VYATTA, INC. | Vyatta System

LAN Interfaces

REFERENCE GUIDE

Loopback Interface
Ethernet Interfaces
Ethernet Link Bonding Interfaces
Pseudo-Ethernet Interfaces
Wireless Interfaces
VLAN Interfaces
Input Interfaces

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

COPYRIGHT

PROPRIETARY NOTICES

RELEASE DATE: January 2013

DOCUMENT REVISION. 6.6R0S1 d01

RELEASED WITH: 6.6R0S1

PART NO. A0-0211-10-0018

Contents

Quick List of Commands

Use this list to help you quickly locate commands.

List of Examples

Use this list to help you locate examples you'd like to look at or try.

Preface

This document describes the system interfaces supported on the Vyatta router for local area network (LAN) connections.

This preface provides information about using this guide. The following topics are presented:

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP services

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- Quick List of Commands

  Use this list to help you quickly locate commands.

- List of Examples

  Use this list to help you locate examples you'd like to try or look at.

This guide has the following chapters:

Chapter	Description	Page
Chapter 1: Loopback Interface	This chapter explains how to work with the Vyatta system's software loopback interface.	1
Chapter 2: Ethernet Interfaces	This chapter describes basic configuration for Ethernet interfaces.	15
Chapter 3: Ethernet Link Bonding Interfaces	This chapter explains how to bond Ethernet links into a larger virtual link.	79
Chapter 4: Pseudo-Ethernet Interfaces	This chapter describes how to create a pseudo-Ethernet interface by defining multiple MAC addresses on a single physical interface.	125
Chapter 5: Wireless Interfaces	This chapter explains how to work with wireless interfaces on the Vyatta system.	157

Document Conventions

This guide uses the following advisory paragraphs, as follows.

WARNING *Warnings alert you to situations that may pose a threat to personal safety.*

CAUTION *Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service.*

NOTE *Notes provide information you might need to avoid problems or configuration errors.*

This document uses the following typographic conventions.

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[key1 | key2]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.

arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[*arg*...] *arg*[,*arg*...]	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

Vyatta Publications

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Additional information is available on www.vyatta.com and www.vyatta.org.

Chapter 1: Loopback Interface

This chapter explains how to work with the Vyatta system's software loopback interface.

This chapter presents the following topics:

Loopback Interface Overview

The loopback interface is a special software-only interface that emulates a physical interface and allows the router to "connect" to itself. Packets routed to the loopback interface are rerouted back to the router and processed locally. Packets routed out the loopback interface but not destined for the loopback interface are dropped.

The loopback interface provides a number of advantages:

- As long as the router is functioning, the loopback interface is always up, and so is very reliable. As long as there is even one functioning link to the router, the loopback interface can be accessed. The loopback interface thus eliminates the need to try each IP address of the router until you find one that is still up.
- Because the loopback interface is always up, a routing session (such as a BGP session) can continue even if the outbound interface fails.
- You can simplify collection of management information by specifying the loopback interface as the interface for sending and receiving management information such as logs and SNMP traps.
- The loopback interface can be used to increase security, by filtering incoming traffic using access control rules that specify the local interface as the only acceptable destination.
- In OSPF, you can advertise a loopback interface as an interface route into the network, regardless of whether physical links are up or down. This increases reliability by allowing traffic to take alternate paths if one or more physical links go down.
- In BGP, parallel paths can be configured to the loopback interface on a peer device. This provides improved load sharing and redundancy.

The router automatically creates the loopback interface on startup, with an interface name of **lo**. It also automatically configures the loopback address with standard IP addressing:

- As per RFC 5735, the IPv4 address 127.0.0.1/8 is assigned to the loopback address. Typically, the IPv4 address assigned to the loopback device is 127.0.0.1 for IPv4, although any address in the range 127.0.0.0 to 127.255.255.255 is mapped to it.
- As per RFC 3513, the IPv6, IP address ::1/128 is assigned to the loopback interface.
- As per RFC 2606, the domain name **localhost** is mapped to the loopback addresses.

When configuring the router, it is good practice to take advantage of the loopback interface's reliability:

- The router's hostname should be mapped to the loopback interface address, rather than to a physical interface.
- In OSPF and BGP, the router ID should be set to the loopback address. This will prevent a possible dynamic recalculation and reassignment of the loopback address when physical interfaces are added or removed from the system. This action will be disruptive to active BGP and OSPF sessions.

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on the loobback interface are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

Loopback Interface Configuration Examples

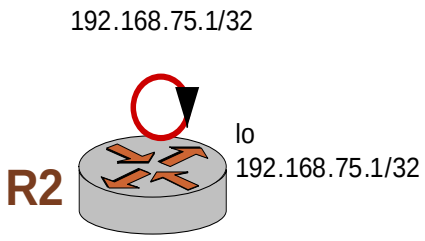
This section presents the following topics:

Configuring Network Addresses

The system automatically creates and addresses the loopback interface, so you need not configure any additional addressing. If you delete the loopback node, the system recreates and re-addresses the loopback address again the next time the system starts.

There may be times when you want to configure a smaller network prefix than /8 to the loopback interface. The example in this section assigns the address 192.168.75.1/32 to the loopback interface. When you have finished, the interface will be configured as in Figure 1-1.

Figure 1-1 Configuring the loopback interface



To configure the loopback interface, perform the following steps in configuration mode.

Example 1-1 Configuring the loopback interface

Step	Command
Assign the IP address to the loopback interface.	`vyatta@R1# set interfaces loopback lo address 192.168.75.1/32`
Commit the configuration.	`vyatta@R1# commit`

View the configuration.	`vyatta@R1# show interfaces loopback`
	`loopback lo {`
	`  address 192.168.75.1/32`
	`}`

IPv6 on the Loopback Interface

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Loopback Interface Commands

Configuration Commands	
interfaces loopback lo	Defines the loopback interface.
interfaces loopback lo address	Sets an IP address and network prefix for the loopback interface.
interfaces loopback lo description <descr>	Allows you to record a description for the loopback interface.
interfaces loopback lo redirect <interface>	Redirects inbound traffic from from one interface to another interface.
Operational Commands	
clear interfaces loopback counters	Clears statistics counters for loopback interfaces.
show interfaces loopback	Displays information about the loopback interface.

Commands for using other system features with loopback interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
OSPF	OSPF is supported on the loopback interface. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
QoS	QoS traffic policies are supported on the loopback interface. Commands for configuring quality of service on the loopback interface are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on the loopback interface. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on the loopback interface. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.

clear interfaces loopback counters

Clears statistics counters for loopback interfaces.

Syntax

clear interfaces loopback [**lo**] **counters**

Command Mode

Operational mode.

Parameters

lo	Optional. Clears statistics for the loopback interface only. Equivalent to using the command with no option.

Default

Clears all counters for the loopback interface.

Usage Guidelines

Use this command to clear counters on the loopback interface.

interfaces loopback lo

Defines the loopback interface.

Syntax

set interfaces loopback lo

delete interfaces loopback lo

show interfaces loopback

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   loopback lo
}
```

Parameters

None.

Default

A configuration node is automatically created for the loopback interface on startup.

Usage Guidelines

Use this command to configure the loopback interface.

You can use the **set** form of this command to create the loopback interface. However, the system automatically creates a configuration node for the loopback interface on startup, so you should not need to use the **set** form of this command to create the loopback interface unless you have deleted it.

Use the **delete** form of this command to remove all configuration for the loopback interface. The system will create an empty configuration node for the interface the next time the system starts.

Use the **show** form of this command to view loopback interface configuration.

interfaces loopback lo address

Sets an IP address and network prefix for the loopback interface.

Syntax

set interfaces loopback lo address {*ipv4* | *ipv6*}

delete interfaces loopback lo address {*ipv4* | *ipv6*}

show interfaces loopback lo address

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    loopback lo {
        address [ipv4|ipv6]
    }
}
```

Parameters

ipv4	An IPv4 address and network prefix for this interface. The format is *ip-address/prefix* (for example, 127.0.0.1/8). You can define multiple IP addresses for the loopback interface by creating multiple **address** configuration nodes.
ipv6	An IPv6 address and network prefix for this interface. The format is *ipv6-address/prefix* (for example, ::1/128). You can define multiple IPv6 addresses for a single interface, by creating multiple **address** configuration nodes.

Default

None.

Usage Guidelines

Use the **set** form of this command to specify the IP address and network mask for the loopback interface. You can set more than one IP address for the loopback interface by creating multiple **address** configuration nodes.

Use the **delete** form of this command to remove the loopback interface address.

Use the **show** form of this command to view loopback interface address configuration.

interfaces loopback lo description <descr>

Allows you to record a description for the loopback interface.

Syntax

set interfaces loopback lo description *descr*

delete interfaces loopback lo description

show interfaces loopback lo description

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    loopback lo {
        description descr
    }
}
```

Parameters

descr	A description for the loopback interface.

Default

None.

Usage Guidelines

Use this command to set a description for the loopback interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

interfaces loopback lo redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces loopback lo redirect *interface*

delete interfaces loopback lo redirect *interface*

show interfaces loopback lo redirect

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    loopback lo {
        redirect interface
    }
}
```

Parameters

interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

show interfaces loopback

Displays information about the loopback interface.

Syntax

show interfaces loopback [**lo** [**brief**] | **detail**]

Command Mode

Operational mode.

Parameters

lo	Displays detailed statistics and configuration information for the loopback interface.
brief	Displays brief status information for the loopback interface. Enter the **show interfaces loopback** command with the **lo brief** option is equivalent to entering it with no option.
detail	Displays detailed information and statistics about the loopback interface.

Default

When used with no option, this command displays brief status information for the loopback interface.

Usage Guidelines

Use this command to view information and status for the loopback interface.

The **show interfaces loopback** and **show interfaces loopback lo brief** commands are equivalent.

The **show interfaces loopback lo** and **show interfaces loopback detail** commands are equivalent.

Examples

Example 1-2 shows brief status information for the loopback interface.

Example 1-2 Displaying loopback interface information.

```
vyatta@R1:~$ show interfaces loopback
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
lo               127.0.0.1/8                       u/u
                 ::1/128
```

Example 1-3 shows additional detail for the loopback interface.

Example 1-3 Displaying additional loopback interface detail.

```
vyatta@R1:~$ show interfaces loopback lo
lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
         44056        730          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
         44056        730          0          0          0          0
```

Chapter 2: Ethernet Interfaces

This chapter describes basic configuration for Ethernet interfaces.

This chapter presents the following topics:

Ethernet Interfaces Overview

A system receives packets from neighboring systems through its network interfaces. On the LAN, this interface is typically an Ethernet interface.

Ethernet interfaces are viewed and configured in the interfaces ethernet node of the configuration tree. The system automatically discovers the physical interfaces on the system and creates entries for them on startup in the configuration tree. For example, on a system with two Ethernet interfaces, the router automatically creates configuration nodes for eth0 and eth1.

Once the interface is enabled and provided with an address, you can configure it with various system features—such as firewall, routing protocols, Quality of Service, and so on.

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on Ethernet interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

Ethernet Interface Configuration Examples

This section presents the following topics:

Viewing System Interfaces

You can only configure interfaces that actually are physically available to the operating system on the hardware you are using. To view all the interfaces known to the operating system, use the **show interfaces system** command in operational mode, as shown in Example 2-1. In this example, the system has two physical Ethernet interfaces, eth0 and eth1, plus a VLAN interface (vif) configured for VLAN 40 under eth1.

Example 2-1 Viewing available system interfaces

```
vyatta@vyatta:~$ show interfaces system
vyatta@R1:~$ show interfaces system
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 3a:26:db:4d:63:a2 brd ff:ff:ff:ff:ff:ff
```

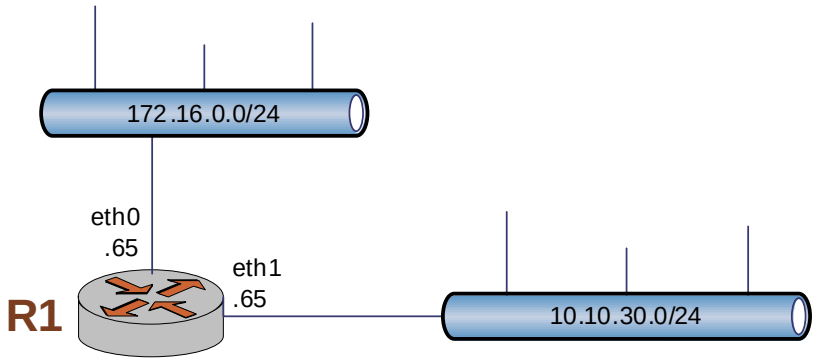
```
    inet 192.168.1.81/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::3826:dbff:fe4d:63a2/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
      18512372     189264          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
        867299       7494          0          0          0          0

eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether aa:31:57:a8:ee:90 brd ff:ff:ff:ff:ff:ff
    inet 10.10.30.65/24 brd 10.10.30.255 scope global eth1
    inet6 fe80::a831:57ff:fea8:ee90/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
             0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
           776          8          0          0          0          0

eth1.40@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP
    link/ether aa:31:57:a8:ee:90 brd ff:ff:ff:ff:ff:ff
    inet 10.10.40.65/24 brd 10.10.40.255 scope global eth1.40
    inet6 fe80::a831:57ff:fea8:ee90/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
             0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
           368          4          0          0          0          0

lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
      11170696     186174          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
      11170696     186174          0          0          0          0
```

Basic Ethernet Interface Configuration

This section presents a sample configuration for an Ethernet interface connected to an Ethernet LAN.

When you have finished, the system will be configured as shown in Figure 2-1.

Figure 2-1 Basic Ethernet configuration



Example 2-2 applies IP addresses directly to the two Ethernet interfaces already discovered for the system—eth0 and eth1. These interfaces were automatically created by the system on startup, when the system detected the physical interfaces. In.

Each IP address is applied directly to the interface. The system automatically discovers the MAC address (hardware ID) of the Network Interface Card (NIC) housing the Ethernet interface, and applies default values for a number of other options.

To configure these interfaces, perform the following steps in configuration mode.

Example 2-2 Configuring Ethernet interfaces

Step	Command
Assign an IP address to interface eth0.	`vyatta@R1# set interfaces ethernet eth0 address 176.16.0.65/24`
Assign an IP address to interface eth1.	`vyatta@R1# set interfaces ethernet eth1 address 10.10.30.65/24` `[edit]`

Example 2-2 Configuring Ethernet interfaces

Commit and view the configuration.	`vyatta@R1# commit`
	`OK`
	`[edit]`
	`vyatta@R1# show interfaces ethernet`
	` eth0 {`
	`  address 176.16.0.65/24`
	`  duplex auto`
	`  hw-id 3a:26:db:4d:63:a2`
	`  smp_affinity auto`
	`  speed auto`
	` }`
	` eth1 {`
	`  address 10.10.30.65/24`
	`  duplex auto`
	`  hw-id 00-14-22-01-5c-6f`
	`  smp_affinity auto`
	`  speed auto`
	` }`
	`[edit]`

Ethernet Multinetting

Each physical interface can have multiple IP addresses assigned to it.If you want to have multiple networks on the same physical interface (called multinetting), but you don't want to use 802.1Q VLANs, simply create multiple **address** configuration nodes directly under the primary interface.

To configure Ethernet multinetting, perform the following steps in configuration mode.

Example 2-3 Configuring Ethernet multinetting

Step	Command
Assign the first IP address to eth0.	`vyatta@R2# set interfaces ethernet eth0 address 172.16.0.65/24 [edit]`
Assign the second IP address to eth0.	`vyatta@R1# set interfaces ethernet eth0 address 192.168.1.17/24` `[edit]`

Example 2-3 Configuring Ethernet multinetting

Commit and view the configuration.	`vyatta@R1# commit`
	`OK`
	`[edit]`
	`vyatta@Test1# show interfaces ethernet eth0`
	` address 192.168.1.81/24`
	` address 192.168.1.17/24`
	` duplex auto`
	` hw-id 3a:26:db:4d:63:a2`
	` smp_affinity auto`
	` auto`
	`[edit]`

IPv6 on Ethernet Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Ethernet Interface Commands

Configuration Commands	
interfaces ethernet <ethx> address	Sets an IP address and network prefix for an Ethernet interface.
interfaces ethernet <ethx> bond-group <bondx>	Adds an Ethernet interface to a bonding group.
interfaces ethernet <ethx> description <descr>	Specifies a description for an Ethernet interface.
interfaces ethernet <ethx> dhcpv6-options	Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.
interfaces ethernet <ethx> disable	Disables an Ethernet interface without discarding configuration.
interfaces ethernet <ethx> disable-flow-control	Disables IEEE 802.3x full-duplex flow control (“PAUSE” frames).
interfaces ethernet <ethx> disable-link-detect	Directs an Ethernet interface not to detect physical link-state changes.
interfaces ethernet <ethx> duplex <duplexity>	Sets the duplex mode for an Ethernet interface.
interfaces ethernet <ethx> hw-id <mac-addr>	Associates the Ethernet interface name with a hardware MAC address.
interfaces ethernet <ethx> ip enable-proxy-arp	Enables proxy ARP on an Ethernet interface.
interfaces ethernet <ethx> ipv6 address	Assigns an IPv6 address to an Ethernet interface.
interfaces ethernet <ethx> ipv6 disable-forwarding	Disables IPv6 forwarding on an Ethernet interface.
interfaces ethernet <ethx> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces ethernet <ethx> ipv6 router-advert	Specifies the router advertisements to be sent from an Ethernet interface.
interfaces ethernet <ethx> mac <mac-addr>	Sets the MAC address of an Ethernet interface.
interfaces ethernet <ethx> mirror <interface>	Mirrors inbound traffic from an Ethernet interface to another interface.
interfaces ethernet <ethx> mtu <mtu>	Sets the MTU for an Ethernet interface.
interfaces ethernet <ethx> redirect <interface>	Redirects inbound traffic from from one interface to another interface.

interfaces ethernet <ethx> smp_affinity	Sets the symmetric multiprocessor affinity for an Ethernet interface.
interfaces ethernet <ethx> speed <speed>	Sets the speed of an Ethernet interface.
Operational Commands	
clear interfaces ethernet counters	Clears statistics counters for Ethernet interfaces.
monitor interfaces ethernet <ethx> traffic	Displays (captures) traffic on an Ethernet interface.
show interfaces ethernet	Displays information and statistics about Ethernet interfaces.
show interfaces ethernet detail	Displays detailed information about Ethernet interfaces.
show interfaces ethernet <ethx> brief	Displays a brief status for an Ethernet interface.
show interfaces ethernet <ethx> identify	Blinks the LEDs on an Ethernet interface in order to identify it.
show interfaces ethernet <ethx> physical	Displays physical layer information for Ethernet interfaces.
show interfaces ethernet <ethx> queue	Displays Ethernet queuing information.
show interfaces ethernet <ethx> statistics	Displays Ethernet statistics.

Commands for using other system features with Ethernet interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Bridging	Layer 2 bridging is supported on Ethernet interfaces. Commands for configuring bridge groups are described in the *Vyatta Bridging Reference Guide*.
Firewall	Firewall is supported on Ethernet interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.
Link Bonding	Ethernet link bonding is supported for Ethernet interfaces. Commands for configuring Ethernet link bonding are described in "Chapter 3: Ethernet Link Bonding Interfaces."
OSPF	OSPF is supported on Ethernet interfaces. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on Ethernet interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
PPPoE	PPPoE encapsulation is supported on Ethernet interfaces. Commands for configuring PPP-based encapsulation are described in the *Vyatta PPP-Based Encapsulations Reference Guide.*

QoS	Quality of service traffic policies are supported on Ethernet interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on Ethernet interfaces. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on Ethernet interfaces. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the *Vyatta Basic System Reference Guide*.
VLAN interfaces	802.1Q VLAN operation is supported on Ethernet interfaces. Commands for configuring VLAN interfaces (vifs) are described in "Chapter 6: VLAN Interfaces."
VRRP	VRRP is supported on Ethernet interfaces and on VLAN interfaces configured under Ethernet interfaces. Commands for configuring VRRP are described in the *Vyatta High Availability Reference Guide*.

clear interfaces ethernet counters

Clears statistics counters for Ethernet interfaces.

Syntax

clear interfaces ethernet [*ethx*] **counters**

Command Mode

Operational mode.

Parameters

ethx	Clears statistics for the specified Ethernet interface.

Default

Clears counters for all Ethernet interfaces.

Usage Guidelines

Use this command to clear counters on Ethernet interfaces.

interfaces ethernet <ethx>

Defines an Ethernet interface.

Syntax

set interfaces ethernet *ethx*

delete interfaces ethernet *ethx*

show interfaces ethernet *ethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet ethx {
   }
}
```

Parameters

ethx	Multi-node. The identifier for the Ethernet interface you are defining. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system. These configuration nodes are automatically created by the system by detecting the physical Ethernet ports available on the system. There will be as many Ethernet interface configuration nodes created as there are physical Ethernet ports on your system.

Default

Configuration nodes are created for all available physical Ethernet interfaces on startup.

Usage Guidelines

Use this command to configure an Ethernet interface.

You can use the **set** form of this command to create an Ethernet interface, provided the interface physically exists on your system. However, the system automatically creates a configuration node for each system interface, so you should not need to use the set form of this command to create an Ethernet interface unless you have deleted it.

To see the interfaces available to the system kernel, use the **system** option of the **show interfaces** command.

Use the **delete** form of this command to remove all configuration for an Ethernet interface. The system will create an empty configuration node for the interface the next time the system starts.

Use the **show** form of this command to view Ethernet interface configuration.

interfaces ethernet <ethx> address

Sets an IP address and network prefix for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

delete interfaces ethernet *ethx* **address** [*ipv4* | *ipv6* | **dhcp** | **dhcpv6**]

show interfaces ethernet *ethx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        address [ipv4|ipv6|dhcp|dhcpv6]
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
ipv4	Defines an IPv4 address on this interface. The format is *ip-address/prefix* (for example, 192.168.1.77/24). You can define multiple IP addresses for a single interface, by creating multiple **address** configuration nodes.
ipv6	Defines an IPv6 address on this interface. The format is *ipv6-address/prefix* (for example, 2001:db8:1234::/48). You can define multiple IPv6 addresses for a single interface, by creating multiple **address** configuration nodes.
dhcp	Defines the interface as a Dynamic Host Configuration Protocol (DHCP) client, which obtains its address and prefix from a DHCP server.
dhcpv6	Defines the interface as a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) client, which obtains its address, prefix, and parameters from a DHCPv6 server.

Default

None.

Usage Guidelines

Use this command to set the IP address and network prefix for an Ethernet interface.

If set to **dhcp**, the MTU value for the interface will be set via DHCP unless it is explicitly defined using interfaces ethernet <ethx> mtu <mtu> command which takes precedence. On lease release, it will set the interface MTU to 1500 if it is not explicitly defined.

Use the **set** form of this command to set the IP address and network prefix. You can set more than one IP address for the interface by creating multiple **address** configuration nodes.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces ethernet <ethx> bond-group <bondx>

Adds an Ethernet interface to a bonding group.

Syntax

set interfaces ethernet *ethx* **bond-group** *bondx*

delete interfaces ethernet *ethx* **bond-group** *bondx*

show interfaces ethernet *ethx* **bond-group**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        bond-group bondx
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.

Default

None.

Usage Guidelines

Use this command to add an Ethernet interface to an Ethernet link bond group.

An Ethernet interface can only be a member of one Ethernet link bond group and the bond group must first be defined using interfaces bonding <bondx>. The maximum number of Ethernet interfaces that can be added to a bonding group depends on available system resources. For most implementations this is essentially unlimited.

NOTE *The Ethernet interface will not be added to the bond group if it is disabled.*

You must not configure any IP address for the Ethernet interface if it is to become part of a bonding group. Instead, the IP address for the group is configured on the bonding interface using interfaces ethernet <ethx> address.

Use the **set** form of this command to add an Ethernet interface to an Ethernet link bond group.

Use the **delete** form of this command to remove an Ethernet interface from an Ethernet link bond group.

Use the **show** form of this command to view bond group configuration.

interfaces ethernet <ethx> description <descr>

Specifies a description for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **description** *descr*

delete interfaces ethernet *ethx* **description**

show interfaces ethernet *ethx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        description descr
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
descr	A mnemonic name or description for the Ethernet interface.

Default

None.

Usage Guidelines

Use this command to set a description for an Ethernet interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

interfaces ethernet <ethx> dhcpv6-options

Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.

Syntax

set interfaces ethernet *ethx* **dhcpv6-options** [**parameters-only** | **temporary**]

delete interfaces ethernet *ethx* **dhcpv6-options** [**parameters-only** | **temporary**]

show interfaces ethernet *ethx* **dhcpv6-options**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        dhcpv6-options [parameters-only|temporary]
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
parameters-only	Acquires only configuration parameters (and not an IPv6 address) from the DHCPv6 server. Only one of the **parameters-only** and the **temporary** parameter may be specified.
temporary	Acquires a temporary IPv6 address as described for IPv6 privacy addressing in RFC 4941. Only one of the **parameters-only** and the **temporary** parameter may be specified.

Default

None.

Usage Guidelines

Use this command to specify in what way the DHCPv6 client is to acquire an IPv6 address and/or parameters from a DHCPv6 server.

Note that these parameters are only relevant if the **dhcpv6** option has been set for the interfaces ethernet <ethx> address command.

The **parameters-only** option is typically used in conjunction with SLAAC or static address configuration. It and the **temporary** parameter are mutually exclusive.

Use the **set** form of this command to specify the DHCPv6 options.

Use the **delete** form of this command to remove the DHCPv6 options.

Use the **show** form of this command to view DHCPv6 option configuration.

interfaces ethernet <ethx> disable

Disables an Ethernet interface without discarding configuration.

Syntax

set interfaces ethernet *ethx* **disable**

delete interfaces ethernet *ethx* **disable**

show interfaces ethernet *ethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet ethx {
          disable
   }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to disable an Ethernet Interface without discarding configuration.

Use the **set** form of this command to disable the interface.

Use the **delete** form of this command to enable the interface.

Use the **show** form of this command to view Ethernet interface configuration.

interfaces ethernet <ethx> disable-flow-control

Disables IEEE 802.3x full-duplex flow control (“PAUSE” frames).

Syntax

set interfaces ethernet *ethx* **disable-flow-control**

delete interfaces ethernet *ethx* **disable-flow-control**

show interfaces ethernet *ethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
            disable-flow-control
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

If this option is not configured, PAUSE frame flow control is in effect.

Usage Guidelines

Use this command to disable “PAUSE” frames on an Ethernet interface.

The full-duplex Ethernet mode of operation includes an optional method of flow control known as PAUSE frames. If one end of the connectio becomes congested, the peer can send a PAUSE to the far-end device. After receiving the PAUSE frame, the far-end devices stops transmitting all traffic except media access CONTROL frames, for a standard interval. When the interval elapses, normal transmission resumes.

Use the **set** form of this command to disable PAUSE frame flow control.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view Ethernet interface configuration.

interfaces ethernet <ethx> disable-link-detect

Directs an Ethernet interface not to detect physical link-state changes.

Syntax

set interfaces ethernet *ethx* **disable-link-detect**

delete interfaces ethernet *ethx* **disable-link-detect**

show interfaces ethernet *ethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        disable-link-detect
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

The interface detects physical link state changes.

Usage Guidelines

Use this command to direct an Ethernet interface to not detect physical state change to the Ethernet link (for example, when the cable is unplugged).

Use the **set** form of this command to disable detection of physical state changes.

Use the **delete** form of this command to enable detection of physical state changes.

Use the **show** form of this command to view Ethernet interface configuration.

interfaces ethernet <ethx> duplex <duplexity>

Sets the duplex mode for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **duplex** *duplexity*

delete interfaces ethernet *ethx* **duplex**

show interfaces ethernet *ethx* **duplex**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        duplex duplexity
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
duplexity	The duplexity of the interface. Supported values are as follows: **auto**: The router automatically negotiates the duplexity with the interface at the other end of the link. **half**: Half duplex. **full**: Full duplex.

Default

The router autonegotiates duplexity.

Usage Guidelines

Use this command to set the duplexity characteristics of an Ethernet interface.

NOTE *Not all hardware supports having the duplex value explicitly set. If this is the case with the hardware you are using, an error will be displayed on commit.*

Use the **set** form of this command to set the duplexity of the interface.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view duplexity configuration.

interfaces ethernet <ethx> hw-id <mac-addr>

Associates the Ethernet interface name with a hardware MAC address.

Syntax

set interfaces ethernet *ethx* **hw-id** *mac-addr*

delete interfaces ethernet *ethx* **hw-id**

show interfaces ethernet *ethx* **hw-id**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        hw-id mac-addr
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
mac-addr	The MAC address burned into an Ethernet NIC. The format is 6 colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

The factory-assigned MAC address of the network interface card with which this Ethernet interface is associated.

Usage Guidelines

Use this command to associate the Ethernet interface (e.g. eth0) with a particular Ethernet NIC. When the system starts up, if no **hw-id** is specified for a particular interface the system will set it. If a **hw-id** is specified then the Ethernet interface is associated with that NIC.

NOTE *If you specify an* ***hw-id*** *it must be a valid MAC address on a NIC within your system.*

This is particularly useful if a new NIC is added to the system or you want to assign a specific interface name (e.g. eth0) to a specific NIC.

Use the **set** form of this command to associate the hardware ID with the interface.

Use the **delete** form of this command to remove the hardware ID configuration. The next time the system is started, a unassigned hardware ID will be assigned to the interface.

Use the **show** form of this command to view hardware ID configuration.

interfaces ethernet <ethx> ip enable-proxy-arp

Enables proxy ARP on an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **ip enable-proxy-arp**

delete interfaces ethernet *ethx* **ip enable-proxy-arp**

show interfaces ethernet *ethx* **ip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        ip {
            enable-proxy-arp
        }
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

Proxy ARP is not enabled on the Ethernet interface.

Usage Guidelines

Use this command to enable proxy Address Resolution Protocol (ARP) on an Ethernet interface.

Proxy ARP allows an Ethernet interface to respond with its own media access control (MAC) address to ARP requests for destination IP addresses on subnets attached to other interfaces on the system. Subsequent packets sent to those destination IP addresses are forwarded appropriately by the system.

Use the **set** form of this command to enable proxy ARP on the interface.

Use the **delete** form of this command to return the system to its default behavior.

Use the **show** form of this command to view the configuration.

interfaces ethernet <ethx> ipv6 address

Assigns an IPv6 address to an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

delete interfaces ethernet *ethx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

show interfaces ethernet *ethx* **ipv6 address** [**autoconf** | **eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        ipv6 {
            address {
                autoconf
                eui64 ipv6prefix
            }
        }
    }
}
```

Parameters

ethx	The interface identifier. Supported values are **eth0** through **eth23**.
autoconf	Generates an IPv6 address using the SLAAC protocol. Set this value if the interface is performing a “host” function rather than a “router” function. This value can be specified in addition to specifying static IPv6, static IPv4, or IPv4 DHCP addresses on the interface.
ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address, in EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value derived from the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to assign an IPv6 address to an Ethernet interface.

You can using the **autoconf** keyword to direct the system to autoconfigure the address, using the SLAAC (Stateless Address Auto-Configuration) protocol defined in RFC 4862. Alternatively, you can provide an EUI-64 IPv6 address prefix so that the system constructs the IPv6 address.

If you want the system to use SLAAC to acquire addresses on this interface, then in addition setting this parameter, you must also disable IPv6 forwarding, either globally (using the system ipv6 disable-forwarding command) or specifically on this interface (using the interfaces ethernet <ethx> ipv6 disable-forwarding command).

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view IPv6 address configuration settings.

interfaces ethernet <ethx> ipv6 disable-forwarding

Disables IPv6 forwarding on an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **ipv6 disable-forwarding**

delete interfaces ethernet *ethx* **ipv6 disable-forwarding**

show interfaces ethernet *ethx* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        ipv6 {
            disable-forwarding
        }
    }
}
```

Parameters

ethx	The interface identifier. Supported values are **eth0** through **eth23**.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on an Ethernet interface.

You can also disable IPv6 forwarding globally (that is, for all interfaces) using the system ipv6 disable-forwarding command.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

Use the **show** form of this command to display IPv6 packet forwarding interface configuration.

interfaces ethernet <ethx> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces ethernet *ethx* **ipv6 dup-addr-detect-transmits** *num*

delete interfaces ethernet *ethx* **ipv6 dup-addr-detect-transmits**

show interfaces ethernet *ethx* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        ipv6 {
            dup-addr-detect-transmits num
        }
    }
}
```

Parameters

ethx	The interface identifier. Supported values are **eth0** through **eth23**.
num	The number of times to transmit NS packets as part of the DAD process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view NS packet configuration for DAD.

interfaces ethernet <ethx> ipv6 router-advert

Specifies the router advertisements to be sent from an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **ipv6 router-advert** [**cur-hop-limit** *limit*] [**default-lifetime** *lifetime*] [**default-preference** *preference*] [**link-mtu** *mtu*] [**managed-flag** *state*] [**max-interval** *interval*] [**min-interval** *interval*] [**other-config-flag** *state*] [**prefix** *ipv6net* [**autonomous-flag** *state* | **on-link-flag** *state* | **preferred-lifetime** *lifetime* | **valid-lifetime** *lifetime*]] [**reachable-time** *time*] [**retrans-timer** *time*] [**send-advert** *state*]

delete interfaces ethernet *ethx* **ipv6 router-advert** [**cur-hop-limit**] [**default-lifetime**] [**default-preference**] [**link-mtu**] [**managed-flag**] [**max-interval**] [**min-interval**] [**other-config-flag**] [**prefix** *ipv6net* [**autonomous-flag** | **on-link-flag** | **preferred-lifetime** | **valid-lifetime**]] [**reachable-time**] [**retrans-timer** [**send-advert**]

show interfaces ethernet *ethx* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        ipv6 {
            router-advert {
                cur-hop-limit limit
                default-lifetime lifetime
                default-preference preference
                link-mtu mtu
                managed-flag state
                max-interval interval
                min-interval interval
                other-config-flag state
                prefix ipv6net {
                    autonomous-flag state
                    on-link-flag state
                    preferred-lifetime lifetime
                    valid-lifetime lifetime
                }
                reachable-time time
                retrans-timer time
                send-advert state
```

```
            }
        }
    }
}
```

Parameters

ethx	The interface identifier. Supported values are **eth0** through **eth23**.
cur-hop-limit *limit*	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets. This value is placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range is 0 to 255. The default is 64. A value of 0 means unspecified by the router.
default-lifetime lifetime	Specifies the lifetime , in seconds, associated with the default router. Supported values are 0, which indicates that the router is not a default router, and the range from the value configured for the **max-interval** option to 9000 (18.2 hours). If not configured, the value for this timer is three times **max-interval**.
default-preference *preference*	The preference associated with the default router. Supported values are as follows: **low**: The default router is low preference. **medium**: The default router is medium preference. **high**: The default router is high preference. The default is **medium**.
link-mtu *mtu*	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the type of link, as defined in RFC 2464. The default is 0, which means the MTU is not specified in the router advertisement message. That is because it is expected that the MTU will configured directly on the interface itself and not for routing advertisements. You can configure this option in cases where the link MTU is not well known. If the value set here does not match the MTU configured on the interface, the system issues a warning but does not fail.

managed-flag *state*	Whether to use the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration. The default is **false**.
max-interval *interval*	The maximum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default is 600 (10 minutes).
min-interval *interval*	The minimum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * **max-interval**. The default is 0.33 * **max-interval**.
other-config-flag *state*	The interfaces uses the administered (stateful) protocol for autoconfiguration of non-address information, as defined in RFC 4862. Supported values are as follows: **true**: Hosts use the administered protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default is **false**.
prefix *ipv6net*	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface, in the format *ipv6-address/prefix*. You can define more than one IPv6 prefix by configuring multiple **prefix** configuration nodes.
autonomous-flag *state*	Specifies whether the prefix can be used for autonomous address configuration as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default is **true**.

on-link-flag *state*	Specifies whether the prefix can be used for on-link determination, as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default is **true**.
preferred-lifetime *lifetime*	The length of time, in seconds, that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) is to remain preferred, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 604800 (seven days).
valid-lifetime *lifetime*	The length of time, in seconds, that the prefix is valid for the purpose of on-link determination, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 2592000 (30 days).
reachable-time *time*	The length of time, in milliseconds, for which the system assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range is 0 to 3600000, where a value of 0 means the reachable time is not specified in the router advertisement message. The default is 0.
retrans-timer *time*	The length of time, in milliseconds, between retransmitted NS messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295, where a value of 0 means the retransmit tme is not specified in the router advertisement message. The default is 0.

send-advert *state*	Specifies whether router advertisements are to be sent from this interface. Supported values are as follows: **true**: Sends router advertisements from this interface. **false**: Does not send router advertisements from this interface. If this value is in effect, parameters in this configuration subtree are still used to configure the local implementation parameters. The default is **true**.

Default

Router advertisements are not sent on an interface.

Usage Guidelines

Use this command to configure router advertisements (RAs) to be sent out of the interface being configured.

Router advertisements are sent out by IPv6 routers in order to advertise their existence to hosts on the network. IPv6 hosts do not send out router advertisements.

If the **router-advert** node of the configuration tree is missing, router advertisements are not sent out. Also, if IPv6 forwarding is disabled either globally (using the **system ipv6 disable-forwarding** command) or on the interface (using the interfaces ethernet <ethx> ipv6 disable-forwarding command), router advertisements are not sent out.

Most router advertisement parameters are required by either the Neighbor Discovery (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view router advertisement configuration.

interfaces ethernet <ethx> mac <mac-addr>

Sets the MAC address of an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **mac** *mac-addr*

delete interfaces ethernet *ethx* **mac**

show interfaces ethernet *ethx* **mac**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        mac mac-addr
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
mac-addr	The MAC address to be set for the Ethernet interface. The format is 6 colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

The default MAC address for an interface is the factory-set MAC address (i.e. the **hw-id**).

Usage Guidelines

Use this command to set the media access control (MAC) address of the interface. This value will override the **hw-id** which is the factory-set MAC address of the NIC.

Some Ethernet interfaces provide the ability to change their MAC address. This command allows you to change the MAC address of these interfaces.

Use the **set** form of this command to set the MAC address of the interface.

Use the **delete** form of this command to remove a configured MAC address for the interface, restoring the factory-assigned MAC address.

Use the **show** form of this command to view MAC address configuration.

interfaces ethernet <ethx> mirror <interface>

Mirrors inbound traffic from an Ethernet interface to another interface.

Syntax

set interfaces ethernet *ethx* **mirror** *interface*

delete interfaces ethernet *ethx* **mirror** *interface*

show interfaces ethernet *ethx* **mirror**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        mirror interface
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
interface	The identifier of the interface to which you are mirroring data; for example, **eth2**.

Default

None.

Usage Guidelines

Use this command to mirror the inbound traffic from one Ethernet interface to another interface.

This feature is typically used to provide a copy of traffic inbound on one interface to a system running a monitoring or IPS application on another interface. The benefit of mirroring the traffic is that the application is isolated from the source traffic and so application processing does not affect the traffic or the system performance.

Use the **set** form of this command to specify the mirror interface.

Use the **delete** form of this command to remove the mirror configuration.

Use the **show** form of this command to view the mirror configuration.

interfaces ethernet <ethx> mtu <mtu>

Sets the MTU for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **mtu** *mtu*

delete interfaces ethernet *ethx* **mtu**

show interfaces ethernet *ethx* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        mtu mtu
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
mtu	Sets the MTU, in octets, for the interface as a whole, including any logical interfaces configured for it. The range is 68 to 9000.

Default

If this value is not set, the default Ethernet MTU of 1500 is used.

Usage Guidelines

Use this command to set the maximum transmission unit (MTU) for an Ethernet interface. The value set for the physical Ethernet interface is inherited by all vifs and other sub-interfaces defined for the interface.

During forwarding, IPv4 packets larger than the MTU are fragmented unless the "Don't Fragment" (DF) bit is set in the IP header. In that case, the packets are dropped and an ICMP "fragmentation needed" message is returned to the sender.

Note that MTU sizes larger than 1500 cause the system to generate “jumbo frames,” which are not compatible with some Ethernet interface cards and devices.

If the Ethernet interface is part of an Ethernet link bond, MTU settings for the bonded link override MTU settings for the physical interface.

Use the **set** form of this command to specify the MTU.

Use the **delete** form of this command to remove MTU value and restore the default behavior.

Use the **show** form of this command to view MTU configuration.

interfaces ethernet <ethx> redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces ethernet *ethx* **redirect** *interface*

delete interfaces ethernet *ethx* **redirect** *interface*

show interfaces ethernet *ethx* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        redirect interface
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

interfaces ethernet <ethx> smp_affinity

Sets the symmetric multiprocessor affinity for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **smp_affinity** {**auto** | *int_mask*[,*rps_mask*]}

delete interfaces ethernet *ethx* **smp_affinity** {**auto** | *int_mask*[,*rps_mask*]}

show interfaces ethernet *ethx* **smp_affinity**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        smp_affinity [auto|int_mask[,rps_mask]]
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
auto	Automatically configures optimal SMP affinity.
int_mask	An interrupt mask. Up to sixteen hex digits (64 bits) that identify the processor(s) that this interface will interrupt; for example, “1” (that is , binary 0001) represents CPU 0, and “80” (that is, binary 1000000) represents CPU 7. You can distribute the interrupts from an interface among multiple processors by setting the bits for each CPU you wish to interrupt in the bitmask; for example, “3” (that is, binary 0011) represents CPUs 0 and 1.

rps_mask	A receive packet steering mask. Up to sixteen hex digits (64 bits) that identify the processor(s) that this interface will steer received traffic to; for example, "1" (that is , binary 0001) represents CPU 0, and "80" (that is, binary 1000000) represents CPU 7. You can distribute the received packets from an interface among multiple processors by setting the bits for each CPU you wish to distribute packes to in the bitmask; for example, "3" (that is, binary 0011) represents CPUs 0 and 1.

Default

SMP affinity is optimally configured automatically.

Usage Guidelines

Use this command to configure and display SMP affinity for an Ethernet interface.

Whenever a piece of hardware, such as disk controller or ethernet card, needs processing resources, it generates an interrupt request (IRQ). The IRQ tells the processor that resources are required and the processor should attend to the task.

In a multi-core computer using symmetric multiprocessing (SMP), any processor could be recruited to process any task. By default, the Vyatta system will automatically determine, based on the hardware used, the optimal SMP affinity settings. In general, this default setting should be used. In special circumstances where full control over the affinity settings is required, setting the SMP affinity masks for an interface allows you to control how the system responds to hardware interrupts by assigning interrupts from a given Ethernet interface to a specific processor and, optionally, controlling which processors the incoming data will be processed by.

For Ethernet interfaces that have multiple queues, and hence, multiple interrupts, setting an absolute *int_mask* is not supported. A setting of **auto** should be used for these interfaces.

Use the **set** form of this command to specify the SMP affinity for an Ethernet interface.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view SMP affinity configuration.

interfaces ethernet <ethx> speed <speed>

Sets the speed of an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **speed** *speed*

delete interfaces ethernet *ethx* **speed**

show interfaces ethernet *ethx* **speed**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        speed speed
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
speed	Sets the speed of the interface. Supported values are as follows: **auto**: The system autonegotiates the speed of the interface with the interface at the other end of the connection. **10**: 10 Mbps **100**: 100 Mbps **1000**: 1000 Mbps

Default

Ethernet link speed is autonegotiated.

Usage Guidelines

Use this command to set the link speed for an Ethernet interface.

NOTE *Not all hardware supports having the speed value explicitly set. If this is the case with the hardware you are using, an error is reported on commit.*

Use the **set** form of this command to set the speed.

Use the **delete** form of this command to restore the default behavior.

Use the **show** form of this command to view Ethernet speed configuration.

monitor interfaces ethernet <ethx> traffic

Displays (captures) traffic on an Ethernet interface.

Syntax

monitor interfaces ethernet *ethx* **traffic** [**not port** *port* | **port** *port*]

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
not port *port*	Shows captured traffic on all but this port.
port *port*	Shows captured traffic on this port only.

Default

Captured traffic for all ports on the specified interface is shown.

Usage Guidelines

Use this command to view Ethernet traffic on an Ethernet interface. Type <Ctrl>+c to stop the output.

Examples

Example 2-4 shows captured data on interface eth0.

Example 2-4 Displaying traffic data

```
vyatta@vyatta:~$ monitor interfaces ethernet eth0 traffic
Capturing traffic on eth0 ...
  0.000000 fe80::ad08:8661:4d:b925 -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  0.000067 fe80::69ca:5c11:bcf6:29da -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  2.608804 fe80::8941:71ef:b55d:e348 -> ff02::1:2    DHCPv6 Solicit
  3.010862 fe80::ad08:8661:4d:b925 -> ff02::c      SSDP M-SEARCH * HTTP/1.1
```

```
  3.010901 fe80::69ca:5c11:bcf6:29da -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  4.568357 192.168.1.254 -> 238.255.255.251 SSDP NOTIFY * HTTP/1.1
  4.568372 192.168.1.254 -> 238.255.255.251 SSDP NOTIFY * HTTP/1.1
...
```

show interfaces ethernet

Displays information and statistics about Ethernet interfaces.

Syntax

show interfaces ethernet [*ethx*]

Command Mode

Operational mode.

Parameters

ethx	Displays information for the specified Ethernet interface.

Default

Information is displayed for all Ethernet interfaces.

Usage Guidelines

Use this command to view operational status of Ethernet interfaces.

Examples

Example 2-5 shows information for all Ethernet interfaces.

Example 2-5 Displaying information for all Ethernet interfaces

```
vyatta@vyatta:~$ show interfaces ethernet
Interface    IP Address         State       Link   Description
eth0         -                  admin down  down
eth1         -                  up          up
eth2         10.1.0.66/24       up          up
eth3         -                  up          down
```

Example 2-6 shows information for interface eth2.

Example 2-6 Displaying information for one Ethernet interface

```
vyatta@vyatta:~$ show interfaces ethernet eth2
eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen
1000
    link/ether 00:13:46:e7:f8:87 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.66/24 brd 10.1.0.255 scope global eth2
    inet6 fe80::211:46ff:fee7:f687/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
        533348       3572          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
         54412        541          0          0          0          0
```

show interfaces ethernet detail

Displays detailed information about Ethernet interfaces.

Syntax

show interfaces ethernet detail

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to view detailed statistics and configuration information about Ethernet interfaces.

Examples

Example 2-7 shows the first screen of output for **show interfaces ethernet detail**.

Example 2-7 Displaying detailed Ethernet interface information

```
vyatta@vyatta:~$ show interfaces ethernet detail
eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:40:63:e2:e4:00 brd ff:ff:ff:ff:ff:ff

    RX:  bytes    packets     errors    dropped    overrun      mcast
            0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
            0          0          0          0          0          0

eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:40:63:e2:e3:dd brd ff:ff:ff:ff:ff:ff
    inet6 fe80::240:63ff:fee2:e3dd/64 scope link
       valid_lft forever preferred_lft forever
```

```
    RX:  bytes    packets     errors    dropped    overrun      mcast
               0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
             468          6          0          0          0          0

eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:13:46:e7:f8:87 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.66/24 brd 10.1.0.255 scope global eth2
    inet6 fe80::211:46ff:fee7:f687/64 scope link
       valid_lft forever preferred_lft forever
lines 1-23
```

show interfaces ethernet <ethx> brief

Displays a brief status for an Ethernet interface.

Syntax

show interfaces ethernet *ethx* **brief**

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to view the status of an Ethernet interface.

Examples

Example 2-8 shows brief status for interface eth2.

Example 2-8 Displaying brief Ethernet interface status

```
vyatta@vyatta:~$ show interfaces ethernet eth2 brief
Interface    IP Address         State       Link   Description
eth2         10.1.0.66/24       up          up
```

show interfaces ethernet <ethx> identify

Blinks the LEDs on an Ethernet interface in order to identify it.

Syntax

show interfaces ethernet *ethx* **identify**

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to help you identify a physical Ethernet port in order to map it to the *ethx* identifier within the Vyatta system.

Examples

Example 2-9 shows the output for **show interfaces ethernet** *ethx* **identify**.

Example 2-9 Identifying an Ethernet interface by blinking its LED

```
vyatta@vyatta:~$ show interfaces ethernet eth2 identify
Interface eth2 should be blinking now.
Press Enter to stop...
```

show interfaces ethernet <ethx> physical

Displays physical layer information for Ethernet interfaces.

Syntax

show interfaces ethernet *ethx* **physical**

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to view physical layer information of Ethernet interfaces.

Examples

Example 2-10 shows output for **show interfaces ethernet** *ethx* **physical**.

Example 2-10 Displaying physical line characteristics for an Ethernet interface.

```
vyatta@vyatta:~$ show interfaces ethernet eth0 physical
Settings for eth0:
    Current message level: 0x00000007 (7)
    Link detected: yes
driver: pcnet32
version: 1.35
firmware-version:
bus-info: 0000:02:00.0
vyatta@vyatta:~$
```

show interfaces ethernet <ethx> queue

Displays Ethernet queuing information.

Syntax

show interfaces ethernet *ethx* **queue** [**class** | **filter**]

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
class	Display queue classes for the specified interface.
filter	Display queue filters for the specified interface.

Default

None.

Usage Guidelines

Use this command to view Ethernet queue information.

Examples

Example 2-11 shows queue information for interface eth0.

Example 2-11 Displaying Ethernet queue information

```
vyatta@vyatta:~$ show interfaces ethernet eth0 queue
qdisc pfifo_fast 0: root bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 810323 bytes 6016 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
```

show interfaces ethernet <ethx> statistics

Displays Ethernet statistics.

Syntax

show interfaces ethernet *ethx* **statistics**

Command Mode

Operational mode.

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to view Ethernet statistics information on an Ethernet interface.

Examples

Example 2-12 shows Ethernet statistics information for interface eth3.

Example 2-12 Displaying Ethernet statistics information

```
vyatta@vyatta:~$ show interfaces ethernet eth3 statistics
NIC statistics:
     tx_ok: 1111
     rx_ok: 1467
     tx_err: 0
     rx_err: 4
     rx_fifo: 0
     frame_align: 0
     tx_ok_1col: 0
     tx_ok_mcol: 0
     rx_ok_phys: 1376
     rx_ok_bcast: 1
     rx_ok_mcast: 0
```

```
        tx_abort: 0
        tx_underrun: 0
        rx_frags: 0
    vyatta@vyatta:~$
```

Chapter 3: Ethernet Link Bonding Interfaces

This chapter explains how to bond Ethernet links into a larger virtual link.

This chapter presents the following topics:

Ethernet Link Bonding Overview

In some operational scenarios, it makes sense to group together multiple physical links to create a larger virtual link. This offers a ways to increase performance between two devices without having to pay for a higher-speed physical link, and to provide redundancy so that there is still connectivity in the event that a link fails. In the wide area network, multilink Point-to-Point Protocol (MLPPP) is used to bundle multiple PPP links; In the local area network, Ethernet link bonding is used to bundle multiple Ethernet links.

Many implementations of Ethernet link bonding have been non-standard. The IEEE 802.3ad specification (now called IEEE 802.1ax) was defined to attempt to increase standardization in the market. The IEEE 802.3ad standard has been adopted to varying degrees by all manufacturers. This standard specifies the general properties of the link, as well as the defining the Link Aggregation Control Protocol (LACP).

The 802.3ad LACP is an active protocol that runs on Ethernet links configured for bonding. LACP allows peers to negotiate the automatic bonding of multiple links and helps detect situations where one side is not configured correctly for link bonding. The LACP also actively tests each of the physical connections between each device so that link failures can be detected even if there are other physical devices attached to either end (e.g. physical media converters) which would otherwise not show link-down if a fault occurs in the middle of the physical link. If a link fails, traffic is simply redistributed dynamically to the remaining links.

The standard assumes that all physical links comprising the bonded virtual link are full-duplex and point-to-point. Violation of either of these assumptions can cause unexpected behavior in the bonded link.

The 802.3ad standard specifies that all packets belonging to a “conversation” must travel across the same physical link and that no packets may be duplicated. However, both the abstraction of “conversation” and the algorithm for assigning conversations to each link are incompletely specified; as a result, specific implementations vary, even between the ends of the bonded virtual link. This can lead to asymmetric traffic flow.

The number of links that can be bonded is limited by your system capacity, especially memory. The Ethernet links in a bonded link need not be all the same speed.

Physical links that are added to a bonded link need not be operational when they are added. Of the configuration for the bonded link, only maximum transmission unit (MTU) is inherited from the bundle. That is, if you change the MTU of the bonded link, the MTU of the underlying Ethernet links is overridden. The remaining configuration is always taken from the configuration specified for the individual Ethernet link. The exception is that a physical link cannot be assigned an IP address if it is to be added to a bond group.

You can include VLANs within a bonded link; however, bundling multiple VLANs together as a bonded trunk is not recommended. Since the purpose of bonding is to improve availability and performance, the bonded link requires actual physical links as a base.

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on Ethernet link bonding interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*

Ethernet Bonding Configuration Examples

This section presents the following topics:

Basic Ethernet Bonding

To configure an Ethernet bonded link, you create a "bonding interface" and configure it as any other Ethernet interface. Then, for each Ethernet interface that is to belong to the bonded link, specify the bond group—that is, point to the bonding interface you created.

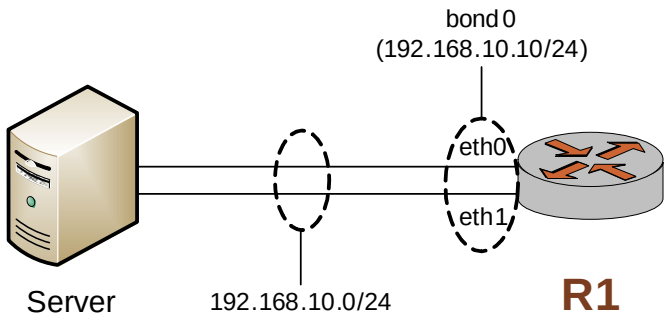
Figure 3-1 shows a simple Ethernet link bonding scenario, with an Ethernet bonded link consisting of two physical Ethernet links. In this example:

- The bond group bond0 is created using the default bonding mode (802.3ad).
- Interfaces eth0 and eth1 are the physical links. They are both added as member links to the bonded interface bond0.

Note that no IP addresses are assigned to the individual physical Ethernet links. The bonding does not work if any of the component Ethernet links has an IP address assigned to it.

Use the **show interfaces** and **show interfaces bonding** commands to determine the status of the bonding interface and its constituent Ethernet interfaces.

Figure 3-1 Creating a bond group with two Ethernet interfaces



To configure this scenario, perform the following steps in configuration mode.

Example 3-1 Creating a bond group with two Ethernet interfaces

Step	Command
Create the bond0 bonding group.	`vyatta@R1# set interfaces bonding bond0`
Set the IP address for the bonding group.	`vyatta@R1# set interfaces bonding bond0 address 192.168.10.10/24`
Set the bonding mode of the bonding group.	`vyatta@R1# set interfaces bonding bond0 mode 802.3ad`
Add eth0 to the bond0 bonding group.	`vyatta@R1# set interfaces ethernet eth0 bond-group bond0`
Add eth1 to the bond0 bonding group.	`vyatta@R1# set interfaces ethernet eth1 bond-group bond0`
Commit the change.	`vyatta@R1# commit`

Show the bonding group configuration.	`vyatta@R1# show interfaces bonding bond0`
	` address 192.168.10.10/24`
	` mode 802.3ad`

Show the eth0 configuration.	`vyatta@R1# show interfaces ethernet eth0` ` bond-group bond0`
Show the eth1 configuration.	`vyatta@R1# show interfaces ethernet eth1` ` bond-group bond0`

Ethernet Bonding with VLAN

Once a bonding interface has been created it is possible to create a VLAN within it. The following example extends the previous example by adding a VLAN. The resulting bonding interface contains both VLAN and non-VLAN traffic.

To configure this scenario, perform the following steps in configuration mode.

Example 3-2 Adding a VLAN to an existing bonding interface.

Step	Command
Add the vif configuration to the bonding group.	`vyatta@R1# set interfaces bonding bond0 vif 192 address 10.192.248.225/24`
Commit the change.	`vyatta@R1# commit`

Show the new bonding group configuration.	`vyatta@R1# show interfaces bonding bond0`
	` address 192.168.10.10/24`
	` mode 802.3ad`
	` vif 192 {`
	`  address 10.192.248.225/24`
	` }`

IPv6 on Ethernet Link Bonding Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Ethernet Link Bonding Commands

Configuration Commands	
interfaces bonding <bondx>	Defines an Ethernet link bonding interface (bond group).
interfaces bonding <bondx> address	Assigns a network address to an Ethernet link bond group.
interfaces bonding <bondx> arp-monitor	Specifies parameters for link monitoring via ARP.
interfaces bonding <bondx> description <desc>	Specifies a description for an Ethernet link bond group.
interfaces bonding <bondx> dhcpv6-options	Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.
interfaces bonding <bondx> disable	Disables an Ethernet link bond group without discarding configuration.
interfaces bonding <bondx> disable-link-detect	Directs an Ethernet link bond group to not detect physical link-state changes.
interfaces bonding <bondx> hash-policy <policy>	Sets the transmit hash policy for an Ethernet link bond group.
interfaces bonding <bondx> ip enable-proxy-arp	Enables proxy ARP on an Ethernet link bonding interface.
interfaces bonding <bondx> ipv6 address	Assigns an IPv6 address to an Ethernet link bonding interface.
interfaces bonding <bondx> ipv6 disable-forwarding	Disables IPv6 forwarding on an Ethernet link bonding interface.
interfaces bonding <bondx> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces bonding <bondx> ipv6 router-advert	Specifies the router advertisements to be sent from an Ethernet link bonding interface.
interfaces bonding <bondx> mac <mac-addr>	Sets the MAC address of an Ethernet link bond group.
interfaces bonding <bondx> mode	Sets the bonding mode for an Ethernet link bond group.
interfaces bonding <bondx> mtu <mtu>	Specifies the MTU for an Ethernet link bond group.
interfaces bonding <bondx> primary <ethx>	Sets one of the Ethernet links within a bond group as the primary link.
interfaces bonding <bondx> redirect <interface>	Redirects inbound traffic from from one interface to another interface.

Operational Commands	
show interfaces bonding	Shows Ethernet link bond group information.

Commands for using other system features with bonded Ethernet link interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Bridging	Layer 2 bridging is supported on bonding interfaces. Commands for configuring bridge groups are described in the *Vyatta Bridging Reference Guide*.
Firewall	Firewall is supported on bonding interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.
OSPF	OSPF is supported on bonding interfaces. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on bonding interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
QoS	Quality of service traffic policies are supported on bonding interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on bonding interfaces. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on bonding interfaces. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.
VLAN interfaces	802.1Q VLAN operation is supported on bonding interfaces. Commands for configuring r VLAN interfaces (vifs) are described in "Chapter 6: VLAN Interfaces."
VRRP	VRRP is supported on bonding interfaces and on VLAN interfaces configured under bonding interfaces. Commands for configuring VRRP are described in the *Vyatta High Availability Reference Guide*.

interfaces bonding <bondx>

Defines an Ethernet link bonding interface (bond group).

Syntax

set interfaces bonding *bondx*

delete interfaces bonding *bondx*

show interfaces bonding *bondx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
    }
}
```

Parameters

bondx	Multi-node. The identifier of the bond group you are defining. Supported values are **bond0** through **bond99**. You can define more than one bond group by specifying multiple **bonding** configuration nodes.

Default

None.

Usage Guidelines

Use this command to define an Ethernet link bonding interface, also known as a bond group. An Ethernet link bond group allows the bandwidth of individual links to be combined into a single virtual link.

Note that you must create the bond group (using this command or one of its variants) before you can assign Ethernet interfaces to it.

Use the **set** form of this command to define settings on an Ethernet link bond group.

Use the **delete** form of this command to remove all configuration for an Ethernet link bond group.

Use the **show** form of this command to view Ethernet link bond group configuration.

interfaces bonding <bondx> address

Assigns a network address to an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **address** {*ipv4net* | *ipv6net* | **dhcp** | **dhcpv6**}

delete interfaces bonding *bondx* **address** {*ipv4net* | *ipv6net* | **dhcp** | **dhcpv6**}

show interfaces bonding *bondx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        address [ipv4net|ipv6net|dhcp|dhcpv6]
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
ipv4	Defines an IPv4 network address on this interface. The format is *ip-address/prefix* (for example, 192.168.1.77/24). You can define multiple IPv4 network addresses for a single interface, by creating multiple **address** configuration nodes.
ipv6net	Defines an IPv6 network address on this interface. The format is *ipv6-address/prefix* (for example, 2001:db8:1234::/48). You can define multiple IPv6 network addresses for a single interface, by creating multiple **address** configuration nodes.
dhcp	Defines the interface as a DHCP client, which obtains its address and prefix from a DHCP server.
dhcpv6	Defines the interface as a DHCPv6 (DHCP for IPv6) client, which obtains its address and prefix from a DHCPv6 server.

Default

None.

Usage Guidelines

Use this command to set the IP address and network prefix for an Ethernet link bond group.

You can direct the interface to obtain its address and prefix from a Dynamic Host Configuration Protocol (DHCP) server by using the **dhcp** option.

You can direct the interface to obtain its address and prefix from a Dynamic Host Configuration Protocol (DHCP) server for IPv6 by using the **dhcpv6** option.

Use the **set** form of this command to set the IP address and network prefix. You can set more than one IP address for the interface by creating multiple **address** configuration nodes.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces bonding <bondx> arp-monitor

Specifies parameters for link monitoring via ARP.

Syntax

set interfaces bonding *bondx* **arp-monitor** [**interval** *interval* | **target** *ipv4*]

delete interfaces bonding *bondx* **arp-monitor** [**interval** | **target** *ipv4*]

show interfaces bonding *bondx* **arp-monitor** [**interval** | **target**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        arp-monitor {
            interval interval
            target ipv4
        }
    }
}
```

Parameters

bondx	Multi-node. The identifier for the bond group. Supported values are **bond0** through **bond99**.
interval	Specifies the interval (in milliseconds) to check slave devices (defined by **target**) to confirm that they have sent or received traffic recently. A value of 0 disables ARP monitoring. The default is 0.
ipv4	Specifies a target IPv4 address to be used for ARP monitoring. ARP requests are sent periodically (defined by **interval**) to determine the health of the link to the target. You can define multiple target IPv4 addresses for a single interface, by creating multiple **target** configuration nodes. Up to 16 **target**s can be specified.

Default

None.

Usage Guidelines

Use this command to set parameters for link monitoring via ARP. ARP requests are sent to the specified **target**s at the specified **interval**. A link is considered down if a response from the ARP target is not received. ARP monitoring can be used in cases where physical link detection is insufficient to determine whether the link should be considered down. An example of this is when the link to the local switch is up but the link on the other side of the switch is down.

Use the **set** form of this command to set the ARP monitoring parameters

Use the **delete** form of this command to remove ARP monitoring configuration.

Use the **show** form of this command to view ARP monitoring configuration.

interfaces bonding <bondx> description <desc>

Specifies a description for an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **description** *desc*

delete interfaces bonding *bondx* **description**

show interfaces bonding *bondx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        description desc
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
desc	A brief description for the bond group.

Default

None.

Usage Guidelines

Use this command to specify a description for a bond group.

Use the **set** form of this command to specify a description for the bond group.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view the description.

interfaces bonding <bondx> dhcpv6-options

Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.

Syntax

set interfaces bonding *bondx* **dhcpv6-options** [**parameters-only** | **temporary**]

delete interfaces bonding *bondx* **dhcpv6-options** [**parameters-only** | **temporary**]

show interfaces bonding *bondx* **dhcpv6-options**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        dhcpv6-options [parameters-only|temporary]
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
parameters-only	Acquires only configuration parameters (and not an IPv6 address) from the DHCPv6 server. Only one of the **parameters-only** and the **temporary** parameter may be specified.
temporary	Acquires a temporary IPv6 address as described for IPv6 privacy addressing in RFC 4941. Only one of the **parameters-only** and the **temporary** parameter may be specified.

Default

None.

Usage Guidelines

Use this command to specify in what way the DHCPv6 client is to acquire an IPv6 address and/or parameters from a DHCPv6 server.

Note that these parameters are only relevant if the **dhcpv6** option has been set for the interfaces bonding <bondx> address command.

The **parameters-only** option is typically used in conjunction with SLAAC or static address configuration. It and the **temporary** parameter are mutually exclusive.

Use the **set** form of this command to specify the DHCPv6 options.

Use the **delete** form of this command to remove the DHCPv6 options.

Use the **show** form of this command to view DHCPv6 option configuration.

interfaces bonding <bondx> disable

Disables an Ethernet link bond group without discarding configuration.

Syntax

set interfaces bonding *bondx* **disable**

delete interfaces bonding *bondx* **disable**

show interfaces bonding *bondx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        disable
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.

Default

None.

Usage Guidelines

Use this command to disable an Ethernet link bond group without discarding configuration.

Use the **set** form of this command to disable the interface.

Use the **delete** form of this command to enable the interface.

Use the **show** form of this command to view the configuration.

interfaces bonding <bondx> disable-link-detect

Directs an Ethernet link bond group to not detect physical link-state changes.

Syntax

set interfaces bonding *bondx* **disable-link-detect**

delete interfaces bonding *bondx* **disable-link-detect**

show interfaces bonding *bondx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        disable-link-detect
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.

Default

The interface detects physical link state changes.

Usage Guidelines

Use this command to direct an Ethernet link bond group to not detect physical state changes to an underlying physical Ethernet link (for example, when the cable is unplugged).

Use the **set** form of this command to disable detection of physical state changes.

Use the **delete** form of this command to enable detection of physical state changes.

Use the **show** form of this command to view Ethernet link bond group configuration.

interfaces bonding <bondx> hash-policy <policy>

Sets the transmit hash policy for an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **hash-policy** {**layer2** | **layer2+3** | **layer3+4**}

delete interfaces bonding *bondx* **hash-policy**

show interfaces bonding *bondx* **hash-policy**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        hash-policy {
            layer2
            layer2+3
            layer3+4
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
layer2	Uses the source and destination MAC addresses to generate the hash (802.3ad). This is the default.
layer2+3	Combines the source and destination MAC and IP addresses to generate the hash. This policy will place all traffic to a given network peer on the same link. For non-IP traffic this policy works the same as **layer2**.

layer3+4	Combines the source and destination IP address and port to generate the hash. This policy will allow traffic to a specific peer to span multiple bonded links though a single connection will be assigned a specific link. For non-IP traffic this policy works the same as **layer2**. This policy is intended to mimic the behavior of certain switches but is not 802.3ad compliant as out of order packet delivery can occur.

Default

The MAC address is used to generate the hash (**layer2**).

Usage Guidelines

Use this command to set the bonding transmit hash policy for the Ethernet link bond group. This policy is used to determine which of the bonded links is to be used for given outgoing traffic.

Use the **set** form of this command to set the bonding transmit hash policyof the bond group.

Use the **delete** form of this command to restore the default bonding transmit hash policy for the bond group.

Use the **show** form of this command to view bonding transmit hash policy configuration.

interfaces bonding <bondx> ip enable-proxy-arp

Enables proxy ARP on an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **ip enable-proxy-arp**

delete interfaces bonding *bondx* **ip enable-proxy-arp**

show interfaces bonding *bondx* **ip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        ip {
            enable-proxy-arp
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.

Default

Proxy ARP is not enabled on the Ethernet link bonding interface.

Usage Guidelines

Use this command to enable proxy Address Resolution Protocol (ARP) on an Ethernet link bonding interface.

Proxy ARP allows an Ethernet link bonding interface to respond with its own media access control (MAC) address to ARP requests for destination IP addresses on subnets attached to other interfaces on the system. Subsequent packets sent to those destination IP addresses are forwarded appropriately by the system.

Use the **set** form of this command to enable proxy ARP on the interface.

Use the **delete** form of this command to return the system to its default behavior.

Use the **show** form of this command to view the configuration.

interfaces bonding <bondx> ipv6 address

Assigns an IPv6 address to an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

delete interfaces bonding *bondx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

show interfaces bonding *bondx* **ipv6 address** [**autoconf** | **eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        ipv6 {
            address {
                autoconf
                eui64 ipv6prefix
            }
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
autoconf	Generates an IPv6 address using the SLAAC protocol. Set this value if the interface is performing a “host” function rather than a “router” function. This value can be specified in addition to specifying static IPv6, static IPv4, or IPv4 DHCP addresses on the interface.
ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address, in EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value derived from the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to assign an IPv6 address to an Ethernet link bonding interface.

You can using the **autoconf** keyword to direct the system to autoconfigure the address, using the SLAAC (Stateless Address Auto-Configuration) protocol defined in RFC 4862. Alternatively, you can provide an EUI-64 IPv6 address prefix so that the system constructs the IPv6 address.

If you want the system to use SLAAC to acquire addresses on this interface, then in addition setting this parameter, you must also disable IPv6 forwarding, either globally (using the system ipv6 disable-forwarding command) or specifically on this interface (using the interfaces bonding <bondx> ipv6 disable-forwarding command).

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view IPv6 address configuration settings.

interfaces bonding <bondx> ipv6 disable-forwarding

Disables IPv6 forwarding on an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **ipv6 disable-forwarding**

delete interfaces bonding *bondx* **ipv6 disable-forwarding**

show interfaces bonding *bondx* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        ipv6 {
            disable-forwarding
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on an Ethernet link bonding interface.

You can also disable IPv6 forwarding globally (that is, for all interfaces) using the system ipv6 disable-forwarding command.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

Use the **show** form of this command to display IPv6 packet forwarding interface configuration.

interfaces bonding <bondx> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces bonding *bondx* **ipv6 dup-addr-detect-transmits** *num*

delete interfaces bonding *bondx* **ipv6 dup-addr-detect-transmits**

show interfaces bonding *bondx* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        ipv6 {
            dup-addr-detect-transmits num
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
num	The number of times to transmit NS packets as part of the DAD process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view NS packet configuration for DAD.

interfaces bonding <bondx> ipv6 router-advert

Specifies the router advertisements to be sent from an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **ipv6 router-advert** [**cur-hop-limit** *limit*] [**default-lifetime** *lifetime*] [**default-preference** *preference*] [**link-mtu** *mtu*] [**managed-flag** *state*] [**max-interval** *interval*] [**min-interval** *interval*] [**other-config-flag** *state*] [**prefix** *ipv6net* [**autonomous-flag** *state* | **on-link-flag** *state* | **preferred-lifetime** *lifetime* | **valid-lifetime** *lifetime*]] [**reachable-time** *time*] [**retrans-timer** *time*] [**send-advert** *state*]

delete interfaces bonding *bondx* **ipv6 router-advert** [**cur-hop-limit**] [**default-lifetime**] [**default-preference**] [**link-mtu**] [**managed-flag**] [**max-interval**] [**min-interval**] [**other-config-flag**] [**prefix** *ipv6net* [**autonomous-flag** | **on-link-flag** | **preferred-lifetime** | **valid-lifetime**]] [**reachable-time**] [**retrans-timer** [**send-advert**]

show interfaces bonding *bondx* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        ipv6 {
            router-advert {
                cur-hop-limit limit
                default-lifetime lifetime
                default-preference preference
                link-mtu mtu
                managed-flag state
                max-interval interval
                min-interval interval
                other-config-flag state
                prefix ipv6net {
                    autonomous-flag state
                    on-link-flag state
                    preferred-lifetime lifetime
                    valid-lifetime lifetime
                }
```

```
                reachable-time time
                retrans-timer time
                send-advert state
            }
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
cur-hop-limit *limit*	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets. This value is placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range is 0 to 255. The default is 64. A value of 0 means unspecified by the router.
default-lifetime lifetime	Specifies the lifetime , in seconds, associated with the default router. Supported values are 0, which indicates that the router is not a default router, and the range from the value configured for the **max-interval** option to 9000 (18.2 hours). If not configured, the value for this timer is three times **max-interval**.
default-preference *preference*	The preference associated with the default router. Supported values are as follows: **low**: The default router is low preference. **medium**: The default router is medium preference. **high**: The default router is high preference. The default is **medium**.
link-mtu *mtu*	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the type of link, as defined in RFC 2464. The default is 0, which means the MTU is not specified in the router advertisement message. That is because it is expected that the MTU will configured directly on the interface itself and not for routing advertisements. You can configure this option in cases where the link MTU is not well known. If the value set here does not match the MTU configured on the interface, the system issues a warning but does not fail.

managed-flag *state*	Whether to use the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration. The default is **false**.
max-interval *interval*	The maximum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default is 600 (10 minutes).
min-interval *interval*	The minimum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * **max-interval**. The default is 0.33 * **max-interval**.
other-config-flag *state*	The interfaces uses the administered (stateful) protocol for autoconfiguration of non-address information, as defined in RFC 4862. Supported values are as follows: **true**: Hosts use the administered protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default is **false**.
prefix *ipv6net*	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface, in the format *ipv6-address/prefix*. You can define more than one IPv6 prefix by configuring multiple **prefix** configuration nodes.
autonomous-flag *state*	Specifies whether the prefix can be used for autonomous address configuration as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default is **true**.

on-link-flag *state*	Specifies whether the prefix can be used for on-link determination, as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default is **true**.
preferred-lifetime *lifetime*	The length of time, in seconds, that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) is to remain preferred, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 604800 (seven days).
valid-lifetime *lifetime*	The length of time, in seconds, that the prefix is valid for the purpose of on-link determination, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 2592000 (30 days).
reachable-time *time*	The length of time, in milliseconds, for which the system assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range is 0 to 3600000, where a value of 0 means the reachable time is not specified in the router advertisement message. The default is 0.
retrans-timer *time*	The length of time, in milliseconds, between retransmitted NS messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295, where a value of 0 means the retransmit tme is not specified in the router advertisement message. The default is 0.

send-advert *state*	Specifies whether router advertisements are to be sent from this interface. Supported values are as follows: **true**: Sends router advertisements from this interface. **false**: Does not send router advertisements from this interface. If this value is in effect, parameters in this configuration subtree are still used to configure the local implementation parameters. The default is **true**.

Default

Router advertisements are not sent on an interface.

Usage Guidelines

Use this command to configure router advertisements (RAs) to be sent out of the interface being configured.

Router advertisements are sent out by IPv6 routers in order to advertise their existence to hosts on the network. IPv6 hosts do not send out router advertisements.

If the **router-advert** node of the configuration tree is missing, router advertisements are not sent out. Also, if IPv6 forwarding is disabled either globally (using the **system ipv6 disable-forwarding** command) or on the interface (using the interfaces bonding <bondx> ipv6 disable-forwarding command), router advertisements are not sent out.

Most router advertisement parameters are required by either the Neighbor Discovery (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view router advertisement configuration.

interfaces bonding <bondx> mac <mac-addr>

Sets the MAC address of an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **mac** *mac-addr*

delete interfaces bonding *bondx* **mac**

show interfaces bonding *bondx* **mac**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        mac mac-addr
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
mac-addr	The MAC address for the Ethernet link bond group. The format should be appropriate for the interface type. For an Ethernet interface, this is six colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

The MAC address used is the MAC address of the first interface added to the bond group.

Usage Guidelines

Use this command to set the media access control (MAC) address of the bond group.

Use the **set** form of this command to set the MAC address of the bond group.

Use the **delete** form of this command to remove the configured MAC address for the bond group.

Use the **show** form of this command to view MAC address configuration for a bond group.

interfaces bonding <bondx> mode

Sets the bonding mode for an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **mode** {**802.3ad** | **active-backup** | **adaptive-load-balance** | **round-robin** | **transmit-load-balance** | **xor-hash** | **broadcast**}

delete interfaces bonding *bondx* **mode**

show interfaces bonding *bondx* **mode**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        mode {
            802.3ad
            active-backup
            adaptive-load-balance
            round-robin
            transmit-load-balance
            xor-hash
            broadcast
        }
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
802.3ad	Uses IEEE 802.3ad dynamic link aggregation as the bonding mode. This mode creates aggregation groups that share the same speed and duplexity settings. Transmission is based the default transmit hash policy or, if set, the policy defined in interfaces bonding <bondx> hash-policy <policy>.

active-backup	Sets an active-backup policy as the bonding mode. In this mode, only one Ethernet interface within the bonding interface is active (the **primary**). A different Ethernet interface becomes active if and only if the primary Ethernet interface fails. The bonding interface's MAC address is externally visible only on the active Ethernet interface.
adaptive-load-balance	Uses adaptive load balancing as the bonding mode. This mode includes both adaptive transmit load balancing plus receive load balancing for IPv4 traffic, and does not require any special switch support. The receive load balancing is achieved by ARP negotiation.
round-robin	Uses a round-robin policy as the bonding mode. In this mode, the system transmits packets in sequential order from the first available Ethernet interface within the bonding interface through the last. Round-robin load balancing helps manage network load and provides fault tolerance.
transmit-load-balance	Uses adaptive transmit load balancing as the bonding mode. This mode is a type of channel bonding that does not require any special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each Ethernet interface within the bonding interface. Incoming traffic is received by the current Ethernet interface. If the receiving Ethernet interface fails, another Ethernet interface takes over the MAC address of the failed receiving interface.
xor-hash	Uses an XOR policy as the bonding mode. In this mode, transmission is based the default transmit hash policy or, if set, the policy defined in interfaces bonding <bondx> hash-policy <policy>. This mode provides load balancing and fault tolerance.
broadcast	Uses a broadcast policy as the bonding mode. In this mode, the system transmits everything on all Ethernet interfaces. This mode provides fault tolerance but not load balancing.

Default

IEEE 802.3ad dynamic link aggregation is the bonding mode.

Usage Guidelines

Use this command to set the bonding mode for the Ethernet link bond group.

Use the **set** form of this command to set the bonding mode of the bond group.

Use the **delete** form of this command to restore the default bonding mode for the bond group.

Use the **show** form of this command to view bonding mode configuration.

interfaces bonding <bondx> mtu <mtu>

Specifies the MTU for an Ethernet link bond group.

Syntax

set interfaces bonding *bondx* **mtu** *mtu*

delete interfaces bonding *bondx* **mtu**

show interfaces bonding *bondx* **mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        mtu mtu
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
mtu	Sets the MTU, in octets, for the interface as a whole, including any logical interfaces configured for it. The range is 1 to 1500.

Default

The MTU of the first Ethernet link added to the group is used.

Usage Guidelines

Use this command to set the Maximum Transmission Unit (MTU) for an Ethernet link bond group. This value is also applied to any vifs defined for the bonding interface.

Note that changing the MTU changes the MTU on the Ethernet links within the bond. Also, explicitly changing the MTU of the Ethernet links within the bond (by configuring the individual links) is not allowed.

When forwarding, IPv4 packets larger than the MTU will be fragmented unless the DF bit is set. In that case, the packets will be dropped and an ICMP “Packet too big” message is returned to the sender.

Use the **set** form of this command to set the MTU of a bond group.

Use the **delete** form of this command to restore the default MTU and disable fragmentation.

Use the **show** form of this command to view MTU configuration for a bond group.

interfaces bonding <bondx> primary <ethx>

Sets one of the Ethernet links within a bond group as the primary link.

Syntax

set interfaces bonding *bondx* **primary** *ethx*

delete interfaces bonding *bondx* **primary**

show interfaces bonding *bondx* **primary**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        primary ethx
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
ethx	The identifier of the primary Ethernet interface within the bond group. Supported values are **eth0** through **eth23**.

Default

There is no primary link.

Usage Guidelines

Use this command to specify the primary Ethernet interface within the Ethernet link bonding interface.

This option is only available when the bonding mode is Active Backup.

When the bonding mode is Active Backup and an interface is identified as the primary, the primary interface is always the only active member of the bonding interface so long as it is available. Only when the primary is off-line are alternates used.

This option is useful when one member link is to be preferred over another; for example, when one member link has higher throughput than another.

Use the **set** form of this command to designate an Ethernet interface the primary interface for Active Backup Ethernet link bonding.

Use the **delete** form of this command to remove the primary Ethernet interface as the primary interface for Ethernet link bonding.

Use the **show** form of this command to view Ethernet link bonding configuration.

interfaces bonding <bondx> redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces bonding *bondx* **redirect** *interface*

delete interfaces bonding *bondx* **redirect** *interface*

show interfaces bonding *bondx* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        redirect interface
    }
}
```

Parameters

bondx	The identifier for the bond group. Supported values are **bond0** through **bond99**.
interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

show interfaces bonding

Shows Ethernet link bond group information.

Syntax

show interfaces bonding [**detail** | **slaves**]

Command Mode

Operational mode.

Parameters

detail	Displays detailed information for the bonding interface.
slaves	Displays information about the bonding slaves

Default

Information is displayed for all Ethernet link bond groups.

Usage Guidelines

Use this command to view operational status of configured Ethernet link bond groups.

Examples

Example 3-3 shows the output for **show interfaces bonding**.

Example 3-3 Displaying information about the bonding interfaces.

```
vyatta@vyatta:~$ show interfaces bonding
Interface    IP Address         State       Link   Description
bond3        10.192.136.2/29    up          up
bond3.128    10.192.128.2/24    up          up
```

Example 3-4 shows the output for **show interfaces bonding slaves**.

Example 3-4 Displaying information about the bonding slaves.

```
vyatta@vyatta:~$ show interfaces bonding slaves
Interface    Mode                 State    Link   Slaves
bond0        802.3ad              up       up     eth2 eth3
bond1        802.3ad              up       down   eth1
```

Chapter 4: Pseudo-Ethernet Interfaces

This chapter describes how to create a pseudo-Ethernet interface by defining multiple MAC addresses on a single physical interface.

This chapter presents the following topics:

Pseudo-Ethernet Interface Overview

A pseudo-Ethernet interface is a means of creating multiple virtual Ethernet devices, each with a different media access control (MAC) address, for a single physical Ethernet port. Pseudo-Ethernet interfaces have application in virtualized environments, where they can be used by other virtual machines.

Using pseudo-Ethernet interfaces requires less system overhead than using a traditional bridging approach. Pseudo-Ethernet interfaces also provide a means of working around the general limit of 4096 virtual LANs (VLANs) per physical Ethernet port, since that limit is with respect to a single MAC address.

Virtual Ethernet interfaces behave like real Ethernet devices. They are configured with IP address and network information, descriptions, and MAC addresses, and are associated with a physical Ethernet port using interfaces pseudo-ethernet <pethx> link <ethx> command. The virtual device inherits the characteristics (speed, duplexity, and so on) of the physical link with which is is associated.

Once defined, pseudo-Ethernet interfaces can be referenced in just the same way as Ethernet interfaces in firewall rules, quality of service (QoS) policies, and so on.

Note the following about pseudo-Ethernet interfaces:

- You cannot connect to a pseudo-Ethernet interface internal to a system from within that system. For example, if you try to ping a pseudo-Ethernet interface from the system on which it is defined, the ping will fail.
- Any loopback occurs at the IP level, in the same way as for other interfaces. Ethernet packets are not forwarded between pseudo-Ethernet interfaces.
- A pseudo-Ethernet interface cannot be part of an Ethernet link bonding interface.
- Pseudo-Ethernet interfaces may not work in environments that expect a network interface card (NIC) to have a single address; these may include the following:
  - VMware machines with default settings
  - Network switches with security settings allowing only a single address
  - ADSL modems that “learn” the MAC address of the NIC

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on pseudo-Ethernet interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

Pseudo-Ethernet Interface Configuration Examples

This section presents the following topics:

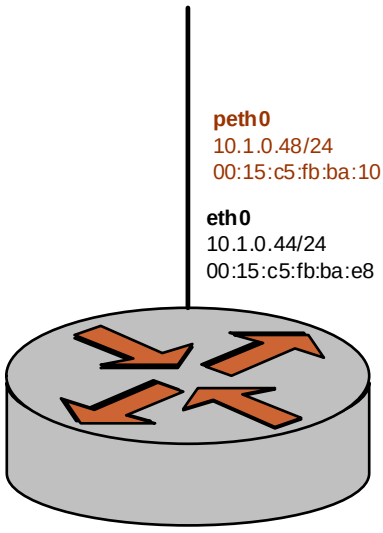
Creating a Pseudo-Ethernet Interface

Figure 4-1 shows a simple pseudo-Ethernet interface scenario. In this example:

- Ethernet interface eth0 is configured with IP address 10.1.0.44/24 and has a hardware MAC address of 00:15:c5:fb:ba:e8.
- Pseudo-Ethernet interface peth0 is associated with eth0 as the physical Ethernet link. It is configured with IP address 10.1.0.48/24 and is configured with a MAC address of 00:15:c5:fb:ba:10.

Note that the pseudo-Ethernet interface need not have the same network prefix as the physical interface. For example, an address of 10.1.0.48/32 is also valid in this scenario.

Figure 4-1 Creating a pseudo-Ethernet interface



To configure this scenario, perform the following steps in configuration mode.

Example 4-1 Creating a pseudo-Ethernet interface

Step	Command
Create the pseudo-Ethernet interface and assign it an address.	`vyatta@vyatta# set interfaces pseudo-ethernet peth0 address 10.1.1.1/24`
Provide a description for the interface.	`vyatta@vyatta# set interfaces pseudo-ethernet peth0 description “Sample virtual Ethernet interface”`
Link the pseudo-Ethernet interface to the physical Ethernet port.	`vyatta@vyatta# set interfaces pseudo-ethernet peth0 link eth0`
Set the MAC address for the pseudo-Ethernet interface	`vyatta@vyatta# set interfaces pseudo-ethernet peth0 mac 00:15:c5:fb:ba:10`
Commit the change.	`vyatta@vyatta# commit`

Show the pseudo-Ethernet interface configuration.	`vyatta@vyatta# show interfaces pseudo-ethernet peth0`
	` address 10.1.1.1/24`
	` description “Sample virtual Ethernet interface”`
	` link eth0`
	` mac 00:15:c5:fb:ba:10`

IPv6 on Pseudo-Ethernet Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Pseudo-Ethernet Interface Commands

Configuration Commands	
interfaces pseudo-ethernet <pethx>	Defines a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> address	Sets an IP address and network prefix for a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> description <descr>	Specifies a description for a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> dhcpv6-options	Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.
interfaces pseudo-ethernet <pethx> disable	Disables a pseudo-Ethernet interface without discarding configuration.
interfaces pseudo-ethernet <pethx> disable-link-detect	Directs a pseudo-Ethernet interface not to detect physical link-state changes.
interfaces pseudo-ethernet <pethx> ipv6 address	Assigns an IPv6 address to a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> ipv6 disable-forwarding	Disables IPv6 forwarding on pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces pseudo-ethernet <pethx> ipv6 router-advert	Specifies the router advertisements to be sent from pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> link <ethx>	Specifies the physical Ethernet interface associated with a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> mac <mac-addr>	Sets the MAC address of a pseudo-Ethernet interface.
interfaces pseudo-ethernet <pethx> redirect <interface>	Redirects inbound traffic from from one interface to another interface.
Operational Commands	
All operational commands applying to Ethernet interfaces can be used with pseudo-Ethernet interfaces. For these commands, see “Chapter 2: Ethernet Interfaces.”	

Commands for using other system features with pseudo-Ethernet interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Firewall	Firewall is supported on pseudo-Ethernet interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.
OSPF	OSPF is supported on pseudo-Ethernet interfaces. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on pseudo-Ethernet interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
QoS	Quality of service traffic policies are supported on pseudo-Ethernet interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on pseudo-Ethernet interfaces. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on pseudo-Ethernet interfaces. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.
VLAN interfaces	802.1Q VLAN operation is supported on pseudo-Ethernet interfaces. Commands for VLAN interfaces (vifs) are described in "Chapter 6: VLAN Interfaces."

interfaces pseudo-ethernet <pethx>

Defines a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx*

delete interfaces pseudo-ethernet *pethx*

show interfaces pseudo-ethernet *pethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   pseudo-ethernet pethx {}
}
```

Parameters

pethx	Multi-node. The identifier for the pseudo-Ethernet interface you are defining; for example **peth0**. You can define multiple pseudo-interfaces by creating multiple **pseudo-ethernet** configuration nodes.

Default

None.

Usage Guidelines

Use this command to define a virtual Ethernet device, or pseudo-Ethernet interface, by associating multiple media access control (MAC) addresses with a single physical Ethernet interface.

There is no necessary association between the physical interface and the integer in the pseudo-Ethernet interface name; for example, peth0 need not be a sub-device of eth0.

Once the pseudo-Ethernet interface is defined, the MAC address can be set using interfaces pseudo-ethernet <pethx> mac <mac-addr> command in the same manner as a physical Ethernet port.

Use the **set** form of this command to create a pseudo-Ethernet interface.

Use the **delete** form of this command to remove a pseudo-Ethernet interface.

Use the **show** form of this command to view pseudo-Ethernet interface configuration.

interfaces pseudo-ethernet <pethx> address

Sets an IP address and network prefix for a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

delete interfaces pseudo-ethernet *pethx* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

show interfaces pseudo-ethernet *pethx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        address [ipv4|ipv6|dhcp|dhcpv6]
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.
ipv4	Defines an IPv4 address on this interface. The format is *ip-address/prefix* (for example, 192.168.1.77/24). You can define multiple IP addresses for a single pseudo-Ethernet interface, by creating multiple **address** configuration nodes.
ipv6	Defines an IPv6 address on this interface. The format is *ipv6-address/prefix* (for example, 2001:db8:1234::/48). You can define multiple IPv6 addresses for a single pseudo-Ethernet interface, by creating multiple **address** configuration nodes.
dhcp	Defines the interface as a Dynamic Host Configuration Protocol (DHCP) client, which obtains its address and prefix from a DHCP server.

dhcpv6	Defines the interface as a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) client, which obtains its address and prefix from a DHCPv6 server.

Default

None.

Usage Guidelines

Use this command to set the IP address and network prefix for a pseudo-Ethernet interface.

Use the **set** form of this command to set the IP address and network prefix. You can set more than one IP address for the interface by creating multiple **address** configuration nodes.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces pseudo-ethernet <pethx> description <descr>

Specifies a description for a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **description** *descr*

delete interfaces pseudo-ethernet *pethx* **description**

show interfaces pseudo-ethernet *pethx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   pseudo-ethernet pethx {
          description descr
   }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth*x***, where x is a positive integer.
descr	A mnemonic name or description for the pseudo-Ethernet interface.

Default

None.

Usage Guidelines

Use this command to set a description for a pseudo-Ethernet interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

interfaces pseudo-ethernet <pethx> dhcpv6-options

Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.

Syntax

set interfaces pseudo-ethernet *pethx* **dhcpv6-options** [**parameters-only** | **temporary**]

delete interfaces pseudo-ethernet *pethx* **dhcpv6-options** [**parameters-only** | **temporary**]

show interfaces pseudo-ethernet *pethx* **dhcpv6-options**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        dhcpv6-options [parameters-only|temporary]
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.
parameters-only	Acquires only configuration parameters (and not an IPv6 address) from the DHCPv6 server. Only one of the **parameters-only** and the **temporary** parameter may be specified.
temporary	Acquires a temporary IPv6 address as described for IPv6 privacy addressing in RFC 4941. Only one of the **parameters-only** and the **temporary** parameter may be specified.

Default

None.

Usage Guidelines

Use this command to specify in what way the DHCPv6 client is to acquire an IPv6 address and/or parameters from a DHCPv6 server.

Note that these parameters are only relevant if the **dhcpv6** option has been set for the interfaces pseudo-ethernet <pethx> address command.

The **parameters-only** option is typically used in conjunction with SLAAC or static address configuration. It and the **temporary** parameter are mutually exclusive.

Use the **set** form of this command to specify the DHCPv6 options.

Use the **delete** form of this command to remove the DHCPv6 options.

Use the **show** form of this command to view DHCPv6 option configuration.

interfaces pseudo-ethernet <pethx> disable

Disables a pseudo-Ethernet interface without discarding configuration.

Syntax

set interfaces pseudo-ethernet *pethx* **disable**

delete interfaces pseudo-ethernet *pethx* **disable**

show interfaces pseudo-ethernet *pethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   pseudo-ethernet pethx {
          disable
   }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.

Default

None.

Usage Guidelines

Use this command to disable a pseudo-Ethernet interface without discarding configuration.

Use the **set** form of this command to disable the interface.

Use the **delete** form of this command to enable the interface.

Use the **show** form of this command to view pseudo-Ethernet interface configuration.

interfaces pseudo-ethernet <pethx> disable-link-detect

Directs a pseudo-Ethernet interface not to detect physical link-state changes.

Syntax

set interfaces pseudo-ethernet *pethx* **disable-link-detect**

delete interfaces pseudo-ethernet *pethx* **disable-link-detect**

show interfaces pseudo-ethernet *pethx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        disable-link-detect
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.

Default

The interface detects physical link state changes.

Usage Guidelines

Use this command to direct a pseudo-Ethernet interface to not detect physical state change to the Ethernet link it is associated with (for example, when the cable is unplugged).

Use the **set** form of this command to disable detection of physical state changes.

Use the **delete** form of this command to enable detection of physical state changes.

Use the **show** form of this command to view pseudo-Ethernet interface configuration.

interfaces pseudo-ethernet <pethx> ipv6 address

Assigns an IPv6 address to a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

delete interfaces pseudo-ethernet *pethx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

show interfaces pseudo-ethernet *pethx* **ipv6 address** [**autoconf** | **eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        ipv6 {
            address {
                autoconf
                eui64 ipv6prefix
            }
        }
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.
autoconf	Generates an IPv6 address using the SLAAC protocol. Set this value if the interface is performing a "host" function rather than a "router" function. This value can be specified in addition to specifying static IPv6, static IPv4, or IPv4 DHCP addresses on the interface.
ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address, in EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value derived from the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to assign an IPv6 address to pseudo-Ethernet interface.

You can using the **autoconf** keyword to direct the system to autoconfigure the address, using the SLAAC (Stateless Address Auto-Configuration) protocol defined in RFC 4862. Alternatively, you can provide an EUI-64 IPv6 address prefix so that the system constructs the IPv6 address.

If you want the system to use SLAAC to acquire addresses on this interface, then in addition setting this parameter, you must also disable IPv6 forwarding, either globally (using the system ipv6 disable-forwarding command) or specifically on this interface (using the interfaces <interface> ipv6 disable-forwarding command).

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view IPv6 address configuration settings.

interfaces pseudo-ethernet <pethx> ipv6 disable-forwarding

Disables IPv6 forwarding on pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **ipv6 disable-forwarding**

delete interfaces pseudo-ethernet *pethx* **ipv6 disable-forwarding**

show interfaces pseudo-ethernet *pethx* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        ipv6 {
            disable-forwarding
        }
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on pseudo-Ethernet interface.

You can also disable IPv6 forwarding globally (that is, for all interfaces) using the system ipv6 disable-forwarding command.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

Use the **show** form of this command to display IPv6 packet forwarding interface configuration.

interfaces pseudo-ethernet <pethx> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces pseudo-ethernet *pethx* **ipv6 dup-addr-detect-transmits** *num*

delete interfaces pseudo-ethernet *pethx* **ipv6 dup-addr-detect-transmits**

show interfaces pseudo-ethernet *pethx* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        ipv6 {
            dup-addr-detect-transmits num
        }
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **pethx**, where x is a positive integer.
num	The number of times to transmit NS packets as part of the DAD process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view NS packet configuration for DAD.

interfaces pseudo-ethernet <pethx> ipv6 router-advert

Specifies the router advertisements to be sent from pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **ipv6 router-advert** [**cur-hop-limit** *limit*] [**default-lifetime** *lifetime*] [**default-preference** *preference*] [**link-mtu** *mtu*] [**managed-flag** *state*] [**max-interval** *interval*] [**min-interval** *interval*] [**other-config-flag** *state*] [**prefix** *ipv6net* [**autonomous-flag** *state* | **on-link-flag** *state* | **preferred-lifetime** *lifetime* | **valid-lifetime** *lifetime*]] [**reachable-time** *time*] [**retrans-timer** *time*] [**send-advert** *state*]

delete interfaces pseudo-ethernet *pethx* **ipv6 router-advert** [**cur-hop-limit**] [**default-lifetime**] [**default-preference**] [**link-mtu**] [**managed-flag**] [**max-interval**] [**min-interval**] [**other-config-flag**] [**prefix** *ipv6net* [**autonomous-flag** | **on-link-flag** | **preferred-lifetime** | **valid-lifetime**]] [**reachable-time**] [**retrans-timer** [**send-advert**]

show interfaces pseudo-ethernet *pethx* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        ipv6 {
            router-advert {
                cur-hop-limit limit
                default-lifetime lifetime
                default-preference preference
                link-mtu mtu
                managed-flag state
                max-interval interval
                min-interval interval
                other-config-flag state
                prefix ipv6net {
                    autonomous-flag state
                    on-link-flag state
                    preferred-lifetime lifetime
                    valid-lifetime lifetime
                }
                reachable-time time
```

```
            retrans-timer time
            send-advert state
        }
    }
  }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **pethx**, where x is a positive integer.
cur-hop-limit *limit*	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets. This value is placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range is 0 to 255. The default is 64. A value of 0 means unspecified by the router.
default-lifetime lifetime	Specifies the lifetime , in seconds, associated with the default router. Supported values are 0, which indicates that the router is not a default router, and the range from the value configured for the **max-interval** option to 9000 (18.2 hours). If not configured, the value for this timer is three times **max-interval**.
default-preference *preference*	The preference associated with the default router. Supported values are as follows: **low**: The default router is low preference. **medium**: The default router is medium preference. **high**: The default router is high preference. The default is **medium**.
link-mtu *mtu*	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the type of link, as defined in RFC 2464. The default is 0, which means the MTU is not specified in the router advertisement message. That is because it is expected that the MTU will configured directly on the interface itself and not for routing advertisements. You can configure this option in cases where the link MTU is not well known. If the value set here does not match the MTU configured on the interface, the system issues a warning but does not fail.

managed-flag *state*	Whether to use the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration. The default is **false**.
max-interval *interval*	The maximum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default is 600 (10 minutes).
min-interval *interval*	The minimum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * **max-interval**. The default is 0.33 * **max-interval**.
other-config-flag *state*	The interfaces uses the administered (stateful) protocol for autoconfiguration of non-address information, as defined in RFC 4862. Supported values are as follows: **true**: Hosts use the administered protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default is **false**.
prefix *ipv6net*	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface, in the format *ipv6-address/prefix*. You can define more than one IPv6 prefix by configuring multiple **prefix** configuration nodes.
autonomous-flag *state*	Specifies whether the prefix can be used for autonomous address configuration as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default is **true**.

on-link-flag *state*	Specifies whether the prefix can be used for on-link determination, as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default is **true**.
preferred-lifetime *lifetime*	The length of time, in seconds, that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) is to remain preferred, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 604800 (seven days).
valid-lifetime *lifetime*	The length of time, in seconds, that the prefix is valid for the purpose of on-link determination, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 2592000 (30 days).
reachable-time *time*	The length of time, in milliseconds, for which the system assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range is 0 to 3600000, where a value of 0 means the reachable time is not specified in the router advertisement message. The default is 0.
retrans-timer *time*	The length of time, in milliseconds, between retransmitted NS messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295, where a value of 0 means the retransmit tme is not specified in the router advertisement message. The default is 0.

send-advert *state*	Specifies whether router advertisements are to be sent from this interface. Supported values are as follows: **true**: Sends router advertisements from this interface. **false**: Does not send router advertisements from this interface. If this value is in effect, parameters in this configuration subtree are still used to configure the local implementation parameters. The default is **true**.

Default

Router advertisements are not sent on an interface.

Usage Guidelines

Use this command to configure router advertisements (RAs) to be sent out of the interface being configured.

Router advertisements are sent out by IPv6 routers in order to advertise their existence to hosts on the network. IPv6 hosts do not send out router advertisements.

If the **router-advert** node of the configuration tree is missing, router advertisements are not sent out. Also, if IPv6 forwarding is disabled either globally (using the **system ipv6 disable-forwarding** command) or on the interface (using the interfaces pseudo-ethernet <pethx> ipv6 disable-forwarding command), router advertisements are not sent out.

Most router advertisement parameters are required by either the Neighbor Discovery (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view router advertisement configuration.

interfaces pseudo-ethernet <pethx> link <ethx>

Specifies the physical Ethernet interface associated with a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **link** *ethx*

delete interfaces pseudo-ethernet *pethx* **link**

show interfaces pseudo-ethernet *pethx* **link**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   pseudo-ethernet pethx {
      link ethx
   }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth*x***, where x is a positive integer.
link	Mandatory. The physical Ethernet interface associated with the pseudo-Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system. The suffixes for **pethx** and **ethx** need not be the same (e.g. peth4 could reside on eth1).

Default

None.

Usage Guidelines

Use this command to specify which physical Ethernet interface is to be associated with a pseudo-Ethernet interface.

Use the **set** form of this command to specify the Ethernet interface.

Use the **delete** form of this command to remove the Ethernet interface. Note that specifying a physical Ethernet link is mandatory in a minimal configuration..

Use the **show** form of this command to view physical Ethernet link configuration for a pseudo-Ethernet interface.

interfaces pseudo-ethernet <pethx> mac <mac-addr>

Sets the MAC address of a pseudo-Ethernet interface.

Syntax

set interfaces pseudo-ethernet *pethx* **mac** *mac-addr*

delete interfaces pseudo-ethernet *pethx* **mac**

show interfaces pseudo-ethernet *pethx* **mac**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        mac mac-addr
    }
}
```

Parameters

pethx	The identifier of the pseudo-Ethernet interface. The range is **peth0** through **peth***x*, where x is a positive integer.
mac-addr	The MAC address to be set for the pseudo-Ethernet interface. The format is 6 colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

If no MAC address is specified, the system automatically generates one for the interface.

Usage Guidelines

Use this command to specify a MAC address for a pseudo-Ethernet interface.

Use the **set** form of this command to specify the the MAC address for the pseudo-Ethernet interface.

Use the **delete** form of this command to remove the MAC address.

Use the **show** form of this command to view the MAC address configuration for a pseudo-Ethernet interface.

interfaces pseudo-ethernet <pethx> redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces pseudo-ethernet *pethx* **redirect** *interface*

delete interfaces pseudo-ethernet *pethx* **redirect** *interface*

show interfaces pseudo-ethernet *pethx* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    pseudo-ethernet pethx {
        redirect interface
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

Chapter 5: Wireless Interfaces

This chapter explains how to work with wireless interfaces on the Vyatta system.

This chapter presents the following topics:

Wireless Interface Overview

The wireless LAN (WLAN) interface provides 802.11 wireless support (commonly referred to as Wi-Fi) by means of compatible hardware. If the hardware supports it, the Vyatta system can support multiple logical wireless interfaces per physical device.

The three modes of operation for a wireless interface are as a wireless access point (WAP), as a station, and as a monitor.

- If the hardware supports acting as a WAP, the system provides network access to connecting Stations.
- As a station, the system acts as a client accessing the network through an available WAP.
- As a monitor, the system passively monitors wireless traffic.

If the system detects an unconfigured wireless device, it adds the device to the configuration tree, specifying any detected settings (for example, its MAC address) and set as a monitor type.

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on wireless interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

Wireless Interface Configuration Examples

This section presents the following topics:

Configuring a Wireless Access Point

The example in this section creates a WAP. When configuring multiple WAP interfaces, you must specify unique IP addresses, channels, Network IDs (SSIDs), and MAC addresses.

The WAP in this example has the following characteristics:

- IP address 192.168.40.1/24
- Network ID (SSID) “Test”
- WPA passphrase “Test phrase”
- Uses the 802.11n protocol

- Operates on channel 1

In this example, the default physical device (phy0) is used and a MAC address is generated.

To create this WAP, perform the following steps in configuration mode.

Example 5-1 Configuring an Access Point

Step	Command
Create a wireless interface and specify that it is to be a Wireless Access Point.	`vyatta@R1# set interfaces wireless wlan0 type access-point`
Specify the IP address.	`vyatta@R1# set interfaces wireless wlan0 address 192.168.40.1/24`
Specify the Network ID.	`vyatta@R1# set interfaces wireless wlan0 ssid Test`
Specify the WPA passphrase.	`vyatta@R1# set interfaces wireless wlan0 security wpa passphrase “Test phrase”`
Specify the 802.11 mode.	`vyatta@R1# set interfaces wireless wlan0 mode n`
Specify the channel.	`vyatta@R1# set interfaces wireless wlan0 channel 1`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show interfaces wireless`
	`wireless wlan0 {`
	`  address 192.168.40.1/24`
	`  channel 1`
	`  mode n`
	`  security {`
	`  wpa {`
	`  passphrase "Test phrase"`
	`  }`
	`  }`
	`  ssid Test`
	`  type access-point`
	`}`

Configuring a Wireless Station

The example in this section creates a wireless station (that is, a client) that accesses the network through the WAP defined in Example 5-1. The default physical device (phy0) is used.

To configure a wireless interface as a station in this way, perform the following steps in configuration mode.

Example 5-2 Configuring a Station

Step	Command
Create a wireless interface and specify that it is to be a Station (i.e. a client).	`vyatta@R2# set interfaces wireless wlan0 type station`
Specify that the IP address will be provided by a DHCP server on the network.	`vyatta@R2# set interfaces wireless wlan0 address dhcp`
Specify the network id.	`vyatta@R2# set interfaces wireless wlan0 ssid Test`
Specify the WPA passphrase.	`vyatta@R2# set interfaces wireless wlan0 security wpa passphrase “Test phrase”`
Commit the changes.	`vyatta@R2# commit`

Show the configuration.	`vyatta@R2# show interfaces wireless`
	`wireless wlan0 {`
	`address dhcp`
	`security {`
	`wpa {`
	`passphrase "Test phrase"`
	`}`
	`}`
	`ssid Test`
	`type station`
	`}`

IPv6 on Wireless Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Wireless Interface Commands

Configuration Commands	
interfaces wireless <wlanx>	Defines a wireless interface.
interfaces wireless <wlanx> address	Sets an IP address and network prefix for a wireless interface.
interfaces wireless <wlanx> channel <channel>	Sets the channel the wireless interface uses.
interfaces wireless <wlanx> country <country>	Sets the country that the wireless interface is deployed in.
interfaces wireless <wlanx> description <descr>	Specifies a description for a wireless interface.
interfaces wireless <wlanx> disable-broadcast-ssid	Sets the wireless interface not to broadcast SSID.
interfaces wireless <wlanx> disable-link-detect	Directs a wireless interface not to detect physical link-state changes.
interfaces wireless <wlanx> ipv6 address	Assigns an IPv6 address to a wireless interface.
interfaces wireless <wlanx> ipv6 disable-forwarding	Disables IPv6 forwarding on a wireless interface.
interfaces wireless <wlanx> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces wireless <wlanx> ipv6 router-advert	Specifies the router advertisements to be sent from a wireless interface.
interfaces wireless <wlanx> mac <mac-addr>	Sets the Media Access Control (MAC) address for a wireless interface.
interfaces wireless <wlanx> mode <mode>	Sets the 802.11 mode for a wireless interface.
interfaces wireless <wlanx> physical-device <device>	Associates a physical device with a wireless interface.
interfaces wireless <wlanx> redirect <interface>	Redirects inbound traffic from from one interface to another interface.
interfaces wireless <wlanx> security wep key <key>	Enables WEP encryption for a wireless interface and specifies the encryption key.
interfaces wireless <wlanx> security wpa	Sets the the encryption cipher for WPA encryption.
interfaces wireless <wlanx> ssid <ssid>	Specifies the SSID for a wireless interface.
interfaces wireless <wlanx> type <type>	Specifies the wireless device type for the wireless interface.

Operational Commands	
monitor interfaces wireless <wlanx> traffic	Displays (captures) traffic on a wireless interface.
show interfaces wireless	Displays status and statistics for wireless interfaces.
show interfaces wireless <wlanx>	Displays status and statistics for a wireless interface.
show interfaces wireless <wlanx> brief	Displays brief summary status for a wireless interface.
show interfaces wireless <wlanx> queue	Displays wireless interface queuing information.
show interfaces wireless <wlanx> scan	Scans for nearby wireless networks.
show interfaces wireless <wlanx> stations	Displays information about stations connected wirelessly to a wireless interface.

Commands for using other system features with wireless interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Bridging	Layer 2 bridging is supported on wireless interfaces configured as wireless access points (not in station mode). Commands for configuring bridge groups are described in the *Vyatta Bridging Reference Guide*.
Firewall	Firewall is supported on wireless interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.
OSPF	OSPF is supported on wireless interfaces. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on wireless interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
QoS	Quality of service traffic policies are supported on wireless interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on wireless interfaces. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on wireless interfaces. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.
VLAN interfaces	802.1Q VLAN operation is supported on wireless interfaces. Commands for configuring VLAN interfaces (vifs) are described in "Chapter 6: VLAN Interfaces."

interfaces wireless <wlanx>

Defines a wireless interface.

Syntax

set interfaces wireless *wlanx*

delete interfaces wireless *wlanx*

show interfaces wireless *wlanx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
   }
}
```

Parameters

wlanx	Mandatory. Multi-node. The identifier for the wireless interface. The range is **wlan0** to **wlan999**. You can define multiple wireless interfaces by creating more than one **wireless** configuration node.

Default

None.

Usage Guidelines

Use this command to configure a wireless interface. You can define multiple wireless interfaces by creating multiple **wireless** configuration nodes.

NOTE *Creating multiple **wireless** configuration nodes on the same physical device is supported for some driver / hardware combinations.*

Note that you cannot use **set** to change the name of the wireless interface. To change the name of a wireless interface, you must delete the old **wireless** configuration node and create a new one.

Use the **set** form of this command to create a wireless interface. Once the interface is created its status can be viewed using the **monitor interfaces wireless <wlanx> traffic** command.

Use the **delete** form of this command to remove all configuration for a wireless interface.

Use the **show** form of this command to view a wireless interface configuration.

interfaces wireless <wlanx> address

Sets an IP address and network prefix for a wireless interface.

Syntax

set interfaces wireless *wlanx* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

delete interfaces wireless *wlanx* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

show interfaces wireless *wlanx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      address [ipv4|ipv6|dhcp|dhcpv6]
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
ipv4	Multi-node. Defines an IPv4 address on this interface. The format is *ip-address/prefix* (for example, 192.168.1.77/24). You can define multiple IP addresses for a single interface, by creating multiple **address** configuration nodes.
ipv6	Multi-node. Defines an IPv6 address on this interface. The format is *ipv6-address/prefix* (for example, 2001:db8:1234::/48). You can define multiple IPv6 addresses for a single interface, by creating multiple **address** configuration nodes.
dhcp	Identifies the interface as a Dynamic Host Configuration Protocol (DHCP) client, which obtains its address and prefix from a DHCP server.

dhcpv6	Defines the interface as a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) client, which obtains its address, prefix, and parameters from a DHCPv6 server.

Default

None.

Usage Guidelines

Use this command to set the IP address and network prefix for a wireless interface.

Use the **set** form of this command to set the IP address and network prefix. You can set more than one IP address for the interface by creating multiple **address** configuration nodes.

Use the **delete** form of this command to remove IP address configuration.

Use the **show** form of this command to view IP address configuration.

interfaces wireless <wlanx> channel <channel>

Sets the channel the wireless interface uses.

Syntax

set interfaces wireless *wlanx* **channel** *channel*

delete interfaces wireless *wlanx* **channel** *channel*

show interfaces wireless *wlanx* **channel**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        channel channel
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
channel	The channel the interface is to use. The range is 1 to 14. By default, the hardware selects the channel.

Default

The wireless hardware selects the channel.

Usage Guidelines

Use this command to set the channel for a wireless interface. In most cases, interfaces where **type** is set to **station** should not set the channel explicitly. This allows the hardware do it automatically. For interfaces where **type** is set to **access-point**, the channel must be set explicitly using this command.

Use the **set** form of this command to set the channel.

Use the **delete** form of this command to remove the channel configuration.

Use the **show** form of this command to view channel configuration.

interfaces wireless <wlanx> country <country>

Sets the country that the wireless interface is deployed in.

Syntax

set interfaces wireless *wlanx* **country** *country*

delete interfaces wireless *wlanx* **country** *country*

show interfaces wireless *wlanx* **country**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      country country
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
country	The country in which the wireless interface is deployed. The value is a two-letter country code as defined in the ISO 639 standard. Examples are **US**, **EU**, and **JP**). The default is **US**.

Default

The country is **US**.

Usage Guidelines

Use this command to set the country that a wireless interface is deployed in. This controls the allowable frequencies and power used, based on the regulations for the specified country.

Use the **set** form of this command to set the country.

Use the **delete** form of this command to remove the country configuration.

Use the **show** form of this command to view country configuration.

interfaces wireless <wlanx> description <descr>

Specifies a description for a wireless interface.

Syntax

set interfaces wireless *wlanx* **description** *descr*

delete interfaces wireless *wlanx* **description**

show interfaces wireless *wlanx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      description descr
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
descr	A mnemonic name or description for the wireless interface.

Default

None.

Usage Guidelines

Use this command to set a description for a wireless interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

interfaces wireless <wlanx> dhcpv6-options

Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.

Syntax

set interfaces wireless *wlanx* **dhcpv6-options** [**parameters-only** | **temporary**]

delete interfaces wireless *wlanx* **dhcpv6-options** [**parameters-only** | **temporary**]

show interfaces wireless *wlanx* **dhcpv6-options**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        dhcpv6-options [parameters-only|temporary]
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
parameters-only	Acquires only configuration parameters (and not an IPv6 address) from the DHCPv6 server. Only one of the **parameters-only** and the **temporary** parameter may be specified.
temporary	Acquires a temporary IPv6 address as described for IPv6 privacy addressing in RFC 4941. Only one of the **parameters-only** and the **temporary** parameter may be specified.

Default

None.

Usage Guidelines

Use this command to specify in what way the DHCPv6 client is to acquire an IPv6 address and/or parameters from a DHCPv6 server.

Note that these parameters are only relevant if the **dhcpv6** option has been set for the interfaces wireless <wlanx> address command.

The **parameters-only** option is typically used in conjunction with SLAAC or static address configuration. It and the **temporary** parameter are mutually exclusive.

Use the **set** form of this command to specify the DHCPv6 options.

Use the **delete** form of this command to remove the DHCPv6 options.

Use the **show** form of this command to view DHCPv6 option configuration.

interfaces wireless <wlanx> disable-broadcast-ssid

Sets the wireless interface not to broadcast SSID.

Syntax

set interfaces wireless *wlanx* **disable-broadcast-ssid**

delete interfaces wireless *wlanx* **disable-broadcast-ssid**

show interfaces wireless *wlanx* **disable-broadcast-ssid**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      disable-broadcast-ssid
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

The SSID is broadcast.

Usage Guidelines

Use this command to disable broadcasting of the Service Set Identifier (SSID) by the wireless interface. Disabling transmission of the SSID is typically used to hide a Wireless Access Point.

NOTE *This parameter is only valid when the interface is configured as a Wireless Access Point (that is,* ***type*** *is* ***access-point****). If the interface is configured as a Station (that is,* ***type*** *is* ***station****), this value is ignored.*

Use the **set** form of this command to disable SSID broadcasting.

Use the **delete** form of this command to enable SSID broadcasting.

Use the **show** form of this command to see whether SSID broadcasting is enabled or disabled.

interfaces wireless <wlanx> disable-link-detect

Directs a wireless interface not to detect physical link-state changes.

Syntax

set interfaces wireless *wlanx* **disable-link-detect**

delete interfaces wireless *wlanx* **disable-link-detect**

show interfaces wireless *wlanx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      disable-link-detect
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

The interface detects physical link state changes.

Usage Guidelines

Use this command to direct a wireless interface to not detect physical state change to the wireless link.

NOTE *This parameter is only valid when the interface is configured as a Wireless Access Point (that is,* ***type*** *is* ***access-point****. If the interface is configured as a Station (that is,* ***type*** *is* ***station****), this value is ignored.*

Use the **set** form of this command to disable detection of physical state changes.

Use the **delete** form of this command to enable detection of physical state changes.

Use the **show** form of this command to view wireless interface configuration.

interfaces wireless <wlanx> ipv6 address

Assigns an IPv6 address to a wireless interface.

Syntax

set interfaces wireless *wlanx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

delete interfaces wireless *wlanx* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

show interfaces wireless *wlanx* **ipv6 address** [**autoconf** | **eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      ipv6 {
         address {
            autoconf
            eui64 ipv6prefix
         }
      }
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
autoconf	Generates an IPv6 address using the SLAAC protocol. Set this value if the interface is performing a "host" function rather than a "router" function. This value can be specified in addition to specifying static IPv6, static IPv4, or IPv4 DHCP addresses on the interface.
ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address, in EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value derived from the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to assign an IPv6 address to a wireless interface.

You can using the **autoconf** keyword to direct the system to autoconfigure the address, using the SLAAC (Stateless Address Auto-Configuration) protocol defined in RFC 4862. Alternatively, you can provide an EUI-64 IPv6 address prefix so that the system constructs the IPv6 address.

If you want the system to use SLAAC to acquire addresses on this interface, then in addition setting this parameter, you must also disable IPv6 forwarding, either globally (using the system ipv6 disable-forwarding command) or specifically on this interface (using the interfaces wireless <wlanx> ipv6 disable-forwarding command).

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view IPv6 address configuration settings.

interfaces wireless <wlanx> ipv6 disable-forwarding

Disables IPv6 forwarding on a wireless interface.

Syntax

set interfaces wireless *wlanx* **ipv6 disable-forwarding**

delete interfaces wireless *wlanx* **ipv6 disable-forwarding**

show interfaces wireless *wlanx* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      ipv6 {
         disable-forwarding
      }
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on a wireless interface.

You can also disable IPv6 forwarding globally (that is, for all interfaces) using the system ipv6 disable-forwarding command.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

Use the **show** form of this command to display IPv6 packet forwarding interface configuration.

interfaces wireless <wlanx> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces wireless *wlanx* **ipv6 dup-addr-detect-transmits** *num*

delete interfaces wireless *wlanx* **ipv6 dup-addr-detect-transmits**

show interfaces wireless *wlanx* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        ipv6 {
            dup-addr-detect-transmits num
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
num	The number of times to transmit NS packets as part of the DAD process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view NS packet configuration for DAD.

interfaces wireless <wlanx> ipv6 router-advert

Specifies the router advertisements to be sent from a wireless interface.

Syntax

set interfaces wireless *wlanx* **ipv6 router-advert** [**cur-hop-limit** *limit*] [**default-lifetime** *lifetime*] [**default-preference** *preference*] [**link-mtu** *mtu*] [**managed-flag** *state*] [**max-interval** *interval*] [**min-interval** *interval*] [**other-config-flag** *state*] [**prefix** *ipv6net* [**autonomous-flag** *state* | **on-link-flag** *state* | **preferred-lifetime** *lifetime* | **valid-lifetime** *lifetime*]] [**reachable-time** *time*] [**retrans-timer** *time*] [**send-advert** *state*]

delete interfaces wireless *wlanx* **ipv6 router-advert** [**cur-hop-limit**] [**default-lifetime**] [**default-preference**] [**link-mtu**] [**managed-flag**] [**max-interval**] [**min-interval**] [**other-config-flag**] [**prefix** *ipv6net* [**autonomous-flag** | **on-link-flag** | **preferred-lifetime** | **valid-lifetime**]] [**reachable-time**] [**retrans-timer** [**send-advert**]

show interfaces wireless *wlanx* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        ipv6 {
            router-advert {
                cur-hop-limit limit
                default-lifetime lifetime
                default-preference preference
                link-mtu mtu
                managed-flag state
                max-interval interval
                min-interval interval
                other-config-flag state
                prefix ipv6net {
                    autonomous-flag state
                    on-link-flag state
                    preferred-lifetime lifetime
                    valid-lifetime lifetime
                }
                reachable-time time
```

```
                retrans-timer time
                send-advert state
            }
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
cur-hop-limit *limit*	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets. This value is placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range is 0 to 255. The default is 64. A value of 0 means unspecified by the router.
default-lifetime lifetime	Specifies the lifetime , in seconds, associated with the default router. Supported values are 0, which indicates that the router is not a default router, and the range from the value configured for the **max-interval** option to 9000 (18.2 hours). If not configured, the value for this timer is three times **max-interval**.
default-preference *preference*	The preference associated with the default router. Supported values are as follows: **low**: The default router is low preference. **medium**: The default router is medium preference. **high**: The default router is high preference. The default is **medium**.
link-mtu *mtu*	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the type of link, as defined in RFC 2464. The default is 0, which means the MTU is not specified in the router advertisement message. That is because it is expected that the MTU will configured directly on the interface itself and not for routing advertisements. You can configure this option in cases where the link MTU is not well known. If the value set here does not match the MTU configured on the interface, the system issues a warning but does not fail.

managed-flag *state*	Whether to use the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration. The default is **false**.
max-interval *interval*	The maximum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default is 600 (10 minutes).
min-interval *interval*	The minimum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * **max-interval**. The default is 0.33 * **max-interval**.
other-config-flag *state*	The interfaces uses the administered (stateful) protocol for autoconfiguration of non-address information, as defined in RFC 4862. Supported values are as follows: **true**: Hosts use the administered protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default is **false**.
prefix *ipv6net*	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface, in the format *ipv6-address/prefix*. You can define more than one IPv6 prefix by configuring multiple **prefix** configuration nodes.
autonomous-flag *state*	Specifies whether the prefix can be used for autonomous address configuration as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default is **true**.

on-link-flag *state*	Specifies whether the prefix can be used for on-link determination, as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default is **true**.
preferred-lifetime *lifetime*	The length of time, in seconds, that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) is to remain preferred, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 604800 (seven days).
valid-lifetime *lifetime*	The length of time, in seconds, that the prefix is valid for the purpose of on-link determination, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 2592000 (30 days).
reachable-time *time*	The length of time, in milliseconds, for which the system assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range is 0 to 3600000, where a value of 0 means the reachable time is not specified in the router advertisement message. The default is 0.
retrans-timer *time*	The length of time, in milliseconds, between retransmitted NS messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295, where a value of 0 means the retransmit tme is not specified in the router advertisement message. The default is 0.

send-advert *state*	Specifies whether router advertisements are to be sent from this interface. Supported values are as follows: **true**: Sends router advertisements from this interface. **false**: Does not send router advertisements from this interface. If this value is in effect, parameters in this configuration subtree are still used to configure the local implementation parameters. The default is **true**.

Default

Router advertisements are not sent on an interface.

Usage Guidelines

Use this command to configure router advertisements (RAs) to be sent out of the interface being configured.

Router advertisements are sent out by IPv6 routers in order to advertise their existence to hosts on the network. IPv6 hosts do not send out router advertisements.

If the **router-advert** node of the configuration tree is missing, router advertisements are not sent out. Also, if IPv6 forwarding is disabled either globally (using the **system ipv6 disable-forwarding** command) or on the interface (using the interfaces wireless <wlanx> ipv6 disable-forwarding command), router advertisements are not sent out.

Most router advertisement parameters are required by either the Neighbor Discovery (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view router advertisement configuration.

interfaces wireless <wlanx> mac <mac-addr>

Sets the Media Access Control (MAC) address for a wireless interface.

Syntax

set interfaces wireless *wlanx* **mac** *mac-addr*

delete interfaces wireless *wlanx* **mac**

show interfaces wireless *wlanx* **mac**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      mac mac-addr
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
mac-addr	Set the MAC address for a wireless interface. The format is 6 colon-spearated 8-bit numbers in hexidecimal; for example, 00:0a:59:9a:f2:ba.

Default

None.

Usage Guidelines

Use this command to set the MAC address for a wireless interface. Each wireless interface must have a unique MAC address in access-point mode.

Use the **set** form of this command to specify the MAC address.

Use the **delete** form of this command to remove the MAC address.

Use the **show** form of this command to view the MAC address configuration.

interfaces wireless <wlanx> mode <mode>

Sets the 802.11 mode for a wireless interface.

Syntax

set interfaces wireless *wlanx* **mode** *mode*

delete interfaces wireless *wlanx* **mode**

show interfaces wireless *wlanx* **mode**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        mode mode
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
mode	A letter indicating the 802.11 mode the wireless interface is to use. Supported values are as follows: **a**: Operates in accordance with the IEEE 802.11a-1999 amendment to the 802.11 specification (54 Mbps over a 5 GHz band). **b**: Operates in accordance with the IEEE 802.11b-1999 amendment to the IEEE 802.11 specification (11 Mbps over a 2.4 GHz band). **g**: Operates in accordance with the IEEE 802.11g-2003 specification (54 Mbps over a 2.4 GHz band). **n**: Operates in accordance with the IEEE 802.11n-2009 specification (up to 600 Mbps with four spatial streams over 40 MHz channels).

Default

The interface operates in accordance with the IEEE 802.11g-2003 specification.

Usage Guidelines

Use this command to set the 802.11 mode for a wireless interface. The IEEE 802.11 standard has undergone a number of revisions and amendments, which are referred to as 802.11a, 802.11b, and so on.

NOTE *This parameter is only valid when the interface is configured as a wireless access point (that is,* ***type*** *is* ***access-point****. If the interface is configured as a station (that is,* ***type*** *is* ***station****), this value is ignored.*

Use the **set** form of this command to specify the mode.

Use the **delete** form of this command to remove the mode.

Use the **show** form of this command to view the mode configuration.

interfaces wireless <wlanx> physical-device <device>

Associates a physical device with a wireless interface.

Syntax

set interfaces wireless *wlanx* **physical-device** *device*

delete interfaces wireless *wlanx* **physical-device**

show interfaces wireless *wlanx* **physical-device**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        physical-device device
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
device	A identifier representing physical device to associate with the wireless interface. The range is **phy0** to **phy9**.

Default

The first available physical device, **phy0**, is used.

Usage Guidelines

Use this command to specify the physical device associated with the wireless interface.

This value is optional for the first physical device but is required when there is more than one physical device.

NOTE *Some hardware supports the ability to have more than one wireless interface on the same physical device (on different frequencies).*

Use the **set** form of this command to specify the physical device associated with the wireless interface.

Use the **delete** form of this command to remove the physical device specification.

Use the **show** form of this command to view the physical device configuration.

interfaces wireless <wlanx> redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces wireless *wlanx* **redirect** *interface*

delete interfaces wireless *wlanx* **redirect** *interface*

show interfaces wireless *wlanx* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        redirect interface
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

interfaces wireless <wlanx> security wep key <key>

Enables WEP encryption for a wireless interface and specifies the encryption key.

Syntax

set interfaces wireless *wlanx* **security wep key** *key*

delete interfaces wireless *wlanx* **security wep key**

show interfaces wireless *wlanx* **security wep key**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      security {
         wep {
            key key
         }
      }
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
key	A 10, 26, or 32 digit hexadecimal key corresponding to 64-, 128-, or 152-bit WEP encryption respectively.

Default

The wireless interface is unencrypted.

Usage Guidelines

Use this command to enable Wired Equivalent Privacy (WEP) on a wireless interface and specify encryption key to be used.

NOTE *WEP encryption has been broken and is not secure. Relying on WEP to protect an interface is not recommended.*

NOTE *WEP and WPA security cannot both be configured on the same interface.*

Use the **set** form of this command to enable WEP security on the interface and specify the encryption key.

Use the **delete** form of this command to disable WEP encryption and restore the default behavior.

Use the **show** form of this command to view WEP configuration.

interfaces wireless <wlanx> security wpa

Sets the the encryption cipher for WPA encryption.

Syntax

set interfaces wireless *wlanx* **security wpa** [**cipher** *cipher* | **mode** *mode* | **passphrase** *passphrase* | **radius-server** *address* [**accounting** | **port** *port* | **secret** *secret*]]

delete interfaces wireless *wlanx* **security wpa** [**cipher** | **mode** | **passphrase** | **radius-server**]

show interfaces wireless *wlanx* **security wpa** [**cipher** | **mode** | **passphrase** | **radius-server**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        security {
            wpa {
                cipher cipher
                mode mode
                passphrase passphrase
                radius-server address {
                    accounting
                    port port
                    secret secret
                }
            }
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

cipher	The encryption algorithm to be used for broadcast and multicast frames in WPA mode. Note that the encryption value specified here is used is the WPA mode is WPA, but not if the mode is WPA2; WPA2 mode always uses CCMP encyption. Supported values are as follows: **CCMP**: Requires AES in Counter mode with CBC-MAC, according to the RFC 3610 and IEEE 802.11i/D7.0 specifications. **TKIP**: Requires Temporal Key Integrity Protocol according to the IEEE 802.11i/D7.0 specification. By default, both TKIP and CCMP are permitted; TKIP is tried first to support older clients.
mode	The WPA mode required for the wireless interface. Supported values are as follows: **wpa**: Requires WPA mode, according to the IEEE 802.11i/D3 specification. **wpa2**: Requires WPA2; that is, the full IEEE 802.11i/RSN specification. **both**: Allows both WPA and WPA2. The default is **both**.
passphrase	A string to be used as the WPA shared passphrase for the wireless interface. The passsphrase must be from 8 to 63 printable characters. If it includes spaces, the passphrase must be enclosed in double quotes.
address	Multi-node. The IP address of RADIUS server from which the wireless interface can retrieve WPA encryption keys to which it can send accounting information, if accounting is enabled. You can specify multiple RADIUS servers by creating multiple **radius-server** configuration nodes. If multiple RADIUS servers are specified, the secondary servers are used only if the first does not reply; servers are queried in the order in which they are configured.
accounting	Directs the wireless interface to send accounting information to the RADIUS server..
port	The RADIUS server port to use. By default, port **1812** is used, which is the well-known port for RADIUS.
secret	The secret to be used for accessing the RADIUS server.

Default

None.

Usage Guidelines

Use this command to enable Wired Protected Access (WPA) on a wireless interface and specify WPA parameters.

Note that when WPA is enabled, the interface may use either a passphrase as an encryption key (using the **passphrase** option) or may obtain encryption keys from a RADIUS server (using the **radius-server** option), but may not use both.

NOTE *WEP and WPA security cannot both be configured on the same interface.*

Use the **set** form of this command to enable WPA encryption and set WPA parameters.

Use the **delete** form of this command to disable WPA encryption and remove WPA configuration.

Use the **show** form of this command to view WPA configuration.

interfaces wireless <wlanx> ssid <ssid>

Specifies the SSID for a wireless interface.

Syntax

set interfaces wireless *wlanx* **ssid** *ssid*

delete interfaces wireless *wlanx* **ssid**

show interfaces wireless *wlanx* **ssid**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      ssid ssid
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
ssid	The Service Set Identifier (SSID) for the wireless interface. If the identifier contains space characters, it must be enclosed in double quotes.

Default

None.

Usage Guidelines

Use this command to specify the Service Set Identifier (SSID) for a wireless interface. This token is required for identifying the wireless network; setting this parameter is mandatory. The number of SSIDs that can be set on an interface depend on the hardware you are using.

Use the **set** form of this command to record the SSID.

Use the **delete** form of this command to remove SSID configuration.

Use the **show** form of this command to view SSID configuration.

interfaces wireless <wlanx> type <type>

Specifies the wireless device type for the wireless interface.

Syntax

set interfaces wireless *wlanx* **type** *type*

delete interfaces wireless *wlanx* **type**

show interfaces wireless *wlanx* **type**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      type type
   }
}
```

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
type	The wireless device type for the wireless interface. Supported values are as follows: **access-point**: The wireless interface provides wireless access to the network for clients. **monitor**: The wireless interface passively monitors wireless traffic. **station**: The wireless interface acts as a client on the wireless network.

Default

None.

Usage Guidelines

Use this command to specify the wireless device type for the wireless interface. Setting this parameter is mandatory. Bridging is only available to to interfaces configured as **access-points**.

Use the **set** form of this command to specify the device type for the wireless interface.

Use the **delete** form of this command to remove device type configuration.

Use the **show** form of this command to view device type configuration.

monitor interfaces wireless <wlanx> traffic

Displays (captures) traffic on a wireless interface.

Syntax

monitor interfaces wireless *wlanx* **traffic**

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to capture traffic on the specified wireless interface. Type <Ctrl>+c to stop the output.

Examples

Example 5-3 monitors captured data on interface wlan0.

Example 5-3 Displaying captured data

```
vyatta@vyatta:~$ monitor interfaces wireless wlan0 traffic
Capturing traffic on wlan0 ...
  0.000000 fe80::ad08:8661:4d:b925 -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  0.000067 fe80::69ca:5c11:bcf6:29da -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  2.608804 fe80::8941:71ef:b55d:e348 -> ff02::1:2    DHCPv6 Solicit
  3.010862 fe80::ad08:8661:4d:b925 -> ff02::c      SSDP M-SEARCH * HTTP/1.1
```

```
  3.010901 fe80::69ca:5c11:bcf6:29da -> ff02::c      SSDP M-SEARCH * HTTP/1.1
  4.568357 192.168.1.254 -> 238.255.255.251 SSDP NOTIFY * HTTP/1.1
  4.568372 192.168.1.254 -> 238.255.255.251 SSDP NOTIFY * HTTP/1.1
...
```

show interfaces wireless

Displays status and statistics for wireless interfaces.

Syntax

show interfaces wireless [detail | info]

Command Mode

Operational mode.

Parameters

detail	Displays detailed status information and statistics for all wireless interfaces.
info	Displays wireless-specific information about all wireless interfaces.

Default

Information is displayed for all wireless interfaces.

Usage Guidelines

Use this command to view operational status of wireless interfaces.

Examples

Example 5-4 shows information for all wireless interfaces.

Example 5-4 Displaying wireless interface information

```
vyatta@vyatta:~$ show interfaces wireless
Interface    IP Address         State       Link   Description
wlan0        192.168.40.1/24    up          up
```

Example 5-5 shows detailed information for all wireless interfaces.

Example 5-5 Displaying detailed wireless interfaces information

```
vyatta@vyatta:~$ show interfaces wireless detail
wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN0
    link/ether 00:21:91:d1:18:ca brd ff:ff:ff:ff:ff:ff

    RX:  bytes    packets     errors    dropped    overrun      mcast
            0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
            0          0          0          0          0          0
```

Example 5-6 shows wireless-specific information for all wireless interfaces.

Example 5-6 Displaying wireless-specific information for all wireless interfaces

```
vyatta@vyatta:~$ show interfaces wireless info
Interface    Type              SSID                Channel
mon.wlan0    monitor           -                   ?
wlan0        AP                testing             3
```

show interfaces wireless <wlanx>

Displays status and statistics for a wireless interface.

Syntax

show interfaces wireless *wlanx*

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to view status and statistics on the specified wireless interface.

Examples

Example 5-7 shows status and statistics on interface wlan0.

Example 5-7 Displaying status and statistics for a specific wireless interface

```
vyatta@vyatta:~$ show interfaces wireless wlan0
wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN0
    link/ether 00:21:91:d1:18:ca brd ff:ff:ff:ff:ff:ff

    RX:  bytes    packets     errors    dropped    overrun      mcast
            0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
            0          0          0          0          0          0
```

show interfaces wireless <wlanx> brief

Displays brief summary status for a wireless interface.

Syntax

show interfaces wireless *wlanx* **brief**

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to view brief status and statistics on the specified wireless interface.

Examples

Example 5-8 shows a brief status on interface wlan0.

Example 5-8 Displaying summary status for a wireless interface

```
vyatta@vyatta:~$ show interfaces wireless wlan0 brief
Interface    IP Address         State       Link   Description
wlan0        192.168.40.1/24    up          up
```

show interfaces wireless <wlanx> queue

Displays wireless interface queuing information.

Syntax

show interfaces wireless *wlanx* **queue** [**class** | **filter**]

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
class	Display queue classes for the specified interface.
filter	Display queue filters for the specified interface.

Default

None.

Usage Guidelines

Use this command to view wireless interface queue information.

Examples

Example 5-9 shows queue information for interface wlan0.

Example 5-9 Displaying wireless interface queue information

```
vyatta@vyatta:~$ show interfaces wireless wlan0 queue
qdisc pfifo_fast 0: root bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 810323 bytes 6016 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
```

show interfaces wireless <wlanx> scan

Scans for nearby wireless networks.

Syntax

show interfaces wireless *wlanx* **scan** [**detail**]

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.
detail	Displays detailed scan information for the specified wireless interface.

Default

Displays a list of wireless networks within range of the specified wireless interface.

Usage Guidelines

Use this command to view information about wireless networks within range of the specified wireless interface. This command is used on a wireless interface configured as a Station.

NOTE *Not all wireless drivers and wireless hardware support scanning. Please refer to your driver and wireless hardware documentation for details.*

Examples

Example 5-10 shows scan information on interface wlan0.

Example 5-10 Displaying scan information for a specific wireless interface

```
vyatta@vyatta:~$ show interfaces wireless wlan0 scan
Access-point       SSID                Chan Signal (dbm)
00:22:3f:b5:68:d6  Moore               1    -77
00:40:10:10:00:03  Jbridge2            11   -67
```

```
00:13:46:42:ff:fe  BubbaNet             10   -89
```

Example 5-11 shows detailed scan information on interface wlan0.

Example 5-11 Displaying detailed scan information for a specific wireless interface

```
vyatta@vyatta:~$ show interfaces wireless wlan0 scan detail
BSS 00:22:3f:b5:68:d6 (on wlan0)
   TSF: 13932293222787 usec (161d, 06:04:53)
   freq: 2412
   beacon interval: 100
   capability: ESS Privacy ShortSlotTime (0x0411)
   signal: -84.00 dBm
   SSID: Moore
   Supported rates: 1.0* 2.0* 5.5* 11.0* 18.0 24.0 36.0 54.0
   DS Paramater set: channel 1
   ERP: Barker_Preamble_Mode
   Extended supported rates: 6.0 9.0 12.0 48.0
   WPS:    * Version: 1.0
       * Manufacturer: NETGEAR, Inc.
       * Model: WGR614v8
       * Device name: WGR614v8 (Wireless AP)
       * Config methods: Label, PBC
   WPA:    * Version: 1
       * Group cipher: TKIP
       * Pairwise ciphers: TKIP
       * Authentication suites: PSK
       * Capabilities: 16-PTKSA-RC (0x000c)
     WMM: parameter: 01 80 00 03 a4 00 00 27 a4 00 00 42 43 5e 00 62 32 2f 00
```

show interfaces wireless <wlanx> stations

Displays information about stations connected wirelessly to a wireless interface.

Syntax

show interfaces wireless *wlanx* **stations**

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. The range is **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to display information about stations connected to a wireless interface. This command is used on a wireless interface configured as an Access Point.

Examples

Example 5-12 shows station data on interface wlan0.

Example 5-12 Displaying station data

```
vyatta@vyatta:~$ show interfaces wireless wlan0 stations
Station           Signal      RX: bytes  packets     TX: bytes  packets
00:1d:e0:30:26:3f  -45          59074      1409        75714      631
```

Chapter 6: VLAN Interfaces

This chapter lists the commands for configuring VLAN interfaces on Ethernet interfaces and Ethernet bonded links.

This chapter presents the following sections:

VLAN Interface Overview

This section presents the following topics:

VLAN Operation Using vifs

Some interface types can be configured for IEEE 802.1Q VLAN operation using a virtual interface. On the Vyatta system virtual interfaces for VLANs are called vifs. As the identifier of the vif, you specify the VLAN the vif is to connect to.

Each physical interface can be configured with multiple vifs. Then, like a physical Ethernet interface, each vif can have multiple addresses assigned to it.

Interface Types Supporting VLAN Operation

VLAN interfaces can be configured on the following interface types:

- Physical Ethernet interfaces
- Ethernet bonding links
- Pseudo-Ethernet interfaces
- Wireless interfaces

VLAN Operation vs. Multinetting

VLANs are identified by a 4-byte tag that is inserted in the front of the Layer 2 Ethernet header. Having this additional tag means that interfaces configured for 802.1Q are not compatible with standard Ethernet packets. When considering whether or not to use a VLAN interface, keep the following in mind:

- If you are using 802.1Q VLANs, create **vif** configuration nodes beneath the physical interface and assign the IP address to the vif.
- If you are not using 802.1Q, but you want to have multiple networks on the same physical interface (that is, you want to use multinetting, but not VLANs), simply create multiple address configuration nodes directly under the physical interface, without using vifs.

Simultaneous Ethernet and 802.1Q Operation

If your other network devices support it, an Ethernet interface may be used simultaneously as a standard port and an 802.1Q VLAN port. To do this, configure an IP address directly on the physical interface and then define a vif for the interface. Assign the VLAN ID as the vif identifier and configure an IP address for the vif. (This feature may not be compatible with all Ethernet switches: some switches require a physical Ethernet interface to be exclusively either a 802.1Q interface or a standard Ethernet interface.)

Referring to VLAN Interfaces in Commands

To refer to a vif within an **interfaces** command, such as **show interfaces** or **set interfaces**, specify the whole path to the **vif** configuration node, as in the following example.

```
show interfaces ethernet eth1 vif 40
```

When referring to the same vif within other commands—for example, enabling RIP on the interface—use the format *if-x.vlan-id*, where *if-x* is the interface type plus the interface identifier (for example, **eth1**, **bond0**, or **wlan2**) and *vlan-id* is the VLAN ID (and also the identifier of the vif). The following example refers to the vif on VLAN 40 configured under interface eth1.

```
set protocols rip interface eth1.40 address 10.10.40.65
```

IPv6 Support

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on VLAN interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

VLAN Interface Configuration Examples

This section presents the following topics:

- IPv6 on VLAN Interfaces

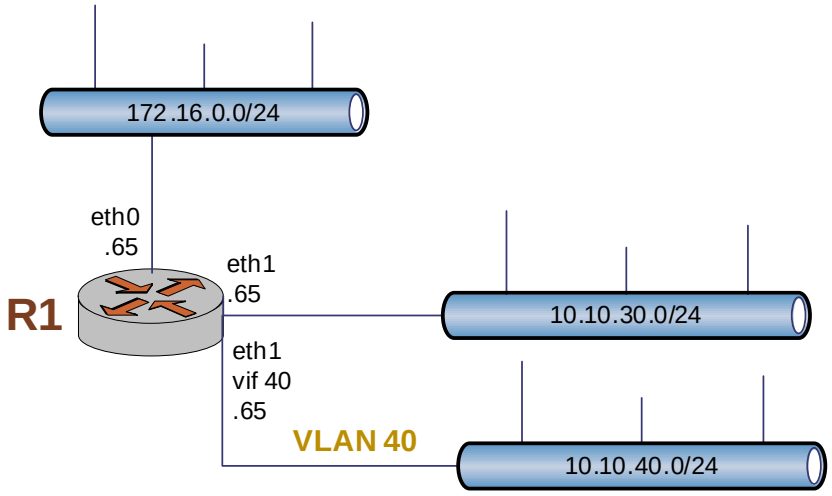
VLAN Configuration

This sequence configures router R1 to have a vif for VLAN 40 on Ethernet interface eth1. After configuring this VLAN, router R1 will have:

- One interface (eth0) that is configured as only a standard Ethernet interface. The IP address for this interface is 172.16.0.65.
- One interface (eth1) that is configured as both a physical Ethernet interface and a VLAN interface. The physical Ethernet interface has IP address 10.10.30.65. The VLAN interface connects to VLAN 40 (the identifier of the vif) and has IP address 10.10.40.65.

When you have finished, the interfaces will be configured as shown in Figure 6-1.

Figure 6-1 VLAN configuration



To create and configure a VLAN interface, perform the following steps in configuration mode.

Example 6-1 Simultaneous Ethernet and VLAN operation

Step	Command
Assign an IP address directly to the eth0 physical Ethernet interface.	`vyatta@R1# set interfaces ethernet eth0 address 172.16.0.65/24`
Assign an IP address directly to the eth1 physical Ethernet interface.	`vyatta@R1# set interfaces ethernet eth1 address 10.10.30.65/24`
Create the configuration node for the vif. Assign the vif VLAN ID 40. Assign the IP address for the vif.	`vyatta@R1# set interfaces ethernet eth1 vif 40 address 10.10.40.65/24`
Commit the configuration.	`vyatta@R1# commit`

Commit and view the configuration.	`vyatta@R1# show interfaces ethernet`
	`ethernet eth0 {`
	`address 172.16.0.65/24`
	`}`
	`ethernet eth1 {`
	`address 10.10.30.65/24`
	`vif 40 {`
	`address 10.10.40.65/24`
	`}`
	`}`

IPv6 on VLAN Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

VLAN Interface Commands

Configuration Commands	
interfaces <interface> vif <vlan-id>	Defines a virtual interface.
interfaces <interface> vif <vlan-id> address	Specifies an IP address and network prefix for a virtual interface.
interfaces <interface> vif <vlan-id> description <descr>	Sets a description for a virtual interface.
interfaces <interface> vif <vlan-id> dhcpv6-options	Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.
interfaces <interface> vif <vlan-id> disable	Disables a virtual interface without discarding configuration.
interfaces <interface> vif <vlan-id> disable-link-detect	Directs a virtual interface not to detect physical link-state changes.
interfaces <interface> vif <vlan-id> ipv6 address	Assigns an IPv6 address to a virtual interface.
interfaces <interface> vif <vlan-id> ipv6 disable-forwarding	Disables IPv6 forwarding on a virtual interface.
interfaces <interface> vif <vlan-id> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces <interface> vif <vlan-id> ipv6 router-advert	Specifies the router advertisements to be sent from a virtual interface.
interfaces <interface> vif <vlan-id> redirect <int>	Redirects inbound traffic from from one interface to another interface.
Operational Commands	
show interfaces <interface> vif <vlan-id>	Displays information about a virtual interface.

Commands for using other system features with VLAN interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Bridging	Layer 2 bridging is supported on Ethernet and bonding vifs. Commands for configuring bridge groups are described in the *Vyatta Bridging Reference Guide*.
Firewall	Firewall is supported on VLAN interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.

OSPF	OSPF is supported on VLAN interfaces. Commands for configuring OSPF are described in the *Vyatta OSPF Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on VLAN interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
QoS	Quality of service traffic policies are supported on VLAN interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.
RIP	RIP is supported on VLAN interfaces. Commands for configuring RIP are described in the *Vyatta RIP Reference Guide*.
RIPng	RIPng is supported on VLAN interfaces. Commands for configuring RIPng are described in the *Vyatta RIPng Reference Guide*.
VRRP	VRRP is supported on Ethernet and bonding vifs. Commands for configuring VRRP are described in the *Vyatta High Availability Reference Guide*.

interfaces <interface> vif <vlan-id>

Defines a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id*

delete interfaces *interface* **vif** [*vlan-id*]

show interfaces *interface* **vif** [*vlan-id*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	Multi-node. The VLAN ID for the vif, for use with 802.1Q VLAN tagging. The range is 0 to 4094. Note that only 802.1Q-tagged packets are accepted on Ethernet vifs. You can define more than one vif for an interface by creating multiple **vif** configuration nodes.

Default

None.

Usage Guidelines

Use this command to create a virtual interface. The vifs function as VLAN interfaces, and only 802.1Q tagged packets are accepted.

NOTE *The interface or bonding group must be defined before a virtual interface can be added. For Ethernet bonded links, group members must also be assigned.*

Use the **set** form of this command to define a virtual interface.

Use the **delete** form of this command to remove the virtual interface and all its configuration.

Use the **show** form of this command to view virtual interface configuration.

interfaces <interface> vif <vlan-id> address

Specifies an IP address and network prefix for a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

delete interfaces *interface* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp** | **dhcpv6**}

show interfaces *interface* **vif** *vlan-id* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
      address [ipv4|ipv6|dhcp|dhcpv6]
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	Multi-node. The VLAN ID for the vif. The range is 0 to 4094.
ipv4	The IPv4 address and network prefix for this vif. The format is *ip-address/prefix* (for example, 192.168.1.77/24). You can define multiple IP addresses for a vif by creating multiple **address** configuration nodes.

ipv6	The IPv6 address and network prefix for this vif. The format is *ipv6-address/prefix* (for example, 2001:db8:1234::/48). You can define multiple IPv6 addresses for a vif by creating multiple **address** configuration nodes.
dhcp	Defines the interface as a Dynamic Host Configuration Protocol (DHCP) client, which obtains its address and prefix from a DHCP server.
dhcpv6	Defines the interface as a Dynamic Host Configuration Protocol (DHCP) for IPv6 client, which obtains its address and prefix from a DHCPv6 server.

Default

None.

Usage Guidelines

Use this command to assign an IP address to a virtual interface.

Use the **set** form of this command to specify an address for this virtual interface.

Use the **delete** form of this command to remove the address for this virtual interface.

Use the **show** form of this command to view the address for this virtual interface.

interfaces <interface> vif <vlan-id> description <descr>

Sets a description for a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **description** *descr*

delete interfaces *interface* **vif** *vlan-id* **description**

show interfaces *interface* **vif** *vlan-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
      description descr
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
descr	The description for the vif.

Default

None.

Usage Guidelines

Use this command to set a description for virtual interface.

Use the **set** form of this command to set a description for a virtual interface

Use the **delete** form of this command to remove the description for a virtual interface.

Use the **show** form of this command to view virtual interface description configuration.

interfaces <interface> vif <vlan-id> dhcpv6-options

Specifies the way in which a DHCPv6 client is to acquire an address and/or parameters from a DHCPv6 server.

Syntax

set interfaces *interface* **vif** *vlan-id* **dhcpv6-options** [**parameters-only** | **temporary**]

delete interfaces *interface* **vif** *vlan-id* **dhcpv6-options** [**parameters-only** | **temporary**]

show interfaces *interface* **vif** *vlan-id* **dhcpv6-options**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    vif vlan-id {
        dhcpv6-options [parameters-only|temporary]
    }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
parameters-only	Acquires only configuration parameters (and not an IPv6 address) from the DHCPv6 server. Only one of the **parameters-only** and the **temporary** parameter may be specified.

temporary	Acquires a temporary IPv6 address as described for IPv6 privacy addressing in RFC 4941. Only one of the **parameters-only** and the **temporary** parameter may be specified.

Default

None.

Usage Guidelines

Use this command to specify in what way the DHCPv6 client is to acquire an IPv6 address and/or parameters from a DHCPv6 server.

Note that these parameters are only relevant if the **dhcpv6** option has been set for the interfaces <interface> vif <vlan-id> address command.

The **parameters-only** option is typically used in conjunction with SLAAC or static address configuration. It and the **temporary** parameter are mutually exclusive.

Use the **set** form of this command to specify the DHCPv6 options.

Use the **delete** form of this command to remove the DHCPv6 options.

Use the **show** form of this command to view DHCPv6 option configuration.

interfaces <interface> vif <vlan-id> disable

Disables a virtual interface without discarding configuration.

Syntax

set interfaces *interface* **vif** *vlan-id* **disable**

delete interfaces *interface* **vif** *vlan-id* **disable**

show interfaces *interface* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
      disable
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.

Default

The vif is enabled.

Usage Guidelines

Use this command to disable a virtual interface without discarding configuration.

Use the **set** form of this command to disable the interface.

Use the **delete** form of this command to enable the interface.

Use the **show** form of this command to view virtual interface configuration.

interfaces <interface> vif <vlan-id> disable-link-detect

Directs a virtual interface not to detect physical link-state changes.

Syntax

set interfaces *interface* **vif** *vlan-id* **disable-link-detect**

delete interfaces *interface* **vif** *vlan-id* **disable-link-detect**

show interfaces *interface* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    vif vlan-id {
        disable-link-detect
    }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.

Default

By default **disable-link-detect** is not set.

Usage Guidelines

Use this command to direct a virtual interface to not detect physical state change to the underlying Ethernet link (for example, when the cable is unplugged).

Use the **set** form of this command to disable detection of physical state changes.

Use the **delete** form of this command to enable detection of physical state changes.

Use the **show** form of this command to view virtual interface configuration.

interfaces <interface> vif <vlan-id> ipv6 address

Assigns an IPv6 address to a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

delete interfaces *interface* **vif** *vlan-id* **ipv6 address** [**autoconf** | **eui64** *ipv6prefix*]

show interfaces *interface* **vif** *vlan-id* **ipv6 address** [**autoconf** | **eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
      ipv6 {
         address {
            autoconf
            eui64 ipv6prefix
         }
      }
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**. This command is not available for vifs on pseudo-Ethernet interfaces.

autoconf	Generates an IPv6 address using the SLAAC protocol. Set this value if the interface is performing a "host" function rather than a "router" function. This value can be specified in addition to specifying static IPv6, static IPv4, or IPv4 DHCP addresses on the interface.
ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address, in EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value derived from the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to assign an IPv6 address to a virtual interface.

You can using the **autoconf** keyword to direct the system to autoconfigure the address, using the SLAAC (Stateless Address Auto-Configuration) protocol defined in RFC 4862. Alternatively, you can provide an EUI-64 IPv6 address prefix so that the system constructs the IPv6 address.

If you want the system to use SLAAC to acquire addresses on this interface, then in addition setting this parameter, you must also disable IPv6 forwarding, either globally (using the system ipv6 disable-forwarding command) or specifically on this interface (using the interfaces <interface> vif <vlan-id> ipv6 disable-forwarding command).

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view IPv6 address configuration settings.

interfaces <interface> vif <vlan-id> ipv6 disable-forwarding

Disables IPv6 forwarding on a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **ipv6 disable-forwarding**

delete interfaces *interface* **vif** *vlan-id* **ipv6 disable-forwarding**

show interfaces *interface* **vif** *vlan-id* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    vif vlan-id {
        ipv6 {
            disable-forwarding
        }
    }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**. This command is not available for vifs on pseudo-Ethernet interfaces.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on an Ethernet interface.

You can also disable IPv6 forwarding globally (that is, for all interfaces) using the system ipv6 disable-forwarding command.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

Use the **show** form of this command to display IPv6 packet forwarding interface configuration.

interfaces <interface> vif <vlan-id> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces *interface* **vif** *vlan-id* **ipv6 dup-addr-detect-transmits** *num*

delete interfaces *interface* **vif** *vlan-id* **ipv6 dup-addr-detect-transmits**

show interfaces *interface* **vif** *vlan-id* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    vif vlan-id {
        ipv6 {
            dup-addr-detect-transmits num
        }
    }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**. This command is not available for vifs on pseudo-Ethernet interfaces.
num	The number of times to transmit NS packets as part of the DAD process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view NS packet configuration for DAD.

interfaces <interface> vif <vlan-id> ipv6 router-advert

Specifies the router advertisements to be sent from a virtual interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **ipv6 router-advert** [**cur-hop-limit** *limit*] [**default-lifetime** *lifetime*] [**default-preference** *preference*] [**link-mtu** *mtu*] [**managed-flag** *state*] [**max-interval** *interval*] [**min-interval** *interval*] [**other-config-flag** *state*] [**prefix** *ipv6net* [**autonomous-flag** *state* | **on-link-flag** *state* | **preferred-lifetime** *lifetime* | **valid-lifetime** *lifetime*]] [**reachable-time** *time*] [**retrans-timer** *time*] [**send-advert** *state*]

delete interfaces *interface* **vif** *vlan-id* **ipv6 router-advert** [**cur-hop-limit**] [**default-lifetime**] [**default-preference**] [**link-mtu**] [**managed-flag**] [**max-interval**] [**min-interval**] [**other-config-flag**] [**prefix** *ipv6net* [**autonomous-flag** | **on-link-flag** | **preferred-lifetime** | **valid-lifetime**]] [**reachable-time**] [**retrans-timer** [**send-advert**]

show interfaces *interface* **vif** *vlan-id* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
    vif vlan-id {
        ipv6 {
            router-advert {
                cur-hop-limit limit
                default-lifetime lifetime
                default-preference preference
                link-mtu mtu
                managed-flag state
                max-interval interval
                min-interval interval
                other-config-flag state
                prefix ipv6net {
                    autonomous-flag state
                    on-link-flag state
                    preferred-lifetime lifetime
                    valid-lifetime lifetime
                }
                reachable-time time
                retrans-timer time
```

```
                send-advert state
            }
        }
    }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**. This command is not available for vifs on pseudo-Ethernet interfaces.
cur-hop-limit *limit*	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets. This value is placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range is 0 to 255. The default is 64. A value of 0 means unspecified by the router.
default-lifetime lifetime	Specifies the lifetime , in seconds, associated with the default router. Supported values are 0, which indicates that the router is not a default router, and the range from the value configured for the **max-interval** option to 9000 (18.2 hours). If not configured, the value for this timer is three times **max-interval**.
default-preference *preference*	The preference associated with the default router. Supported values are as follows: **low**: The default router is low preference. **medium**: The default router is medium preference. **high**: The default router is high preference. The default is **medium**.

link-mtu *mtu*	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the type of link, as defined in RFC 2464. The default is 0, which means the MTU is not specified in the router advertisement message. That is because it is expected that the MTU will configured directly on the interface itself and not for routing advertisements. You can configure this option in cases where the link MTU is not well known. If the value set here does not match the MTU configured on the interface, the system issues a warning but does not fail.
managed-flag *state*	Whether to use the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration. The default is **false**.
max-interval *interval*	The maximum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default is 600 (10 minutes).
min-interval *interval*	The minimum time, in seconds, allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * **max-interva**l. The default is 0.33 * **max-interval**.
other-config-flag *state*	The interfaces uses the administered (stateful) protocol for autoconfiguration of non-address information, as defined in RFC 4862. Supported values are as follows: **true**: Hosts use the administered protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default is **false**.
prefix *ipv6net*	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface, in the format *ipv6-address/prefix*. You can define more than one IPv6 prefix by configuring multiple **prefix** configuration nodes.

autonomous-flag *state*	Specifies whether the prefix can be used for autonomous address configuration as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default is **true**.
on-link-flag *state*	Specifies whether the prefix can be used for on-link determination, as defined in RFC 4862. Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default is **true**.
preferred-lifetime *lifetime*	The length of time, in seconds, that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) is to remain preferred, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 604800 (seven days).
valid-lifetime *lifetime*	The length of time, in seconds, that the prefix is valid for the purpose of on-link determination, as defined in RFC 4862. The interval is with respect to the time the packet is sent. The range is 1 to 4294967296 plus the keyword **infinity**, which represents forever. (The actual value of **infinity** is a byte where all bits are set to ones: 0XFFFFFFFF.) The default is 2592000 (30 days).
reachable-time *time*	The length of time, in milliseconds, for which the system assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range is 0 to 3600000, where a value of 0 means the reachable time is not specified in the router advertisement message. The default is 0.

retrans-timer *time*	The length of time, in milliseconds, between retransmitted NS messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295, where a value of 0 means the retransmit tme is not specified in the router advertisement message. The default is 0.
send-advert *state*	Specifies whether router advertisements are to be sent from this interface. Supported values are as follows: **true**: Sends router advertisements from this interface. **false**: Does not send router advertisements from this interface. If this value is in effect, parameters in this configuration subtree are still used to configure the local implementation parameters. The default is **true**.

Default

Router advertisements are not sent on an interface.

Usage Guidelines

Use this command to configure router advertisements (RAs) to be sent out of the interface being configured.

Router advertisements are sent out by IPv6 routers in order to advertise their existence to hosts on the network. IPv6 hosts do not send out router advertisements.

If the **router-advert** node of the configuration tree is missing, router advertisements are not sent out. Also, if IPv6 forwarding is disabled either globally (using the **system ipv6 disable-forwarding** command) or on the interface (using the interfaces <interface> vif <vlan-id> ipv6 disable-forwarding command), router advertisements are not sent out.

Most router advertisement parameters are required by either the Neighbor Discovery (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view router advertisement configuration.

interfaces <interface> vif <vlan-id> redirect <int>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces *interface* **vif** *vlan-id* **redirect** *int*

delete interfaces *interface* **vif** *vlan-id* **redirect** *int*

show interfaces *interface* **vif** *vlan-id* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces interface {
   vif vlan-id {
      redirect int
   }
}
```

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	Multi-node. The VLAN ID for the vif, for use with 802.1Q VLAN tagging. The range is 0 to 4094. Note that only 802.1Q-tagged packets are accepted on Ethernet vifs. You can define more than one vif for an interface by creating multiple **vif** configuration nodes.
int	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 7: Input Interface.)

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

show interfaces <interface> vif <vlan-id>

Displays information about a virtual interface.

Syntax

show interfaces *interface* **vif** *vlan-id* [**brief | queue** [**class | filter**]]

Command Mode

Operational mode.

Parameters

interface	The interface type and interface identifier. Supported values are as follows: **bonding** *bondx*: An Ethernet bonded link interface. The range for identifiers is **bond0** through **bond99**. **ethernet** *ethx*: A physical Ethernet interface. The range for identifiers is **eth0** through **eth23**. **psuedo-ethernet** *pethx*: A pseudo-Ethernet interface. The range for identifiers is **peth0** through **peth***x*, where x is a positive integer. **wireless** *wlanx*: A wireless interface. The range for identifiers is **wlanx0** through **wlanx999**.
vlan-id	Displays information for the specified virtual interface.
brief	Displays a brief status for the virtual interface.
queue	Displays queue information for the virtual interface.
class	Display queue classes for the virtual interface.
filter	Display queue filters for the virtual interface.

Default

None.

Usage Guidelines

Use this command to view command and operational status of a virtual interface.

Examples

Example 6-2 shows information for vif 9 on interface bond0

Example 6-2 Displaying virtual interface information

```
vyatta@vyatta:~$ show interfaces bonding bond0 vif 9
bond0.9@bond0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    link/ether 00:0c:29:da:3a:3d brd ff:ff:ff:ff:ff:ff
    inet6 fe80::20c:29ff:feda:3a3d/64 scope link
       valid_lft forever preferred_lft forever

    RX:  bytes    packets     errors    dropped    overrun      mcast
               0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
            2914         13          0          0          0          0
vyatta@vyatta:~$
```

Example 6-3 shows brief status for vif 40 on interface eth1.

Example 6-3 Displaying brief status for a virtual interface.

```
vyatta@R1:~$ show interfaces ethernet eth1 vif 40 brief
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth1.40          10.10.40.65/24                 u/u
vyatta@R1:~$
```

Example 6-4 shows queue information for vif 6 on interface peth0.

Example 6-4 Displaying VLAN interface queue information

```
vyatta@vyatta:~$ show interfaces pseudo-ethernet peth0 vif 6 queue
qdisc pfifo_fast 0: root bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 380009 bytes 5177 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
```

Chapter 7: Input Interface

This chapter explains how to work with Input interfaces on the Vyatta system.

This chapter presents the following topics:

Input Interface Overview

An Input interface is a special-purpose interface that is often used in conjunction with interface input redirection. It is typically used in this way to extend QoS functionality. There are two main uses for it:

- Applying a single QoS policy across the combined inbound traffic from multiple interfaces
- Applying outbound QoS policies to inbound traffic

You cannot configure an IP address or link parameters for Input interfaces. In addition, Input interfaces cannot be specified as arguments for routing protocol commands (for example, you cannot enable BGP on an Input interface).

The Vyatta system has extensive support for IPv6, including IPv6 interface addressing. The commands for configuring IPv6 on input interfaces are given in this chapter. A full description of Vyatta IPv6 support is provided in the *Vyatta IPv6 Support Reference Guide*.

Input Interface Configuration Examples

This section presents the following topics:

NOTE *In addition to the examples shown in this section, configuration examples showing Input interfaces used in conjunction with QoS are provided in the Vyatta QoS Reference Guide.*

Creating an Input Interface

The example in this section creates an Input interface.

To create an Input interface, perform the following steps in configuration mode.

Example 7-1 Configuring an Input interface

Step	Command
Create an Input interface.	`vyatta@R1# set interfaces input ifb0`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show interfaces input`
	`ifb0 {`
	`}`

IPv6 on Input Interfaces

Examples for configuring IPv6 on interfaces are provided in the *Vyatta IPv6 Support Reference Guide*.

Input Interface Commands

Configuration Commands	
interfaces input <ifbx>	Defines an input interface.
interfaces input <ifbx> description <descr>	Specifies a description for an input interface.
interfaces input <ifbx> redirect <interface>	Redirects inbound traffic from from one interface to another interface.
Operational Commands	
show interfaces input <ifbx>	Displays information about an input interface.

Commands for using other system features with Input interfaces can be found in the following locations

Related Commands Documented Elsewhere	
Firewall	Firewall is supported on input interfaces. Commands for configuring firewall are described in the *Vyatta Firewall Reference Guide*.
Policy Based Routing	Policy Based Routing is supported on input interfaces. Commands for configuring Policy Based Routing are described in the *Vyatta Policy Based Routing Reference Guide*.
QoS	Quality of service traffic policies are supported on input interfaces. Commands for configuring QoS are described in the *Vyatta QoS Reference Guide*.

interfaces input <ifbx>

Defines an input interface.

Syntax

set interfaces input *ifbx*

delete interfaces input *ifbx*

show interfaces input *ifbx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    input ifbx {
    }
}
```

Parameters

ifbx	Multi-node. The identifier for the Input interface. The range is **ifb0** to **ifb999**. You can define multiple input interfaces by creating more than one **input** configuration node.

Default

None.

Usage Guidelines

Use this command to configure an input interface. You can define multiple input interfaces by creating multiple **input** configuration nodes.

Note that you cannot use **set** to change the name of the input interface. To change the name of an input interface, you must delete the old **input** configuration node and create a new one.

Use the **set** form of this command to create an input interface.

Use the **delete** form of this command to remove all configuration for an input interface.

Use the **show** form of this command to view an input interface configuration.

interfaces input <ifbx> description <descr>

Specifies a description for an input interface.

Syntax

set interfaces input *ifbx* **description** *descr*

delete interfaces input *ifbx* **description**

show interfaces input *ifbx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    input ifbx {
        description descr
    }
}
```

Parameters

ifbx	The identifier for the Input interface. The range is **ifb0** to **ifb999**.
descr	A mnemonic name or description for the input interface.

Default

None.

Usage Guidelines

Use this command to set a description for an Input interface.

Use the **set** form of this command to specify the description.

Use the **delete** form of this command to remove the description.

Use the **show** form of this command to view description configuration.

interfaces input <ifbx> redirect <interface>

Redirects inbound traffic from from one interface to another interface.

Syntax

set interfaces input *ifbx* **redirect** *interface*

delete interfaces input *ifbx* **redirect** *interface*

show interfaces input *ifbx* **redirect**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    input ifbx {
        redirect interface
    }
}
```

Parameters

ifbx	The identifier for the Input interface. The range is **ifb0** to **ifb999**.
interface	The identifier of the interface to which you are redirecting data; for example, **ifb0**.

Default

None.

Usage Guidelines

Use this command to redirect inbound traffic from an interface to another interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface.

Redirecting traffic from several interfaces to a single Input interface allows you to apply a single QoS policy to the combined traffic—for example, to limit the combined inbound traffic bandwidth.

Use the **set** form of this command to specify the redirect interface.

Use the **delete** form of this command to remove the redirect configuration.

Use the **show** form of this command to view the redirect configuration.

show interfaces input <ifbx>

Displays information about an input interface.

Syntax

show interfaces input *ifbx* [**brief | queue** [**class | filter**]]

Command Mode

Operational mode.

Parameters

ifbx	Displays information for only the specified interface. The range is **ifb0** to **ifb999**.
brief	Displays a brief status for the Input interface. This option is equivalent to using the command with no option.
queue	Displays queue information for the Input interface.
class	Display queue classes for the Input interface.
filter	Display queue filters for the Input interface.

Default

When used with no option, this command displays brief status information for all defined Input interfaces.

Usage Guidelines

Use this command to view command and operational status of an Input interface.

Examples

Example 7-2 shows information for an Input interface.

Example 7-2 Displaying Input interface information

```
vyatta@R1:~$ show interfaces input
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
```

```
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
ifb0             -                              u/u  Input interface for
                                                     consolidating QoS
                                                     mechanisms
```

Example 7-3 shows queue information for an Input interface.

Example 7-3 Displaying queue information for an Input interface

```
vyatta@R1:~$ show interfaces input ifb0 queue
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
```

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol

DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System

IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol

NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point

RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network

VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access