

## Chapter 16

# Configuring VPLS

Virtual private LAN service (VPLS) allows you to provide a point-to-multipoint LAN between a set of sites in a virtual private network (VPN).

To configure VPLS functionality, you must enable VPLS support on the provider edge (PE) router. You must also configure PE routers to distribute routing information to the other PE routers in the VPLS and configure the circuits between the PE routers and the customer edge (CE) routers.

Each VPLS is configured under a routing instance of type `vpls`. A `vpls` routing instance can transparently carry Ethernet traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a VPLS routing instance are listed under that instance.

To configure VPLS, include the following statements:

```

description text;
forwarding-options {
  family vpls {
    filter input input-filter-name;
    flood input flood-filter-name;
  }
}
instance-type vpls;
interface interface-name;
route-distinguisher (as-number:id | ip-address:id);
vrf-export [ policy-names ];
vrf-import [ policy-names ];
vrf-target target:target-id;
protocols {
  vpls {
    active-interface {
      any;
      primary interface-name;
    }
    interface-mac-limit limit;
    mac-table-size size;
    no-tunnel-services;
    site site-name {
      site-identifier identifier;
    }
    site-range number;
    traceoptions {
      file filename <replace> <size size> <files number> <nostamp>;
      flag flag <flag-modifier> <disable>;
    }
    tunnel-services {
      devices device-names;
      primary primary-device-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*]

For VPLS, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Routing Protocols Configuration Guide*.

In addition to these statements, you must configure Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) between the PE routers, internal border gateway protocol (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider (P) routers.

By default, VPLS is disabled.

Many configuration procedures for VPLS are identical to the procedures for Layer 2 VPNs and Layer 3 VPNs. These procedures are described in detail in “Configuring VPNs” on page 11 and include the following:

- 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

This chapter describes how to configure VPLS, discussing the following topics:

- Configuring VPLS Without a Tunnel Services PIC on page 385
- Configuring Interfaces for VPLS Routing on page 386
- Configuring the VPLS Routing Instance on page 391
- Configuring an Ethernet Switch as the CE Device on page 395
- Mapping VPLS Traffic to a Specific LSP on page 396
- Configuring VPLS Filters and Policers on page 397
- Specifying the VT Interfaces Used by VPLS Routing Instances on page 403
- Configuring VPLS Multihoming on page 404
- Tracing VPLS Traffic and Operations on page 407

## Configuring VPLS Without a Tunnel Services PIC

---

VPLS normally uses a dynamic virtual tunnel logical interface on a Tunnel Services PIC to model traffic from a remote site (a site on a remote PE router that is in a VPLS domain). All traffic coming from a remote site is treated as coming in over the virtual port representing this remote site, for the purposes of Ethernet flooding, forwarding, and learning. An MPLS lookup based on the inner VPN label is done on a PE router. The label is stripped and the Layer 2 Ethernet frame contained within is forwarded to a Tunnel Services PIC. The PIC loops back the packet and then a lookup based on Ethernet MAC addresses is completed. This approach requires that the router have a Tunnel Services PIC and that the PE router completes two protocol lookups.

You can configure VPLS without a Tunnel Services PIC. To do so, you use a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

By default, VPLS requires a Tunnel Services PIC. To configure VPLS on a router without a Tunnel Services PIC, include the `no-tunnel-services` statement:

```
no-tunnel-services;
```

You can configure this statement at the following hierarchy levels:

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

The `no-tunnel-services` statement is not supported when the core-facing interface is aggregated SONET (AS).

## Configuring Interfaces for VPLS Routing

---

On each PE router and for each VPLS routing instance, specify which interfaces are intended for the VPLS traffic traveling between PE and CE routers. To specify the interface for VPLS traffic, include the `interface` statement in the routing instance configuration:

```
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*]

You must also define each interface by including the following statements:

```
vlan-tagging;
encapsulation encapsulation-type;
unit logical-unit-number {
    vlan-id vlan-id-number;
    family vpls (Interfaces);
}
```

You can include these statements at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-routers *logical-router-name* interfaces *interface-name*]

The following sections provide enough information to enable you to configure interfaces for VPLS routing. For detailed information on configuring interfaces and the statements at the [edit interfaces] hierarchy level, see the *JUNOS Network Interfaces Configuration Guide*.

To configure an interface for VPLS, you perform the steps in the following sections:

- Configuring the Interface Name on page 387
- Configuring the Interface Encapsulation on page 388
- Enabling VLAN Tagging on page 390

### **Configuring the Interface Name**

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

*physical.logical*

For example, in `ge-1/2/1.2`, `ge-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 you configure a specific interface for VPLS routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for VPLS.

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

## Configuring the Interface Encapsulation

You need to specify an encapsulation type for each PE-router-to-CE-router interface configured for VPLS. This section describes the **encapsulation** statement configuration options available for VPLS. For a full description of all of the options available for this statement, see the *JUNOS Network Interfaces Configuration Guide*.

To configure the encapsulation type on the physical interface, include the **encapsulation** statement:

```
encapsulation (ethernet-vpls | extended-vlan-vpls | vlan-vpls);
```

You can include the **encapsulation** statement for physical interfaces at the following hierarchy levels:

- [edit interfaces *interface-name*]
- [edit logical-routers *logical-router-name* interfaces *interface-name*]

You can configure the following physical interface encapsulations for VPLS routing instances:

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard Tag Protocol ID (TPID) values. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.
- **extended-vlan-vpls**—Use extended virtual LAN (VLAN) VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

To configure the encapsulation type for logical interfaces, include the `encapsulation` statement:

```
encapsulation (ether-vpls-over-atm-llc | vlan-vpls);
```

You can include the `encapsulation` statement for logical interfaces at the following hierarchy levels:

- [edit interfaces *interface-name* unit *number*]
- [edit logical-routers *logical-router-name* interfaces *interface-name* unit *number*]

You can configure the following logical interface encapsulations for VPLS routing instances:

- **ether-vpls-over-atm-llc**—Use Ethernet VPLS over Asynchronous Transfer Mode (ATM) logical link control (LLC) encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3-encapsulated Ethernet frames with the frame check sequence (FCS) field removed. This encapsulation type is supported on ATM intelligent queuing (IQ) interfaces only.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M-series routers (except the M320), the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

When you configure the physical interface encapsulation as `vlan-vpls`, you also need to configure the same interface encapsulation for the logical interface. You need to configure the `vlan-vpls` encapsulation on the logical interface because the `vlan-vpls` encapsulation allows you to configure a mixed mode, where some of the logical interfaces use regular Ethernet encapsulation (the default for logical interfaces) and some use `vlan-vpls`. For more information, see the *JUNOS Network Interfaces Configuration Guide*.

## Enabling VLAN Tagging

The JUNOS software supports receiving and forwarding routed Ethernet frames with 802.1Q virtual local area network (VLAN) tags and running Virtual Router Redundancy Protocol (VRRP) over 802.1Q-tagged interfaces. For VPLS to function properly, configure the router to receive and forward frames with 802.1Q VLAN tags by including the `vlan-tagging` statement:

```
vlan-tagging;
```

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

- [edit interfaces *interface-name*]
- [edit logical-routers *logical-router-name* interfaces *interface-name*]

Gigabit Ethernet interfaces can be partitioned; you can assign up to 4095 different logical interfaces, one for each VLAN, but you are limited to a maximum of 1024 VLANs on any single Gigabit Ethernet or 10-Gigabit Ethernet port. Fast Ethernet interfaces can also be partitioned, with a maximum of 1024 logical interfaces for the 4-port FE PIC and 16 logical interfaces for the M40e Internet router. Table 10 lists VLAN ID range by interface type.

**Table 10: VLAN ID Range by Interface Type**

Interface Type	VLAN ID Range
Fast Ethernet	512 through 1023
Gigabit Ethernet	512 through 4094

To bind a VLAN ID to a logical interface, include the `vlan-id` statement:

```
vlan-id number;
```

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

- [edit interfaces *interface-name* unit *logical-unit-number*]
- [edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number*]

For more information on how to configure VLANs, see the *JUNOS Network Interfaces Configuration Guide*.

## Configuring the VPLS Routing Instance

---

To configure a VPLS routing instance, include the `vpls` statement:

```
vpls {
  active-interface {
    any;
    primary interface-name;
  }
  interface-mac-limit limit;
  mac-table-size size;
  no-tunnel-services;
  site site-name {
    active-interface {
      any;
      primary interface-name;
    }
    interface interface-name {
      interface-mac-limit limit;
    }
    multi-homing;
  }
  site-identifier identifier;
}
site-range number;
traceoptions {
  file filename <replace> <size size> <files number> <nostamp>;
  flag flag <flag-modifier> <disable>;
}
tunnel-services {
  devices device-names;
  primary primary-device-name;
}
}
```

You can include this statement at the following hierarchy levels:

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

The configuration for the VPLS routing instance statements is explained in the following sections:

- Configuring the VPLS Site Name and Site Identifier on page 392
- Configuring the VPLS Site Interfaces on page 392
- Configuring the Site Range on page 393
- Configuring the VPLS MAC Table Timeout Interval on page 393
- Configuring the Size of the VPLS MAC Address Table on page 394
- Limiting the Number of MAC Addresses Learned from an Interface on page 394

### Configuring the VPLS Site Name and Site Identifier

On each PE router, you must configure each VPLS site that has a connection to the PE router. All the Layer 2 circuits provisioned for a VPLS site are listed as the set of logical interfaces (using the `interface` statement) within the `site` statement.

You must configure a site name and site identifier for each VPLS site.

To configure the site name and the site identifier, include the `site` and the `site-identifier` statements:

```
site site-name {
  interface interface-name {
    interface-mac-limit limit;
  }
  site-identifier identifier;
}
```

The site identifier can be any number between 1 and 65,534 that uniquely identifies the VPLS site.

You can include these statements at the following hierarchy levels:

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

### Configuring the VPLS Site Interfaces

All the Layer 2 circuits you configure for a VPLS site are listed as a set of logical interfaces within the VPLS site configuration.

To configure a logical interface for the VPLS site, include the `interface`:

```
interface interface-name {
  interface-mac-limit limit;
}
```

You can include these statements at the following hierarchy levels:

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

You can also configure a limit on the number of MAC addresses that can be learned from the specified interface. See “Limiting the Number of MAC Addresses Learned from an Interface” on page 394 for more information.

### Configuring the Site Range

For each VPLS routing instance, you need to configure a site range. The site range specifies the total number of sites in the VPLS. To configure a site range, include the `site-range` statement:

```
site-range number;
```

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

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

### Configuring the VPLS MAC Table Timeout Interval

You can modify the timeout interval for the VPLS table. We recommend you configure longer values for small, stable VPLS networks and shorter values for large, dynamic VPLS networks. If the VPLS table does not receive any updates during the timeout interval, the router waits one additional interval before automatically clearing the MAC address entries from the VPLS table.

To modify the timeout interval for the VPLS table, include the `mac-table-aging-time` statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level:

```
mac-table-aging-time seconds;
```

You can include this statement at the following hierarchy levels:

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

### Configuring the Size of the VPLS MAC Address Table

You can modify the size of the VPLS media access control (MAC) address table. The default table size is 512 MAC addresses, the minimum is 16 addresses, and the maximum is 65,536 addresses.

If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

To change the VPLS MAC table size for each VPLS or VPN routing instance, include the `mac-table-size` statement:

```
mac-table-size size;
```

You can include the `mac-table-size` statement at the following hierarchy levels:

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

### Limiting the Number of MAC Addresses Learned from an Interface

You can configure a limit on the number of MAC addresses learned by a VPLS routing instance using the `mac-table-size` statement. If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are dropped.

Since this limit applies to each VPLS routing instance, it is possible for the MAC addresses of a single interface to consume all the available space in the table, preventing the routing instance from acquiring addresses from other interfaces.

You can limit the number of MAC addresses learned from each interface configured for a VPLS routing instance. To do so, include the `interface-mac-limit` statement:

```
interface-mac-limit limit;
```

You can include this statement at the following hierarchy levels:

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

Configuring the `interface-mac-limit` statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level causes the same limit to be applied to all of the interfaces configured for that specific routing instance.

You can also limit the number of MAC addresses learned by a specific interface configured for a VPLS routing instance. This gives you the ability to limit particular interfaces that you expect might generate a lot of MAC addresses.

To limit the number of MAC addresses learned by a specific interface, include the `interface-mac-limit` statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols vpls site *site-name* interfaces *interface-name*]

The MAC limit configured for an individual interface at this hierarchy level overrides any value configured at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level. Also, the MAC limit configured using the `mac-table-size` statement can override the limit configured using the `interface-mac-limit` statement.

The MAC address limit applies to customer facing interfaces only.

## Configuring an Ethernet Switch as the CE Device

---

For VPLS configurations, the CE device does not necessarily need to be a router. You can link the PE routers directly to Ethernet switches. However, there are a few configuration issues you should be aware of:

- When you configure VPLS routing instances and establish two or more connections between a CE Ethernet switch and a PE router, you must enable the Spanning Tree Protocol (STP) on the switch to prevent loops.
- The JUNOS software allows standard Bridge Protocol Data Unit (BPDU) frames to pass through emulated Layer 2 connections, such as those configured with Layer 2 VPNs, Layer 2 circuits, and VPLS instances. However, CE Ethernet switches that generate proprietary BPDU frames might not be able to run STP across Juniper Networks routing platforms configured for these emulated Layer 2 connections.

## Mapping VPLS Traffic to a Specific LSP

You can map VPLS traffic to specific LSPs by configuring forwarding table policies. This procedure is optional but can be useful. The following example illustrates how you can map lower priority VPLS routing instances to slower LSPs while mapping other higher priority VPLS routing instances to faster LSPs. In this example configuration, **a-to-b1** and **a-to-c1** are high-priority LSPs between the PE routers, while **a-to-b2** and **a-to-c2** are low-priority LSPs between the PE routers.

To map VPLS traffic, include the `policy-statement vpls-priority` statement:

```

policy-statement vpls-priority {
  term a {
    from {
      rib mpls.0;
      community company-1;
    }
    then {
      install-nexthop lsp [ a-to-b1 a-to-c1 ];
      accept;
    }
  }
  term b {
    from {
      rib mpls.0;
      community company-2;
    }
    then {
      install-nexthop lsp-regex [ "^a-to-b2$" "^a-to-c2$" ];
      accept;
    }
  }
}
community company-1 members target:11111:1;
community company-2 members target:11111:2;

```

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

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

Include the `export vpls-priority` statement in the `forwarding-table` statement configuration:

```
forwarding-table {  
    export vpls-priority;  
}
```

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

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

For more information on how to configure routing policies, see the *JUNOS Policy Framework Configuration Guide*.

## Configuring VPLS Filters and Policers

---

You can configure both firewall filters and policers for VPLS. Firewall filters allow you to filter packets based on their components and to perform an action on packets that match the filter. Policers allow you to limit the amount of traffic that passes into or out of an interface.

You can apply VPLS filters and policers on the PE router to customer-facing interfaces only.

The following sections explain how to configure filters and policers for VPLS:

- Configuring a VPLS Filter on page 398
- Configuring a VPLS Policer on page 402

## Configuring a VPLS Filter

To configure a filter for VPLS, include the `filter` statement at the `[edit firewall family vpls]` hierarchy level:

```
[edit firewall family vpls]
filter filter-name {
  interface-specific;
  term term-name {
    from {
      match-conditions;
    }
    then {
      actions;
    }
  }
}
```

To configure a filter for VPLS traffic, you complete the following tasks:

- Configuring an Interface-Specific Counter for VPLS on page 399
- Configuring the VPLS Filter Match Conditions on page 399
- Configuring an Action for the VPLS Filter on page 400
- Configuring VPLS FTFs on page 400
- Changing Precedence for Spanning Tree BPDU Packets on page 400
- Applying a VPLS Filter to an Interface on page 401
- Applying a VPLS Filter to a VPLS Routing Instance on page 401
- Configuring a Filter for Flooded Traffic on page 402

For more information on how to configure firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

### Configuring an Interface-Specific Counter for VPLS

When you configure a firewall filter for VPLS and apply it to multiple interfaces, you can specify individual counters specific to each interface. This allows you to collect separate statistics on the traffic transiting each interface.

To generate an interface-specific counter for VPLS, you configure the **interface-specific** statement. A separate instantiation of the filter is generated. This filter instance has a different name (based on the interface name) and collects statistics on the interface specified only.

To configure interface-specific counters, include the **interface-specific** statement at the `[edit firewall family vpls filter filter-name]` hierarchy level:

```
[edit firewall family vpls filter filter-name]
  interface-specific;
```



**NOTE:** The counter name is restricted to 24 bytes. If the renamed counter exceeds this maximum length, it might be rejected.

---

For more information on the **interface-specific** statement and an example of how to configure it, see the *JUNOS Policy Framework Configuration Guide*.

### Configuring the VPLS Filter Match Conditions

In the **from** statement in the VPLS filter term, you specify conditions that the packet must match for the action in the **then** statement to be taken. All conditions in the **from** statement must match for the action to be taken. The order in which you specify match conditions is not important, because a packet must match all the conditions in a term for a match to occur.

If you specify no match conditions in a term, that term matches all packets.

An individual condition in a **from** statement can contain a list of values. For example, you can specify numeric ranges or multiple source or destination addresses. When a condition defines a list of values, a match occurs if one of the values in the list matches the packet.

Individual conditions in a **from** statement can be negated. When you negate a condition, you are defining an explicit mismatch. For example, the negated match condition for **forwarding-class** is **forwarding-class-except**. If a packet matches a negated condition, it is immediately considered not to match the **from** statement, and the next term in the filter is evaluated, if there is one; if there are no more terms, the packet is discarded.

To specify the match conditions for a VPLS filter term, include the **from** statement at the `[edit firewall family vpls filter filter-name term term-name]` hierarchy level. Table 11 on page 400 describes the match conditions available for VPLS filters.

```
[edit firewall family vpls filter filter-name term term-name]
  from match-conditions;
```

**Table 11: VPLS Filter Match Conditions**

Match Condition	Description
<code>destination-mac-address</code> <i>mac-address</i>	Specified destination MAC address.
<code>ether-type</code> <i>value</i>	Ethernet packets. Configure the <code>ether-type</code> match condition when the encapsulation of the associated interfaces is <code>ethernet-vpls</code> .
<code>forwarding-class</code> <i>value</i>	Specified forwarding class.
<code>interface-group</code> <i>index</i>	Interface group on which the packet was received. An interface group is a set of one or more logical interfaces.
<code>source-mac-address</code> <i>mac-address</i>	Source MAC address.
<code>vlan-ether-type</code> <i>value</i>	VLAN Ethernet packets. Configure the <code>vlan-ether-type</code> match condition when the encapsulation of the associated interfaces is either <code>vlan-vpls</code> or <code>extended-vlan-vpls</code> .

### Configuring an Action for the VPLS Filter

You can configure the following actions for a VPLS filter at the [edit firewall family vpls filter *filter-name* term *term-name* then] hierarchy level: `accept`, `count`, `discard`, `forwarding-class`, `loss-priority`, `next`, `policer`.

### Configuring VPLS FTFs

Forwarding table filters (FTFs) are filters configured for forwarding tables. For VPLS, they are attached to the destination MAC (DMAC) forwarding table of the VPLS routing instance. You define VPLS FTFs in the same manner as any other type of FTF. You can only apply a VPLS FTF as an input filter.

To specify a VPLS FTF, include the `filter input` statement:

```
filter input filter-name;
```

You can include the `filter input` statement at the following hierarchy levels:

- [edit routing-instance *routing-instance-name* forwarding-options family vpls]
- [edit logical-routers *logical-router-name* routing-instance *routing-instance-name* forwarding-options family vpls]

For the statement summaries of these statements, see the *JUNOS Policy Framework Configuration Guide*.

### Changing Precedence for Spanning Tree BPDU Packets

Spanning tree BPDU packets are automatically set to a high precedence. The queue number on these packets is set to 3. On M-series routers (except the M320) by default, a queue value of 3 indicates high precedence. To enable this higher precedence on BPDU packets, an instance-specific BPDU precedence filter named `default_bpdu_filter` is automatically attached to the VPLS DMAC table. This filter places a high precedence on all packets sent to `01:80:c2:00:00:00/24`.

You can overwrite this filter by configuring a VPLS FTF filter and applying it to the VPLS routing instance. For more information, see “Configuring VPLS FTFs” on page 400 and “Applying a VPLS Filter to a VPLS Routing Instance” on page 401.

### Applying a VPLS Filter to an Interface

To apply a VPLS filter to an interface, include the `filter` statement:

```
filter {
    input input-filter-name;
    output output-filter-name;
    group index;
}
```

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

- [edit interfaces *interface-name* unit *number* family vpls]
- [edit logical-routers *logical-router-name* interfaces *interface-name* unit *number* family vpls]

In the `input` statement, list the name of the VPLS filter to be evaluated when packets are received on the interface. In the `output` statement, list the name of the VPLS filter to be evaluated when packets are transmitted on the interface.



**NOTE:** For output interface filters, MAC addresses are learned after the filter action is completed. When an output interface filter's action is `discard`, the packet is dropped before the MAC address is learned. However, an input interface filter would learn the MAC address before discarding the packet.

---

For the statement summaries for these statements, see the *JUNOS Network Interfaces Configuration Guide*.

### Applying a VPLS Filter to a VPLS Routing Instance

You can apply a VPLS filter to a VPLS routing instance. The filter checks traffic passing through the specified routing instance.

Input routing instance filters learn the MAC address before the filter action is completed, so if the filter action is `discard`, the MAC address is learned before the packet is dropped.

To apply a VPLS filter to packets arriving at a VPLS routing instance and specify the filter, include the `filter input` statement:

```
filter input input-filter-name;
```

You can include the `filter input` statement at the following hierarchy levels:

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

### Configuring a Filter for Flooded Traffic

You can configure a VPLS filter to filter flooded packets. CE routers typically flood the following types of packets to PE routers in VPLS routing instances:

- Layer 2 broadcast packets
- Layer 2 multicast packets
- Layer 2 unicast packets with an unknown destination MAC address
- Layer 2 packets with a MAC entry in the DMAC routing table

You can configure filters to manage how these flooded packets are distributed to the other PE routers in the VPLS routing instance.

To apply a flooding filter to packets arriving at the PE router in the VPLS routing instance, and specify the filter, include the `flood input` statement:

```
flood input filter-name;
```

You can include the `flood input` statement at the following hierarchy levels:

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

### Configuring a VPLS Policer

You can configure a policer for VPLS traffic. The VPLS policer configuration is similar to the configuration of any other type of policer.

VPLS policers have the following characteristics:

- You cannot police the default VPLS routes stored in the flood table from PE router-sourced flood traffic.
- When specifying policing bandwidth, the VPLS policer considers all Layer 2 bytes in a packet to determine the packet length.

To configure a VPLS policer, include the `policer` statement at the [edit firewall] hierarchy level:

```
[edit firewall]
policer name {
    bandwidth-limit limit;
    burst-size-limit limit;
    then action;
}
```

For the statement summaries of these statements and more information on how to configure policers, see the *JUNOS Policy Framework Configuration Guide*.

To apply a VPLS policer to an interface, include the `policer` statement:

```
policer {
    input input-policer-name;
    output output-policer-name;
}
```

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

- [edit interfaces *interface-name* unit *number* family vpls]
- [edit logical-routers *logical-router-name* interfaces *interface-name* unit *number* family vpls]

In the `input` statement, list the name of the VPLS policer to be evaluated when packets are received on the interface. In the `output` statement, list the name of the VPLS policer to be evaluated when packets are transmitted on the interface.

For the statement summaries for these statements, see the *JUNOS Network Interfaces Configuration Guide*.

## Specifying the VT Interfaces Used by VPLS Routing Instances

---

By default, the JUNOS software automatically selects one of the virtual tunnel (VT) interfaces available to the router for de-encapsulating traffic from a remote site. The JUNOS software cycles through the currently available VT interfaces, regularly updating the list of available VT interfaces as new remote sites are discovered and new connections are brought up. However, you can also explicitly configure which VT interfaces will receive the VPLS traffic.

By configuring the `tunnel-services` statement at the [edit routing-instances *routing-instance-name* protocols vpls] hierarchy level, you can specify that traffic for particular VPLS routing instances be forwarded to specific VT interfaces. Doing so allows you to load-balance VPLS traffic among all the available VT interfaces on the router.

The `tunnel-services` statement includes the following options:

- **devices**—Specifies the VT interfaces acceptable for use by the VPLS routing instance. If you do not configure this option, all VT interfaces available to the router can be used for de-encapsulating traffic for this instance.
- **primary**—Specifies the primary VT interface to be used by the VPLS routing instance. The VT interface specified is used to de-encapsulate all VPLS traffic from the MPLS core network for this routing instance. If the VT interface specified is unavailable, then one of the other acceptable VT interfaces (specified in the `devices` option) is used for handling the VPLS traffic. If you do not configure this option, any acceptable VT interface can be used to de-encapsulate VPLS traffic from the core.

To specify that traffic for a particular VPLS routing instance be forwarded to specific VT interfaces, include the `tunnel-services` statement:

```
tunnel-services {  
    devices device-names;  
    primary primary-device-name;  
}
```

These statements can be configured at the following hierarchy levels:

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

## Configuring VPLS Multihoming

---

VPLS multihoming allows you to connect a customer site to multiple PE routers to provide redundant connectivity while preventing the formation of Layer 2 loops in the service provider's network. A VPLS site multihomed to two or more PE routers provides redundant connectivity in the event of a PE router-to-CE device link failure or the failure of a PE router. For more information on VPLS multihoming, see "VPLS Multihoming" on page 380.

The following sections describe how to configure VPLS multihoming. Some information is also provided on single-homed site configuration versus multihomed site configuration.

- VPLS Multihomed Site Configuration on page 405
- VPLS Single-Homed Site Configuration on page 406

## VPLS Multihomed Site Configuration

The following describes the requirements for a VPLS multihomed site configuration:

- Assign the same site ID on all PE routers connected to the same CE devices.
- Assign the same route distinguisher on all PE routers connected to the same CE devices.
- Reference all interfaces assigned to the multihomed VPLS site on each PE router. Only one of these interfaces is used to send and receive traffic for this site at a time.
- Either designate a primary interface or allow the router to select the interface to be used as the primary interface.

If the router selects the interface, the interface used to connect the PE router to the site depends on the order in which interfaces are listed in the PE router's configuration. The first operational interface in the set of configured interfaces is chosen to be the designated interface. If this interface fails, the next interface in the list is selected to send and receive traffic for the site.

- Configure multihoming for the site.

The following configuration shows the statements you need to configure to enable VPLS multihoming:

```
[edit routing-instances routing-instance-name]
instance-type vpls;
interface interface-name;
interface interface-name;
protocols vpls {
  site site-name {
    active-interface {
      any;
      primary interface-name;
    }
    site-identifier number;
    interface interface-name;
    interface interface-name;
  }
}
route-distinguisher (as-number:id | ip-address:id);
```

Most of these statements are explained in more detail in the rest of this chapter. The following sections explain how to configure the statements that are specific to VPLS multihoming:

- Specifying an Interface as the Active Interface on page 406
- Configuring Multihoming on the PE Router on page 406



**NOTE:** If you add a direct connection between CE devices that are multihomed to the same VPLS site on different PE routers, traffic loops and loss of connectivity might occur. We do not recommend this topology.

### Specifying an Interface as the Active Interface

You need to specify one of the interfaces for the multihomed site as the primary interface. If there are multiple interfaces, the remaining interfaces are activated only when the primary interface goes down. If no active interfaces are configured at the site level, all traffic for a VPLS site travels through a single, non-multihomed PE router.

You must configure one of the following options for the `active-interface` statement:

- *any*—One configured interface is randomly designated as the active interface for the VPLS site.
- *primary*—Specify the name of the multihomed interface to be used as the primary interface by the VPLS site.

To specify a multihomed interface as the primary interface for the VPLS site, include the `active-interface` statement:

```
active-interface {
    any;
    primary interface-name;
}
```

You can configure this statement at the following hierarchy levels:

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

### Configuring Multihoming on the PE Router

When a CE device is connected to the same VPLS site on more than one PE router, include the `multi-homing` statement on all associated PE routers. Configuration of this statement tracks BGP peers. If no BGP peer is available, VPLS deactivates all active interfaces for a site. To specify that the PE router is part of a multihomed VPLS site, include the `multi-homing` statement:

```
multi-homing;
```

You can configure this statement at the following hierarchy levels:

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

Include the `multi-homing` statement on all PE routers associated with a particular VPLS site.

### VPLS Single-Homed Site Configuration

All VPLS single-homed sites are connected to the same default VE device. All interfaces in a VPLS routing instance that are not configured as part of a multihomed site are assumed to be single-homed to the default VE device.

## Tracing VPLS Traffic and Operations

---

To trace VPLS traffic, you can specify options using the `traceoptions` statement:

```
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
    (<world-readable> | <no-world-readable>);
  flag flag <flag-modifier> <disable>;
}
```

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

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

The following trace flags display the operations associated with VPLS:

- `all`—All VPLS tracing options
- `connections`—VPLS connections (events and state changes)
- `error`—Error conditions
- `nlri`—VPLS advertisements received or sent using BGP
- `route`—Trace-routing information
- `topology`—VPLS topology changes caused by reconsideration or advertisements received from other PE routers using BGP

