE and provider (P) 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 P 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 customer edge (CE) routers.

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

2. Configure the 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 P 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 Routing Protocols Configuration Guide*.

Using 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 P router, enable RSVP on the 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 P 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 P 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 P 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 MPLS Applications Configuration Guide*.

Configuring an IGP on the PE and P Routers

For Layer 2 VPNs, Layer 3 VPNs, virtual-router routing instances, and VPLS to function properly, the service provider's PE and P routers must be able to exchange routing information. To allow them to do this, you must configure either an 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 Routing Protocols Configuration Guide*.

Configuring an IBGP Session Between PE Routers

You must configure an internal Border Gateway Protocol (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 (inet-vpn | inet6-vpn) {
        unicast;
      }
      family l2vpn {
        signaling;
      }
      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 an IBGP session for Layer 2 VPNs and VPLS, include the `signaling` statement at the `[edit protocols bgp group group-name family l2vpn]` hierarchy level:

```
[edit protocols bgp group group-name family l2vpn]
  signaling;
```

To configure an IPv4 IBGP session for Layer 3 VPNs, configure the `unicast` statement at the `[edit protocols bgp group group-name family inet-vpn]` hierarchy level:

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

To configure an IPv6 IBGP session for Layer 3 VPNs, configure the `unicast` statement at the `[edit protocols bgp group group-name family inet6-vpn]` hierarchy level:

```
[edit protocols bgp group group-name family inet6-vpn]
  unicast;
```

Configuring 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 following statements:

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

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

To configure VPN routing instances, you perform the steps in the following sections:

- Configuring the Description on page 18
- Configuring the Instance Type on page 18
- Configuring Interfaces for VPN Routing on page 19
- Configuring the Route Distinguisher on page 21
- Configuring Policy for the PE Router's VRF Table on page 22
- Configuring BGP Route Target Filtering on page 29

Configuring the Description

To provide a text description for the routing instance, include the **description** statement. If the text includes one or more spaces, enclose 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.

To configure a text description, include the **description** statement:

```
description text;
```

You can include the **description** statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring the Instance Type

The instance type you configure varies depending on whether you are configuring Layer2 VPNs, Layer 3 VPNs, VPLS, or virtual routers. Specify the instance type by configuring the **instance-type** statement:

- To enable Layer 2 VPN routing on a PE router, include the **instance-type** statement and specify the value **l2vpn**:

```
instance-type l2vpn;
```

- To enable VPLS routing on a PE router, include the **instance-type** statement and specify the value **vpls**:

```
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, include the **instance-type** statement and specify the value **vrf**:

```
instance-type vrf;
```

- To enable the virtual-router routing instance, include the **instance-type** statement and specify the value **virtual-router**:

```
instance-type virtual-router;
```

You can include the **instance-type** statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring 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 sections that follow describe how to configure interfaces for VPNs:

- General Configuration for VPN Routing on page 19
- Configuring Interfaces for Layer 3 VPNs on page 20
- Configuring Interfaces for Carrier-of-Carriers VPNs on page 20
- Configuring Unicast RPF on VPN Interfaces on page 20

The configuration described in “General Configuration for VPN Routing” on page 19 applies to all types of VPNs. For Layer 3 VPNs and carrier-of-carriers VPNs, complete the configuration described in that section before proceeding to the interface configuration sections specific to those topics.

General Configuration for VPN Routing

The configuration described in this section applies to all types of VPNs.

To configure interfaces for VPN routing, include the `interface` statement:

```
interface interface-name;
```

You can include the `interface` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-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 or at the [edit logical-routers *logical-router-name* 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 the [edit protocols] hierarchy level and at either the [edit routing-instances *routing-instance-name*] or [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*] hierarchy levels, an attempt to commit the configuration fails.

Configuring 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;
  }
}
```

Configuring 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 routing instance, the **family mpls** statement is automatically removed from the configuration for the logical interface, since it is not needed.

Configuring Unicast RPF on VPN Interfaces

For VPN interfaces that carry IP version 4 or version 6 (IPv4 or IPv6) traffic, you can reduce the impact of denial-of-service (DoS) attacks by configuring unicast reverse-path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled.

You can configure unicast RPF on a VPN interface by enabling unicast RPF on the interface and including the **interface** statement at the [edit routing-instances routing-instance-name] hierarchy level.

You cannot configure unicast RPF on the core-facing interfaces. You can only configure unicast RPF on the CE router-to-PE router interfaces on the PE router. However, for virtual-router routing instances, unicast RPF is supported on all interfaces you specify in the routing instance.

For information on how to configure unicast RPF on VPN interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Configuring 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 the Border Gateway Protocol (BGP) to distinguish between potentially identical network layer reachability information (NLRI) messages received from different VPNs.

We recommend that you use a unique route distinguisher 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:

```
route-distinguisher (as-number:number | ip-address:number);
```

You can include the `route-distinguisher` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

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:

```
route-distinguisher-id ip-address;
```

You can include the `route-distinguisher-id` statement at the following hierarchy levels:

- [edit routing-options]
- [edit logical-routers *logical-router-name* routing-options]

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.

Configuring 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.

To configure policy for the VRF tables, you perform the steps in the following sections:

- Configuring the Route Target on page 23
- Configuring the Route Origin on page 23
- Configuring Import Policy for the PE Router's VRF Table on page 25
- Configuring Export Policy for the PE Router's VRF Table on page 26
- Applying Both the VRF Export and the BGP Export Policies on page 27
- Configuring a VRF Target on page 28

Configuring 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.

To configure the route target, include the `target` option in the community statement:

```
community name members target:community-id;
```

You can include the community statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-routers *logical-router-name* policy-options]

name is the name of the community.

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

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community value. 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 value can be a number in the range 0 through 4,294,967,295 ($2^{32} - 1$).
- *ip-address:number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community 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. The community value can be a number in the range 1 through 65,535.

Configuring the Route Origin

In the import and export policies for the PE router's VRF table, you can optionally assign the route origin (also known as the site of origin) for a PE router's VRF routes using a VRF export policy applied to multiprotocol external Border Gateway Protocol (MP-EBGP) VPN IPv4 route updates sent to other PE routers.

Matching on the assigned route origin attribute in a receiving PE's VRF import policy helps ensure that VPN-IPv4 routes learned through MP-EBGP updates from one PE are not reimported to the same VPN site from a different PE connected to the same site.

To configure a route origin, complete the following steps:

1. Include the `origin` option in the `community` statement:

```
community name members origin:community-id;
```

You can include the `community` statement at the following hierarchy levels:

- [edit policy-options]
- [edit logical-routers *logical-router-name* policy-options]

name is the name of the community.

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

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community value. 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 value 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 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. The community value 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 "Configuring Import Policy for the PE Router's VRF Table" on page 25.
 3. Include the community in the export 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 *export-policy-name* term *export-term-name* then] hierarchy level. See "Configuring Export Policy for the PE Router's VRF Table" on page 26.

See "Route Origin for VPNs" on page 52 for a configuration example.

Configuring 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. For all PE routers, an import policy must always include the `policy-statement` statement, at a minimum:

```

policy-statement import-policy-name {
  term import-term-name {
    from {
      protocol bgp;
      community community-id;
    }
    then accept;
  }
  term term-name {
    then reject;
  }
}

```

You can include the `policy-statement` statement at the following hierarchy levels:

- `[edit policy-options]`
- `[edit logical-routers logical-router-name policy-options]`

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 Policy Framework Configuration Guide*.

- To configure an import policy, include the `vrf-import` statement:

```
vrf-import import-policy-name;
```

You can include the `vrf-import` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring 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:

- 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.

To define an export policy, include the `policy-statement` statement. An export policy must always include the `policy-statement` statement, at a minimum:

```
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;
  }
}
```



NOTE: Configuring the `community add` statement is a requirement for Layer 2 VPN VRF export policies.

You can include the `policy-statement` statement at the following hierarchy levels:

- [edit `policy-options`]
- [edit `logical-routers logical-router-name policy-options`]

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 protocols, 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 Policy Framework Configuration Guide*.

2. To apply the policy, include the `vrf-export` statement:

```
vrf-export export-policy-name;
```

You can include the `vrf-export` statement at the following hierarchy levels:

- [edit `routing-instances routing-instance-name`]
- [edit `logical-routers logical-router-name routing-instances routing-instance-name`]

Applying Both the VRF Export and the BGP Export Policies

When you apply a VRF export policy as described in “Configuring Export Policy for the PE Router’s VRF Table” on page 26, 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.

If you configure a PE router as a route reflector or as an AS border router, the behavior enabled by the `vpn-apply-export` statement is enabled on these routers automatically. For information on how to configure a route reflector or an AS border router, see the *JUNOS Routing Protocols Configuration Guide*.

When you configure the `vpn-apply-export` statement, be aware of the following:

- Routes imported into the `I3vpn.bgp.0` routing table retain the attributes of the original routes (for example, an OSPF route remains an OSPF route even when it is stored in the `I3vpn.bgp.0` routing table). You should be aware of this when you configure an export policy for connections between an IBGP PE router and a PE router, a route reflector and a PE router, or AS boundary router (ASBR) peer routers.
- By default, all routes in the `I3vpn.bgp.0` routing table are exported to the IBGP peers. If the last statement of the export policy is `deny all` and if the export policy does not specifically match on routes in the `I3vpn.bgp.0` routing table, no routes are exported.

To apply both the VRF export and BGP export policies to VPN routes, include the `vpn-apply-export` statement:

```
vpn-apply-export;
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

Configuring a VRF Target

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.

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:

```
vrf-target community;
```

You can include the `vrf-target` statement at the following hierarchy levels:

- `[edit routing-instances routing-instance-name]`
- `[edit logical-routers logical-router-name routing-instances routing-instance-name]`

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

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

To configure the `vrf-target` statement with the `export` and `import` options, include the following statements:

```
vrf-target {
    export community-name;
    import community-name;
}
```

You can include the `vrf-target` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Configuring BGP Route Target Filtering

BGP route target filtering allows you to distribute VPN routes to only the routers that need them. In VPN networks without BGP route target filtering configured, BGP distributes all VPN routes to all VPN peer routers.

The following sections provide an overview of BGP route target filtering and how to configure it for VPNs:

- BGP Route Target Filtering Overview on page 29
- Configuring BGP Route Target Filtering for VPNs on page 30

For more information on BGP route target filtering, see the Internet draft `draft-marques-ppvnp-rt-constrain-01.txt`, *Constrained VPN Route Distribution*.

BGP Route Target Filtering Overview

PE routers, unless they are configured as route reflectors or are running an EBGp session, discard any VPN routes that do not include a route target extended community as specified in the local VRF import policies. This is the default behavior of the JUNOS software.

However, unless it is explicitly configured not to store VPN routes, any router configured either as a route reflector or border router for a VPN address family must store all of the VPN routes that exist in the service provider's network. Also, though PE routers can automatically discard routes that do not include a route target extended community, route updates continue to be generated and received.

By reducing the number of routers receiving VPN routes and route updates, BGP route target filtering helps to limit the amount of overhead associated with running a VPN. BGP route target filtering is most effective at reducing VPN-related administrative traffic in networks where there are many route reflectors or AS border routers that do not participate in the VPNs directly (not acting as PE routers for the CE devices).

BGP route target filtering uses standard UPDATE messages to distribute route target extended communities between routers. This allows BGP to use its standard loop detection mechanisms, path selection, policy support, and database exchange implementation.

Configuring BGP Route Target Filtering for VPNs

BGP route target filtering is enabled through the exchange of the `route-target` address family, stored in the `bgp.rtarget.0` routing table. Based on the `route-target` address family, the route target NLRI (address family indicator [AFI] = 1, subsequent AFI [SAFI] = 132) is negotiated with its peers.

On a system that has locally configured VRF instances, BGP automatically generates local routes corresponding to targets referenced in the `vrf-import` policies.

To configure BGP route target filtering, include the `family route-target` statement:

```
family route-target {
  advertise-default;
  external-paths number;
  prefix-limit number;
}
```

For a list of hierarchy levels at which you can configure the `family route-target` statement, see the statement summary section for this statement.

The `advertise-default` statement and the `external-paths` statement affect the BGP route target filtering configuration as follows:

- The `advertise-default` statement causes the router to advertise the default route target route (0:0:0/0) and suppress all routes that are more specific. This can be used by a route reflector on BGP groups consisting of neighbors that act as PE routers only. PE routers often need to advertise all routes to the route reflector.

Suppressing all route target advertisements other than the default route reduces the amount of information exchanged between the route reflector and the PE routers. The JUNOS software further helps to reduce route target advertisement overhead by not maintaining dependency information unless a nondefault route is received.

- The `external-paths` statement (which has a default value of 1) causes the router to advertise the VPN routes that reference a given route target. The number you specify determines the number of external peer routers (currently advertising that route target) that receive the VPN routes.
- The `prefix-limit` statement limits the number of prefixes that can be received from a peer router.

The `route-target`, `advertise-default` and `external-path` statements affect the RIB-OUT state and must be consistent between peer routers that share the same BGP group. The `prefix-limit` statement affects the receive side only and can have different settings between different peer routers in a BGP group.

For examples illustrating how to configure BGP route target filtering for VPNs, see “VPN Examples” on page 39.

Configuring a Virtual-Router Routing Instance

A virtual-router routing instance, like a VRF routing instance, 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 service provider routers.

Configure a virtual-router routing instance by including the following statements:

```
description text;  
instance-type virtual-router;  
interface interface-name;  
protocols { ... }
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

The following sections explain how to configure a virtual-router routing instance:

- Configuring a Routing Protocol Between the Service Provider Routers on page 31
- Configuring Logical Interfaces Between Participating Routers on page 32

Configuring 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
- IS-IS
- LDP
- OSPF
- Protocol Independent Multicast (PIM)
- RIP

You can also configure static routes.

IBGP route reflection is not supported for virtual-router routing instances.

If you configure LDP under a virtual-router instance, LDP routes are placed by default in 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, include the `no-forwarding` statement:

```
no-forwarding;
```

You can include the `no-forwarding` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols ldp]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols ldp]

When you restrict the LDP routes to only the `inet.3` routing table, 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 Routing Protocols Configuration Guide*.

Configuring Logical Interfaces Between Participating Routers

You must configure an interface to each customer router participating in the routing instance and to each P router participating in the routing instance. Each virtual-router routing instance requires its own separate logical interfaces to all P routers participating in the instance. To configure interfaces for virtual-router instances, include the `interface` statement:

```
interface interface-name;
```

You can include the `interface` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-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 IP Security (IPSec), generic routing encapsulation (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 Services Interfaces Configuration Guide*.

Configuring 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, virtual-router routing instances, and VPLS.

To enable VPN graceful restart, include the `graceful-restart` statement:

```
graceful-restart {
  disable;
  restart-duration time-limit;
}
```

You can configure the `restart-duration` option at either the global or routing instance level. The routing instance value overrides the global value if both are configured.

To configure the `graceful-restart` statement globally, include it at the following hierarchy levels:

- [edit routing-options]
- [edit logical-routers *logical-router-name* routing-options]

To configure the `graceful-restart` statement in the routing instance configuration, include it at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* routing-options]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options]

The `restart-duration` option sets the period of time the router waits for a graceful restart to complete. You can configure a time between 1 through 600 seconds. The default value is 300 seconds. At the end of the configured time period, the router performs a standard restart without recovering state from the neighboring routers. This disrupts VPN services, but is probably necessary if the router is not functioning normally.

Configuring Aggregate Labels for VPNs

Aggregate labels for VPNs allow a Juniper Networks routing platform to aggregate a set of incoming labels (labels received from a peer router) into a single forwarding label that is selected from the set of incoming labels. The single forwarding label corresponds to a single next hop for that set of labels. Label aggregation reduces the number of VPN labels that the router must examine.

For a set of labels to share an aggregate forwarding label, they must belong to the same forwarding equivalence class (FEC). The labeled packets must have the same destination egress interface.

Including the `community community-name` statement with the `aggregate-label` statement lets you specify prefixes with a common origin community. Set by policy on the peer PE, these prefixes represent an FEC on the peer PE router.



CAUTION: If the target community is set by mistake instead of the origin community, forwarding problems at the egress PE can result. All prefixes from the peer PE will appear to be in the same FEC, resulting in a single inner label for all CE routers behind a given PE in the same VPN.

To configure aggregate labels for VPNs, include the `aggregate-label` statement:

```
aggregate-label {
    community community-name;
}
```

For a list of hierarchy levels at which you can include this statement, see the statement summary for this statement.

For information on how to configure a community, see the *JUNOS Policy Framework Configuration Guide*.

Rewriting 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 class-of-service (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 experimental (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 Class of Service Configuration Guide*.

Transmitting Nonstandard BPDUs

Circuit cross-connect (CCC) protocol, Layer 2 circuit, and Layer 2 VPN configurations can transmit nonstandard bridge protocol data units (BPDUs) generated by other vendors' equipment. This is the default behavior on all supported PICs and requires no additional configuration.

The following PICs are supported on T-series and M320 routers and can transmit nonstandard BPDUs:

- 1-port Gigabit Ethernet PIC
- 2-port Gigabit Ethernet PIC
- 4-port Gigabit Ethernet PIC
- 10-port Gigabit Ethernet PIC

Pinging VPNs and Layer 2 Circuits

For testing purposes, you can ping Layer 2 VPNs, Layer 3 VPNs, and Layer 2 circuits by using the `ping mpls` command. The `ping mpls` command helps to verify that a VPN or circuit has been enabled. This command tests the integrity of the VPN or Layer 2 circuit connection between the PE routers. It does not test the connection between a PE router and a CE router.

You issue the `ping mpls` command from the ingress PE router of the VPN or Layer 2 circuit to the egress PE router of the same VPN or Layer 2 circuit. When you execute the `ping` command, echo requests are sent as MPLS packets.

The payload is a User Datagram Protocol (UDP) packet forwarded to the address `127.0.0.1`. The contents of this packet are defined in the Internet draft `draft-ietf-mpls-lsp-ping-05.txt`, *Detecting MPLS Data Plane Failures*. The label and interface information for building and sending this information as an MPLS packet is the same as for standard VPN traffic, but the time-to-live (TTL) of the innermost label is set to 1.

When the echo request arrives at the egress PE router, the contents of the packet are checked, and then a reply that contains the correct return is sent by means of UDP. The PE router sending the echo request waits to receive an echo reply after a timeout of 2 seconds (you cannot configure this value).

You must configure MPLS at the `[edit protocols mpls]` hierarchy level on the egress PE router (the router receiving the MPLS echo packets) to be able to ping the VPN or Layer 2 circuit. You must also configure the address `127.0.0.1/32` on the egress PE router's `lo0` interface. If this is not configured, the egress PE router does not have this forwarding entry and therefore simply drops the incoming MPLS pings.

The `ping mpls` command has the following limitations:

- You cannot ping an IPv6 destination prefix.
- You cannot ping a VPN or Layer 2 circuit from a router that is attempting a graceful restart.
- You cannot ping a VPN or Layer 2 circuit from a logical router.

You can also determine whether an LSP linking two PE routers in a VPN is up by pinging the end point address of the LSP. The command you use to ping an MPLS LSP end point is `ping mpls lsp-end-point address`. This command tells you what type of LSP (RSVP or LDP) terminates at the address specified and whether that LSP is up or down.

For a detailed description of this command, see the *JUNOS Routing Protocols and Policies Command Reference*.

Pinging a Layer 2 VPN

To ping a Layer 2 VPN, use one of the following commands:

- `ping mpls l2vpn interface interface-name`

You ping an interface configured for the Layer 2 VPN on the egress PE router.

- `ping mpls l2vpn instance l2vpn-instance-name local-site-id local-site-id-number remote-site-id remote-site-id-number`

You ping a combination of the Layer 2 VPN routing instance name, the local site identifier, and the remote site identifier to test the integrity of the Layer 2 VPN connection (specified by the identifiers) between the ingress and egress PE routers.

Pinging a Layer 3 VPN

To ping a Layer 3 VPN, use the following command:

- `ping mpls l3vpn l3vpn-name prefix prefix <count count>`

You ping a combination of a IPv4 destination prefix and a Layer 3 VPN name on the egress PE router to test the integrity of the VPN connection between the ingress and egress PE routers. The destination prefix corresponds to a prefix in the Layer 3 VPN. However, the ping tests only whether the prefix is present in a PE router's VRF table. It does not test the connection between a PE router and a CE router.

Pinging a Layer 2 Circuit

To ping a Layer 2 circuit, use one of the following commands:

- `ping mpls l2circuit interface interface-name`

You ping an interface configured for the Layer 2 circuit on the egress PE router.

- `ping mpls l2circuit virtual-circuit neighbor <prefix> <virtual-circuit-id>`

You ping a combination of the IPv4 prefix and the virtual circuit identifier on the egress PE router to test the integrity of the Layer 2 circuit between the ingress and egress PE routers.

Setting the Forwarding Class of the Ping Packets

When you execute the `ping mpls` command, the ping packets forwarded to the destination include MPLS labels. It is possible to set the value of the forwarding class for these ping packets by using the `exp` option with the `ping mpls` command. For example, to set the forwarding class to 5 when pinging a Layer 3 VPN, issue the following command:

```
ping mpls l3vpn westcoast source 1.1.1.1 prefix 2.2.2.2 exp 5 count 20 detail
```

This command would make the router attempt to ping the Layer 3 VPN `westcoast` using ping packets with an EXP forwarding class of 5. The default forwarding class used for the `ping mpls` command packets is 7.

Configuring a Path MTU Check for VPNs

By default, the maximum transmission unit (MTU) check for VPN routing instances is disabled on M-series routers (except the M320 router) and enabled for the M320, T-series, and J-series routers. On M-series routers, you can configure path MTU checks on the outgoing interfaces for unicast traffic routed on VRF routing instances and on virtual-router routing instances.

When you enable an MTU check, the routing platform sends an Internet Control Message Protocol (ICMP) message when a packet traversing the routing instance exceeds the MTU size and has the `do-not-fragment` bit set. The ICMP message uses the VRF local address as its source address.

For an MTU check to work in a routing instance, you must both include the `vrf-mtu-check` statement at the `[edit chassis]` hierarchy level and assign at least one interface containing an IP address to the routing instance.

To configure path MTU checks, do the tasks described in the following sections:

- Enabling Path MTU Checks for a VPN Routing Instance on page 38
- Assigning an IP Address to the VPN Routing Instance on page 38

For more information on the Path MTU check, see the *JUNOS System Basics Configuration Guide*.

Enabling Path MTU Checks for a VPN Routing Instance

To enable path checks on the outgoing interface for unicast traffic routed on a VRF or virtual-router routing instance, include the `vrf-mtu-check` statement at the `[edit chassis]` hierarchy level:

```
[edit chassis]
vrf-mtu-check;
```

Assigning an IP Address to the VPN Routing Instance

To ensure that the path MTU check functions properly, at least one IP address must be associated with each VRF or virtual-router routing instance. If an IP address is not associated with the routing instance, ICMP reply messages cannot be sent.

Typically, the VRF or virtual-router routing instance IP address is drawn from among the IP addresses associated with interfaces configured for that routing instance. If none of the interfaces associated with a VRF or virtual-router routing instance is configured with an IP address, you need to explicitly configure a logical loopback interface with an IP address. This interface must then be associated with the routing instance. See “Configuring a Logical Unit on the Loopback Interface” on page 171 for details.