VYATTA, INC. | Vyatta System

LAN Interfaces

REFERENCE GUIDE

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

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: February 2011

DOCUMENT REVISION. R6.2 v01

RELEASED WITH: R6.2.0

PART NO. A0-0211-10-0010

Table of Contents

Quick Reference to Commands

Use this section to help you quickly locate a command.

Quick List of Examples

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

Preface

This document describes the various deployment, installation, and upgrade options for Vyatta software.

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 Reference to Commands

  Use this list to help you quickly locate commands.

- Quick 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: Ethernet Interfaces	This chapter describes basic configuration for Ethernet interfaces.	1
Chapter 2: Loopback Interface	This chapter explains how to work with the Vyatta system's software loopback interface.	45
Chapter 3: VLAN Interfaces	This chapter lists the commands for configuring VLAN interfaces on Ethernet interfaces and Ethernet bonded links.	57
Chapter 4: Bridging	This chapter lists the commands used for Spanning Tree Protocol and bridging.	102
Chapter 5: Ethernet Link Bonding	This chapter explains how to bond Ethernet links into a larger virtual link.	184

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: Ethernet Interfaces

This chapter describes basic configuration for Ethernet interfaces.

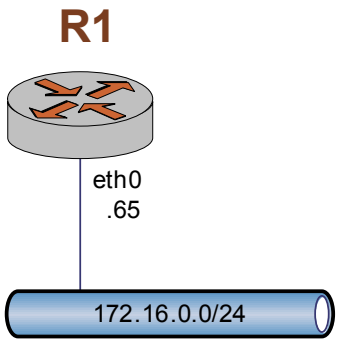
This chapter presents the following topics:

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 1-1.

Figure 1-1 Basic Ethernet configuration



In this example, you create an Ethernet interface and assign an IP address to it.

Example 1-1 creates the Ethernet interface and assigns an IP address to it. To do this, perform the following steps on R1 in configuration mode

Example 1-1 Configuring an Ethernet interface

Step	Command
Create the Etherent interface and assign an address to it.	`vyatta@R1# set interfaces ethernet eth0 address 176.16.0.65/24`
Commit the configuration.	`vyatta@R1# commit`

View the configuration.	`vyatta@R1# show interfaces`
	`ethernet eth0 {`
	`address 176.16.0.65/24`
	`}`

Ethernet Interface Commands

This chapter contains the following commands.

Configuration Commands	
interfaces ethernet <ethx>	Defines an Ethernet interface.
interfaces ethernet <ethx> address	Sets an IP address and network prefix for an Ethernet interface.
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-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> 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>	Specifies the MTU for an Ethernet interface.
interfaces ethernet <ethx> redirect <interface>	Redirects inbound traffic from an Ethernet interface to another interface.
interfaces ethernet <ethx> smp_affinity	Sets the SMP 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.
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> capture	Displays (captures) traffic on 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	Commands for configuring Ethernet interfaces within bridge groups are described in "Chapter 4: Bridging."
Link Bonding	Commands for configuring Ethernet bonded links are described in "Chapter 5: Ethernet Link Bonding."
Firewall	Commands for configuring firewall on Ethernet interfaces are described in the *Vyatta Firewall Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Ethernet interfaces are described in the *Vyatta OSPF Reference Guide*.
PPPoE encapsulation	Commands for configuring Point-to-Point Protocol over Ethernet encapsulation on Ethernet interfaces are described in the *Vyatta PPP-Based Encapsulations Reference Guide.*
QoS	Commands for configuring quality of service on Ethernet interfaces are described in the *Vyatta QoS Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on Ethernet interfaces are described in the *Vyatta RIP 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	Commands for configuring vifs on Ethernet interfaces (VLAN interfaces) are described in "Chapter 3: VLAN Interfaces."
VRRP	Commands for configuring Virtual Router Redundancy Protocol on Ethernet interfaces 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> 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-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> 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 inbound traffic on one interface to a system running a monitoring or IDS 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>

Specifies 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 1 to 1500.

Default

If this value is not set, fragmentation is never performed.

Usage Guidelines

Use this command to set the maximum transmission unit (MTU) for an Ethernet interface. This value is also applied to all vifs defined for the interface.

Note that the MTU of an Ethernet interface that is part of an Ethernet link bonding interface is not allowed to be changed.

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 specify the MTU.

Use the **delete** form of this command to remove MTU value and disable fragmentation.

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

interfaces ethernet <ethx> redirect <interface>

Redirects inbound traffic from an Ethernet 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 the 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 the inbound traffic from an Ethernet interface to an Input interface.

This feature is typically used to redirect traffic from a number of interfaces to an Input interface. (Input interfaces are described in Chapter 8: Input Interfaces.) Redirecting traffic from several interfaces to a single 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 SMP affinity for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **smp_affinity** {**auto** | *mask*}

delete interfaces ethernet *ethx* **smp_affinity** {**auto** | *mask*}

show interfaces ethernet *ethx* **smp_affinity**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        smp_affinity [auto|mask]
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
auto	Automatically configure optimal SMP affinity.
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.

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 mask 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.

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 will be displayed 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.

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 1-2 shows information for all Ethernet interfaces.

Example 1-2 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 1-3 shows information for interface eth2.

Example 1-3 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 1-4 shows the first screen of output for **show interfaces ethernet detail**.

Example 1-4 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 1-5 shows brief status for interface eth2.

Example 1-5 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> capture

Displays (captures) traffic on an Ethernet interface.

Syntax

show interfaces ethernet *ethx* **capture** [**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 1-6 shows captured data on interface eth0.

Example 1-6 Displaying captured data

```
vyatta@vyatta:~$ show interfaces ethernet eth0 capture
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 <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 1-7 shows the output for **show interfaces ethernet** *ethx* **identify**.

Example 1-7 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 1-8 shows output for **show interfaces ethernet** *ethx* **physical**.

Example 1-8 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 1-9 shows queue information for interface eth0.

Example 1-9 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 1-10 shows Ethernet statistics information for interface eth3.

Example 1-10 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 2: Loopback Interface

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

This chapter presents the following topics:

Loopback Commands

This chapter contains the following 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>	Specifies a description for the loopback interface.
Operational Commands	
clear interfaces loopback counters	Clears statistics counters for loopback interfaces.
show interfaces loopback	Displays information about the loopback interface.
show interfaces loopback detail	Displays detailed information and statistics about the loopback interface.
show interfaces loopback lo brief	Displays brief status information for the loopback interface.

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

Related Commands Documented Elsewhere	
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Ethernet interfaces are described in the *Vyatta OSPF Reference Guide*.
QoS	Commands for configuring quality of service on Ethernet interfaces are described in the *Vyatta QoS Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on Ethernet interfaces are described in the *Vyatta RIP 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 lo interface only.

Default

Clears counters for all loopback interfaces.

Usage Guidelines

Use this command to clear counters on loopback interfaces.

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

The loopback interface is a special software-only interface that emulates a physical interface and allows the system to “connect” to itself. Packets routed to the loopback interface are rerouted back to the system 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 system is functioning, the loopback interface is always up, and so is very reliable. As long as there is even one functioning link to the system, the loopback interface can be accessed. The loopback interface thus eliminates the need to try each IP address of the system 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 as 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, since the the routing traffic is more likely to be received and subsequently forwarded.
- In BGP, parallel paths can be configured to the loopback interface on a peer device. This provides improved load sharing.

Use this command to define 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 Ethernet 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, 2001:db8:1234::/48). You can define multiple IPv6 addresses for a single interface, by creating multiple **address** configuration nodes.

Default

None.

Usage Guidelines

The system automatically creates the loopback interface on startup, with an interface name of **lo**. You must configure an IP address for the interface. The IP address for the loopback interface must be unique, and must not be used by any other interface.

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

- The system's hostname should be mapped to the loopback interface address, rather than a physical interface.
- In OSPF and iBGP configurations, the router ID should be set to the loopback address.

The network for the loopback interface can be small, since IP address space is not a consideration in this case. Often a network prefix of /32 is assigned.

NOTE: By default, the RIB contains connected routes for **lo**: 127.0.0.1/8 (IPv4), and ::1/128 (IPv6).

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>

Specifies 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.

show interfaces loopback

Displays information about the loopback interface.

Syntax

show interfaces loopback [lo]

Command Mode

Operational mode.

Parameters

lo	Displays detailed statistics and configuration information for the loopback interface.

Default

Displays brief status information for the loopback interface.

Usage Guidelines

Use this command to view status of the loopback interface.

Examples

Example 2-1 shows information for the loopback interface.

Example 2-1 Displaying loopback interface information.

```
vyatta@vyatta:~$ show interfaces loopback
Interface    IP Address         State       Link    Description
lo           127.0.0.1/8        up          up
```

Example 2-2 shows detailed information for the loopback interface.

Example 2-2 Displaying detailed loopback interface information.

```
vyatta@vyatta:~$ show interfaces loopback lo
```

```
lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    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
             0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
             0          0          0          0          0          0
```

show interfaces loopback detail

Displays detailed information and statistics about the loopback interface.

Syntax

show interfaces loopback detail

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

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

Examples

Example 2-3 shows detailed statistics for the loopback interface.

Example 2-3 Displaying loopback interface statistics

```
vyatta@vyatta:~$ show interfaces loopback detail
lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    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
             0          0          0          0          0          0
    TX:  bytes    packets     errors    dropped    carrier collisions
             0          0          0          0          0          0
```

show interfaces loopback lo brief

Displays brief status information for the loopback interface.

Syntax

show interfaces loopback lo brief

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

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

Examples

Example 2-4 shows brief status information for the loopback interface.

Example 2-4 Displaying loopback interface status.

```
vyatta@vyatta:~$ show interfaces loopback lo brief
Interface    IP Address         State       Link   Description
lo           127.0.0.1/8        up          up
```

Chapter 3: 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 Commands

This chapter contains the following commands.

Configuration Commands	
Vifs on Ethernet Interfaces	
interfaces ethernet <ethx> vif <vlan-id>	Defines a virtual interface on an Ethernet interface.
interfaces ethernet <ethx> vif <vlan-id> address	Specifies an IP address and network prefix for an Ethernet virtual interface.
interfaces ethernet <ethx> vif <vlan-id> description <descr>	Sets a description for a vif on an Ethernet interface.
interfaces ethernet <ethx> vif <vlan-id> disable	Disables a virtual interface without discarding configuration.
interfaces ethernet <ethx> vif <vlan-id> disable-link-detect	Directs an Ethernet vif not to detect physical link-state changes.
interfaces ethernet <ethx> vif <vlan-id> ip enable-proxy-arp	Enables proxy ARP on an Ethernet vif interface.
Vifs on Ethernet Link Bonding Interfaces	
interfaces bonding <bondx> vif <vlan-id>	Defines a virtual interface on an Ethernet link bonding interface.
interfaces bonding <bondx> vif <vlan-id> address	Specifies an IP address and network prefix for an Ethernet link bonding virtual interface.
interfaces bonding <bondx> vif <vlan-id> description <descr>	Sets a description for a vif on an Ethernet link bonding interface.
interfaces bonding <bondx> vif <vlan-id> disable	Disables a virtual interface without discarding configuration.
interfaces bonding <bondx> vif <vlan-id> disable-link-detect	Directs an Ethernet link bonding vif not to detect physical link-state changes.
Vifs on Wireless Interfaces	
interfaces wireless <wlanx> vif <vlan-id>	Defines a virtual interface on a wireless interface.
interfaces wireless <wlanx> vif <vlan-id> address	Specifies an IP address and network prefix for a wireless virtual interface.
interfaces wireless <wlanx> vif <vlan-id> description <descr>	Sets a description for a vif on a wireless interface.

interfaces wireless <wlanx> vif <vlan-id> disable	Disables a virtual interface without discarding configuration.
interfaces wireless <wlanx> vif <vlan-id> disable-link-detect	Directs a wireless vif not to detect physical link-state changes.
Operational Commands	
show interfaces bonding <bondx> vif <vlan-id>	Displays information about an Ethernet link bonding vif.
show interfaces bonding <bondx> vif <vlan-id> brief	Displays a brief status for an Ethernet link bonding vif.
show interfaces bonding <bondx> vif <vlan-id> queue	Displays vif queuing information.
show interfaces ethernet <ethx> vif <vlan-id>	Displays information about an Ethernet vif.
show interfaces ethernet <ethx> vif <vlan-id> brief	Displays a brief status for an Ethernet vif.
show interfaces ethernet <ethx> vif <vlan-id> queue	Displays vif queuing information.

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

Related Commands Documented Elsewhere	
clear interfaces ethernet counters	Clears statistics counters for Ethernet interfaces. See page 5.
show interfaces ethernet detail	Displays detailed information about Ethernet interfaces. See page 35.
Bridging	Commands for adding VLAN interfaces to bridge groups are described in “Chapter 4: Bridging.”
Firewall	Commands for configuring firewall on VLAN interfaces are described in the *Vyatta Firewall Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on VLAN interfaces are described in the *Vyatta OSPF Reference Guide*.
PPPoE encapsulation	Commands for configuring Point-to-Point Protocol over Ethernet encapsulation on VLAN interfaces are described in *Vyatta PPP-Based Encapsulations Reference Guide.*
RIP	Commands for configuring the Routing Information Protocol on VLAN interfaces are described in the *Vyatta RIP Reference Guide*.

QoS	Commands for configuring qulaity of service on VLAN interfaces are described in the *Vyatta QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the *Vyatta Basic System Reference Guide*.
VRRP	Commands for configuring Virtual Router Redundancy Protocol on VLAN interfaces are described in the *Vyatta High Availability Reference Guide*.

interfaces bonding <bondx> vif <vlan-id>

Defines a virtual interface on an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id*

delete interfaces bonding *bondx* **vif** [*vlan-id*]

show interfaces bonding *bondx* **vif** [*vlan-id*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
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 (vif) on an Ethernet link bonding interface.

On Ethernet link bonding interfaces, vifs function as Virtual LAN (VLAN) interfaces, and only 802.1Q tagged packets are accepted.

Use the **set** form of this command to define a vif.

Use the **delete** form of this command to remove vif and all its configuration.

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

interfaces bonding <bondx> vif <vlan-id> address

Specifies an IP address and network prefix for an Ethernet link bonding virtual interface.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

delete interfaces bonding *bondx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

show interfaces bonding *bondx* **vif** *vlan-id* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            address [ipv4|ipv6|dhcp]
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
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.

Default

None.

Usage Guidelines

Use this command to assign an IP address to a virtual interface (vif, or VLAN interface).

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

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

Use the **show** form of this command to view the address for this **vif**.

interfaces bonding <bondx> vif <vlan-id> description <descr>

Sets a description for a vif on an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **description** *descr*

delete interfaces bonding *bondx* **vif** *vlan-id* **description**

show interfaces bonding *bondx* **vif** *vlan-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            description descr
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
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 a vif on an Ethernet link bonding interface.

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

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

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

interfaces bonding <bondx> vif <vlan-id> disable

Disables a virtual interface without discarding configuration.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **disable**

delete interfaces bonding *bondx* **vif** *vlan-id* **disable**

show interfaces bonding *bondx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            disable
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
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 vif on an Ethernet link bonding 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 vif configuration.

interfaces bonding <bondx> vif <vlan-id> disable-link-detect

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

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **disable-link-detect**

delete interfaces bonding *bondx* **vif** *vlan-id* **disable-link-detect**

show interfaces bonding *bondx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            disable-link-detect
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
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 an Ethernet link bonding vif 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 Ethernet link bonding vif configuration.

interfaces ethernet <ethx> vif <vlan-id>

Defines a virtual interface on an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id*

delete interfaces ethernet *ethx* **vif** [*vlan-id*]

show interfaces ethernet *ethx* **vif** [*vlan-id*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
        }
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
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 (vif) on an Ethernet interface.

On Ethernet interfaces, vifs function as Virtual LAN (VLAN) interfaces, and only 802.1Q tagged packets are accepted.

Use the **set** form of this command to define a vif.

Use the **delete** form of this command to remove an Ethernet vif and all its configuration.

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

interfaces ethernet <ethx> vif <vlan-id> address

Specifies an IP address and network prefix for an Ethernet virtual interface.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

delete interfaces ethernet *ethx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

show interfaces ethernet *ethx* **vif** *vlan-id* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            address [ipv4|ipv6|dhcp]
        }
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
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.

Default

None.

Usage Guidelines

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

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

Use the **show** form of this command to view the address for this **vif**.

interfaces ethernet <ethx> vif <vlan-id> description <descr>

Sets a description for a vif on an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **description** *descr*

delete interfaces ethernet *ethx* **vif** *vlan-id* **description**

show interfaces ethernet *ethx* **vif** *vlan-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            description descr
        }
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
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 a vif on an Ethernet interface.

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

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

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

interfaces ethernet <ethx> vif <vlan-id> disable

Disables a virtual interface without discarding configuration.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **disable**

delete interfaces ethernet *ethx* **vif** *vlan-id* **disable**

show interfaces ethernet *ethx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            disable
        }
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
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 vif on 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 vif configuration.

interfaces ethernet <ethx> vif <vlan-id> disable-link-detect

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

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **disable-link-detect**

delete interfaces ethernet *ethx* **vif** *vlan-id* **disable-link-detect**

show interfaces ethernet *ethx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            disable-link-detect
        }
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
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 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> vif <vlan-id> ip enable-proxy-arp

Enables proxy ARP on an Ethernet vif interface.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **ip enable-proxy-arp**

delete interfaces ethernet *ethx* **vif** *vlan-id* **ip enable-proxy-arp**

show interfaces ethernet *ethx* **vif** *vlan-id* **ip**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            ip {
                enable-proxy-arp
            }
        }
    }
}
```

Parameters

ethx	The identifier of an Ethernet interface. The range is **eth0** to **eth23**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.

Default

Proxy ARP is not enabled on the Ethernet vif interface.

Usage Guidelines

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

Proxy ARP allows an Ethernet vif 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 wireless <wlanx> vif <vlan-id>

Defines a virtual interface on a wireless interface.

Syntax

set interfaces wireless *wlanx* **vif** *vlan-id*

delete interfaces wireless *wlanx* **vif** [*vlan-id*]

show interfaces wireless *wlanx* **vif** [*vlan-id*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        vif vlan-id {
        }
    }
}
```

Parameters

wlanx	Multi-node. An identifier for the wireless interface you are defining. The range is **wlan0** to **wlan999**.
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 wireless 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 (vif) on a wireless interface.

On wireless interfaces, vifs function as Virtual LAN (VLAN) interfaces, and only 802.1Q tagged packets are accepted.

Use the **set** form of this command to define a vif.

Use the **delete** form of this command to remove a wireless vif and all its configuration.

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

interfaces wireless <wlanx> vif <vlan-id> address

Specifies an IP address and network prefix for a wireless virtual interface.

Syntax

set interfaces wireless *wlanx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

delete interfaces wireless *wlanx* **vif** *vlan-id* **address** {*ipv4* | *ipv6* | **dhcp**}

show interfaces wireless *wlanx* **vif** *vlan-id* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        vif vlan-id {
            address [ipv4|ipv6|dhcp]
        }
    }
}
```

Parameters

wlanx	Multi-node. An identifier for the wireless interface you are defining. The range is **wlan0** to **wlan999**.
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.

Default

None.

Usage Guidelines

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

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

Use the **show** form of this command to view the address for this **vif**.

interfaces wireless <wlanx> vif <vlan-id> description <descr>

Sets a description for a vif on a wireless interface.

Syntax

set interfaces wireless *wlanx* **vif** *vlan-id* **description** *descr*

delete interfaces wireless *wlanx* **vif** *vlan-id* **description**

show interfaces wireless *wlanx* **vif** *vlan-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        vif vlan-id {
            description descr
        }
    }
}
```

Parameters

wlanx	Multi-node. An identifier for the wireless interface you are defining. The range is **wlan0** to **wlan999**.
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 a vif on a wireless interface.

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

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

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

interfaces wireless <wlanx> vif <vlan-id> disable

Disables a virtual interface without discarding configuration.

Syntax

set interfaces wireless *wlanx* **vif** *vlan-id* **disable**

delete interfaces wireless *wlanx* **vif** *vlan-id* **disable**

show interfaces wireless *wlanx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   wireless wlanx {
      vif vlan-id {
         disable
      }
   }
}
```

Parameters

wlanx	Multi-node. An identifier for the wireless interface you are defining. The range is **wlan0** to **wlan999**.
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 vif on a wireless 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 wireless vif configuration.

interfaces wireless <wlanx> vif <vlan-id> disable-link-detect

Directs a wireless vif not to detect physical link-state changes.

Syntax

set interfaces wireless *wlanx* **vif** *vlan-id* **disable-link-detect**

delete interfaces wireless *wlanx* **vif** *vlan-id* **disable-link-detect**

show interfaces wireless *wlanx* **vif** *vlan-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        vif vlan-id {
            disable-link-detect
        }
    }
}
```

Parameters

wlanx	Multi-node. An identifier for the wireless interface you are defining. The range is **wlan0** to **wlan999**.
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 wireless interface to not detect physical state change to the wireless link.

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.

show interfaces bonding <bondx> vif <vlan-id>

Displays information about an Ethernet link bonding vif.

Syntax

show interfaces bonding *bondx* **vif** *vlan-id*

Command Mode

Operational mode.

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	Displays information for the specified vif.

Default

None.

Usage Guidelines

Use this command to view command and operational status of Ethernet link bonding vifs.

Examples

Example 3-1 shows information for vif 9 on interface bond0

Example 3-1 Displaying Ethernet link bonding vif 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:~$
```

show interfaces bonding <bondx> vif <vlan-id> brief

Displays a brief status for an Ethernet link bonding vif.

Syntax

show interfaces bonding *bondx* **vif** *vlan-id* **brief**

Command Mode

Operational mode.

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	Displays information for the specified vif.

Default

None.

Usage Guidelines

Use this command to view the status of a vif.

Examples

Example 3-2 shows brief status for interface bond2.6.

Example 3-2 Displaying brief status for a vif.

```
vyatta@vyatta:~$ show interfaces bonding bond2 vif 6 brief
Interface     IP Address          State       Link   Description
bond2.6       10.2.6.66/24        up          up
```

show interfaces bonding <bondx> vif <vlan-id> queue

Displays vif queuing information.

Syntax

show interfaces bonding *bondx* **vif** *vlan-id* **queue** [**class** | **filter**]

Command Mode

Operational mode.

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	Displays information for the specified vif.
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 vif queue information.

Examples

Example 3-3 shows queue information for interface bond0.6.

Example 3-3 Displaying VLAN interface queue information

```
vyatta@vyatta:~$ show interfaces bonding bond0 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
```

show interfaces ethernet <ethx> vif <vlan-id>

Displays information about an Ethernet vif.

Syntax

show interfaces ethernet *ethx* **vif** *vlan-id*

Command Mode

Operational mode.

Parameters

ethx	The Ethernet interface you are defining. The range is **eth0** to **eth23**.
vlan-id	Displays information for the specified vif.

Default

None.

Usage Guidelines

Use this command to view command and operational status of Ethernet vifs.

Examples

Example 3-4 shows information for vif 11 on interface eth0

Example 3-4 Displaying Ethernet vif information

```
vyatta@vyatta:~$ show interfaces ethernet eth0 vif 11
eth0.11@eth0: <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:~$
```

show interfaces ethernet <ethx> vif <vlan-id> brief

Displays a brief status for an Ethernet vif.

Syntax

show interfaces ethernet *ethx* **vif** *vlan-id* **brief**

Command Mode

Operational mode.

Parameters

ethx	The specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.
vlan-id	Displays information for the specified vif.

Default

None.

Usage Guidelines

Use this command to view the status of a vif.

Examples

Example 3-5 shows brief status for interface eth2.6.

Example 3-5 Displaying brief status for a vif.

```
vyatta@vyatta:~$ show interfaces ethernet eth2 vif 6 brief
Interface     IP Address          State       Link   Description
eth2.6        10.1.6.66/24        up          up
```

show interfaces ethernet <ethx> vif <vlan-id> queue

Displays vif queuing information.

Syntax

show interfaces ethernet *ethx* **vif** *vlan-id* **queue** [**class** | **filter**]

Command Mode

Operational mode.

Parameters

ethx	The specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.
vlan-id	Displays information for the specified vif.
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 vif queue information.

Examples

Example 3-6 shows queue information for interface eth0.6.

Example 3-6 Displaying VLAN interface queue information

```
vyatta@vyatta:~$ show interfaces ethernet eth0 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 4: Bridging

This chapter lists the commands used for Spanning Tree Protocol and bridging.

This chapter presents the following topics:

Bridging Configuration

This section presents the following topics:

Bridging Overview

Bridging allows you to connect multiple network segments (typically LAN segments) at the Layer 2 level.

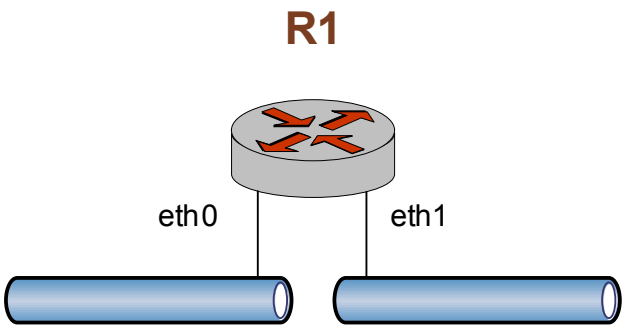
Since bridging occurs at Layer 2 (the data link layer) and IP addresses are relevant only on Layer 3 (the network layer), IP addresses are not required on the interfaces being bridged. IP addresses can be added to the Ethernet interfaces being bridged but have no effect on bridging.

Basic Bridging Configuration

This section presents a sample configuration for a basic bridge between two Ethernet segments on a Vyatta system.

When you have finished, the system will be configured as shown in Figure 4-1.

Figure 4-1 Basic bridging



In this example, you create a bridge interface and assign the Ethernet interfaces to the bridge group.

Example 4-1 creates the bridge interface and adds the Ethernet interfaces to the bridge group. To do this, perform the following steps on R1 in configuration mode

Example 4-1 Configuring a bridge between two Ethernet interfaces

Step	Command
Create the bridge interface.	`vyatta@R1# set interfaces bridge br0`
Add eth0 to the bridge group.	`vyatta@R1# set interfaces ethernet eth0 bridge-group bridge br0`
Add eth1 to the bridge group.	`vyatta@R1# set interfaces ethernet eth1 bridge-group bridge br0`
Commit the configuration.	vyatta@R1# **commit**

View the configuration.	`vyatta@R1# show interfaces`
	`bridge br0 {`
	`}`
	`ethernet eth0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`}`
	`ethernet eth1 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`}`

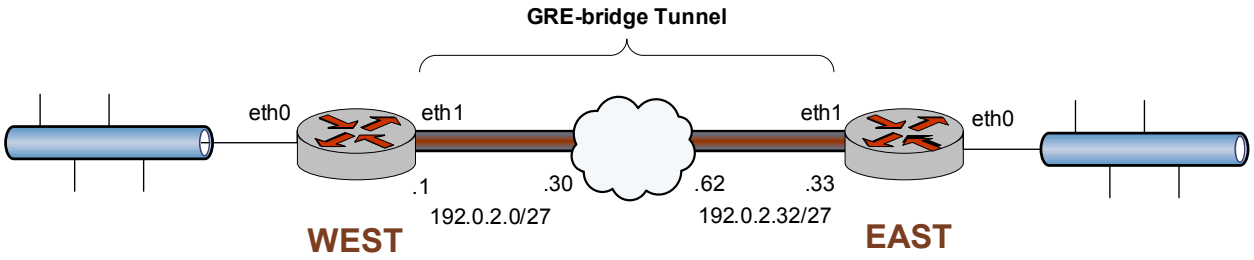
Bridging Across a WAN Using a GRE Tunnel

This section presents a sample configuration for bridging remote network segments using a GRE-bridge encapsulated tunnel between Vyatta systems WEST and EAST. First WEST is configured, and then EAST.

This basic tunnel is not protected by a key: this means it is not secure.

When you have finished, these systems will be configured as shown in Figure 4-2 with bridged network segments connected to eth0 on each of the two systems.

Figure 4-2 Bridging across a WAN using a GRE-bridge encapsulated tunnel



Configure WEST

GRE tunnels are explained in detail in the *Vyatta Tunnels Reference Guide*. Please see that guide for further details.

The GRE-bridge tunnel in the example configuration extends from eth1 on WEST through the wide-area network to eth1 on EAST. In this example, you create the bridge interface, add eth0 to the bridge group, and then create a tunnel interface and add it to the bridge group.

- The source IP address of the tunnel endpoint (the **local-ip**) is the same as the address associated with eth1 in this example.
- The destination IP address of the tunnel endpoint (the **remote-ip**) is 192.0.2.33 on EAST.
- The tunnel encapsulation is **gre-bridge**.
- The tunnel is added to the bridge group.

Example 4-2 creates the bridge and tunnel interfaces and adds eth0 and the tunnel interface to the bridge group. To do this, perform the following steps on WEST in configuration mode.

Example 4-2 Creating a basic GRE-bridge tunnel endpoint and bridge on WEST

Step	Command
Create the bridge interface.	`vyatta@WEST# set interfaces bridge br0`
Add eth0 to the bridge group.	`vyatta@WEST# set interfaces ethernet eth0 bridge-group bridge br0`
Configure an address on eth1.	`vyatta@WEST# set interfaces ethernet eth1 address 192.0.2.1/27`
Create the tunnel interface and specify the source IP address for the tunnel.	`vyatta@WEST# set interfaces tunnel tun0 local-ip 192.0.2.1`

Example 4-2 Creating a basic GRE-bridge tunnel endpoint and bridge on WEST

Specify the IP address of the other end of the tunnel.	`vyatta@WEST# set interfaces tunnel tun0 remote-ip 192.0.2.33`
Specify the GRE-bridge encapsulation mode for the tunnel.	`vyatta@WEST# set interfaces tunnel tun0 encapsulation gre-bridge`
Add tun0 to the bridge group.	`vyatta@WEST# set interfaces tunnel tun0 bridge-group bridge br0`
Commit the configuration.	`vyatta@WEST# commit`

View the configuration.	`vyatta@WEST# show interfaces`
	`bridge br0 {`
	`}`
	`ethernet eth0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`}`
	`ethernet eth1 {`
	`address 192.0.2.1/27`
	`}`
	`tunnel tun0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`encapsulation gre-bridge`
	`local-ip 192.0.2.1`
	`remote-ip 192.0.2.33`
	`}`

Configure EAST

EAST is configured similarly to WEST. The differences are:

- The address assigned to eth1.
- The **local-ip** address.
- The **remote-ip** address.

Example 4-3 shows the completed configuration.

Example 4-3 Configuration for a basic GRE-bridge tunnel endpoint and bridge on EAST

Step	Command

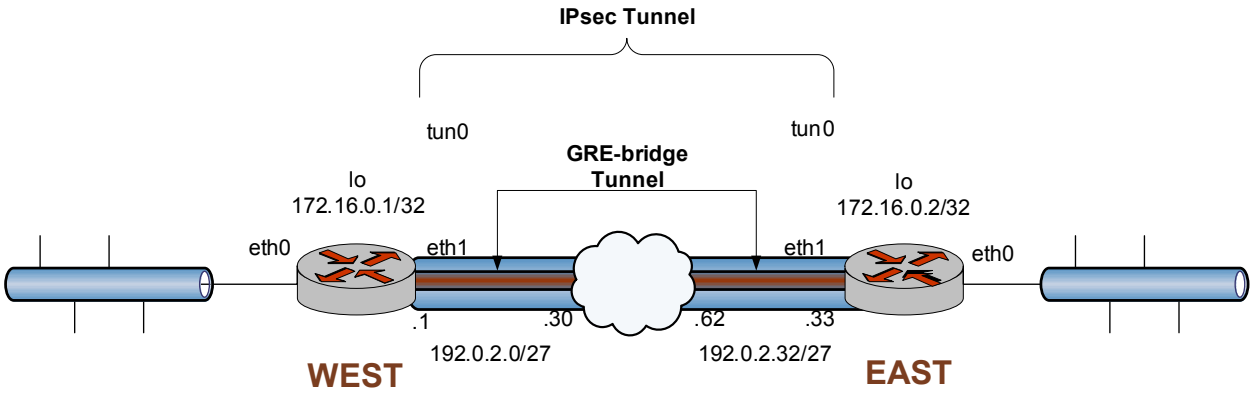
View the configuration.	vyatta@EAST# **show interfaces**
	bridge br0 {
	}
	ethernet eth0 {
	bridge-group {
	bridge br0
	}
	}
	ethernet eth1 {
	address 192.0.2.33/27
	}
	tunnel tun0 {
	bridge-group {
	bridge br0
	}
	encapsulation gre-bridge
	local-ip 192.0.2.33
	remote-ip 192.0.2.1
	}

Bridging Across a WAN Using a GRE Tunnel over IPsec VPN

This example configures a GRE-bridge tunnel between WEST and EAST and protects it within an IPsec tunnel between the same endpoints.

When you have finished, WEST and EAST will be configured as shown in Figure 4-3.

Figure 4-3 GRE-bridge tunnel protected by an IPsec tunnel



Configure WEST

This section presents the following examples:

- Example 4-4 Defining the bridge, Ethernet, and loopback interfaces on WEST
- Example 4-5 Defining the GRE-bridge tunnel from WEST to EAST
- Example 4-6 Defining the IPsec tunnel from WEST to EAST

Define the Bridge, Ethernet, and Loopback Interfaces on "WEST"

Example 4-4 defines the bridge, Ethernet, and loopback interfaces on WEST. In this example:

- The bridge interface br0 is created.
- Ethernet interface eth0 is added to the bridge group
- Ethernet interface eth1 is configured with IP address 192.0.2.1/27.
- Loopback interface lo is configured with IP address 172.16.0.1/32.

To create the bridge, Ethernet, and loopback interfaces on WEST, perform the following steps in configuration mode:

Example 4-4 Defining the bridge, Ethernet, and loopback interfaces on WEST

Step	Command
Create the bridge interface.	`vyatta@WEST# set interfaces bridge br0`
Add eth0 to the bridge group.	`vyatta@WEST# set interfaces ethernet eth0 bridge-group bridge br0`

Example 4-4 Defining the bridge, Ethernet, and loopback interfaces on WEST

Configure an address on eth1.	`vyatta@WEST# set interfaces ethernet eth1 address 192.0.2.1/27`
Configure an address on lo.	`vyatta@WEST# set interfaces loopback lo address 172.16.0.1/32`

Define the GRE Tunnel on "WEST"

GRE tunnels are explained in detail in the *Vyatta Tunnels Reference Guide*. Please see that guide for further details.

Example 4-5 defines WEST's end of the GRE-bridge tunnel. In this example:

- The IP address on the local side of the GRE tunnel (**local-ip**) is assigned the local loopback address 172.16.0.1.
- The IP address of the other end of the GRE tunnel (**remote-ip**) is assigned the loopback address of the remote system 172.16.0.2.
- The tunnel encapsulation is **gre-bridge**.
- The tunnel is added to the bridge group.

To create the tunnel interface and the tunnel endpoint on WEST, perform the following steps in configuration mode:

Example 4-5 Defining the GRE-bridge tunnel from WEST to EAST

Step	Command
Specify the local IP address for the GRE tunnel.	`vyatta@WEST# set interfaces tunnel tun0 local-ip 172.16.0.1`
Specify the remote IP address for the GRE tunnel.	`vyatta@WEST# set interfaces tunnel tun0 remote-ip 172.16.0.2`
Specify the encapsulation mode for the tunnel.	`vyatta@WEST# set interfaces tunnel tun0 encapsulation gre-bridge`
Add tun0 to the bridge group.	`vyatta@WEST# set interfaces tunnel tun0 bridge-group bridge br0`
Commit the configuration.	`vyatta@WEST# commit`

Example 4-5 Defining the GRE-bridge tunnel from WEST to EAST

View the modified configuration.

```
vyatta@WEST# show interfaces
    bridge br0 {
    }
    ethernet eth0 {
        bridge-group {
            bridge br0
        }
    }
    ethernet eth1 {
        address 192.0.2.1/27
    }
    loopback lo{
        address 172.16.0.1/32
    }
    tunnel tun0 {
        bridge-group {
            bridge br0
        }
        encapsulation gre-bridge
        local-ip 172.16.0.1
        remote-ip 172.16.0.2
    }
```

Define the IPsec Tunnel on "WEST"

Example 4-6 creates the IPsec tunnel from WEST to EAST.

- WEST uses IP address 192.0.2.1 on eth1.
- EAST uses IP address 192.0.2.33 on eth1.
- The IKE group is IKE-1W
- The preshared secret is "test_key_1".
- The IPsec tunnel is between subnet 172.16.0.1/32 on WEST and 172.16.0.2/32 on EAST, using ESP group ESP-1W.

This examples assumes that you have already configured the following:

- IKE group IKE-1W
- ESP group ESP-1W

IKE and ESP groups are explained in detail in the *Vyatta VPN Reference Guide*. Please see that guide for further details.

To create the IPsec tunnel from WEST to EAST, perform the following steps on WEST in configuration mode:

Example 4-6 Defining the IPsec tunnel from WEST to EAST

Step	Command
Enable VPN on eth1.	vyatta@WEST# **set vpn ipsec ipsec-interfaces interface eth1**
Define the site-to-site connection to EAST. Set the authentication mode.	vyatta@WEST# **set vpn ipsec site-to-site peer 192.0.2.33 authentication mode pre-shared-secret**
Navigate to the node for the peer for easier editing.	vyatta@WEST# **edit vpn ipsec site-to-site peer 192.0.2.33** [edit vpn ipsec site-to-site peer 192.0.2.33]
Provide the string that will be used to authenticate the peers.	vyatta@WEST# **set authentication pre-shared-secret test_key_1** [edit vpn ipsec site-to-site peer 192.0.2.33]
Specify the IKE group.	vyatta@WEST# **set ike-group IKE-1W** [edit vpn ipsec site-to-site peer 192.0.2.33]
Identify the IP address on this system to be used for this connection.	vyatta@WEST# **set local-ip 192.0.2.1** [edit vpn ipsec site-to-site peer 192.0.2.33]
Create a tunnel configuration, and provide the local subnet for this tunnel.	vyatta@WEST# **set tunnel 1 local-subnet 172.16.0.1/32** [edit vpn ipsec site-to-site peer 192.0.2.33]
Specify the remote subnet for the tunnel.	vyatta@WEST# **set tunnel 1 remote-subnet 172.16.0.2/32** [edit vpn ipsec site-to-site peer 192.0.2.33]
Specify the ESP group for this tunnel.	vyatta@WEST# **set tunnel 1 esp-group ESP-1W** [edit vpn ipsec site-to-site peer 192.0.2.33]
Return to the top of the configuration hierarchy.	vyatta@WEST# **top**
Commit the configuration.	vyatta@WEST# **commit**

Example 4-6 Defining the IPsec tunnel from WEST to EAST

View the modified configuration.

```
vyatta@WEST# show vpn ipsec ipsec-interfaces interface
 eth1
vyatta@WEST# show vpn ipsec site-to-site peer 192.0.2.33
 authentication
      mode pre-shared-secret
      pre-shared-secret test_key_1
 }
 ike-group IKE-1W
 local-ip 192.0.2.1
 tunnel 1 {
      esp-group ESP-1W
      local-subnet 172.16.0.1/32
      remote-subnet 172.16.0.2/32
 }
```

Configure EAST

EAST is configured similarly to WEST. The differences in the interface configuration are:

- The address assigned to eth1.
- The address assigned to lo.
- The **local-ip** address.
- The **remote-ip** address.

Example 4-7 shows the completed interfaces configuration.

Example 4-7 Configuration for interfaces on EAST

Step	Command

View the modified configuration.	`vyatta@EAST# show interfaces`
	`bridge br0 {`
	`}`
	`ethernet eth0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`}`
	`ethernet eth1 {`
	`address 192.0.2.33/27`
	`}`
	`loopback lo{`
	`address 172.16.0.2/32`
	`}`
	`tunnel tun0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`encapsulation gre-bridge`
	`local-ip 172.16.0.2`
	`remote-ip 172.16.0.1`
	`}`

The differences in the IPsec VPN configuration are:

- The **vpn ipsec site-to-site peer** address.
- The **ike-group**.
- The **local-ip** address.
- The **esp-group**.
- The **local-subnet** address.
- The **remote-subnet** address.

Example 4-8 shows the completed IPsec VPN configuration.

Example 4-8 Configuration for IPsec VPN on EAST

Step	Command

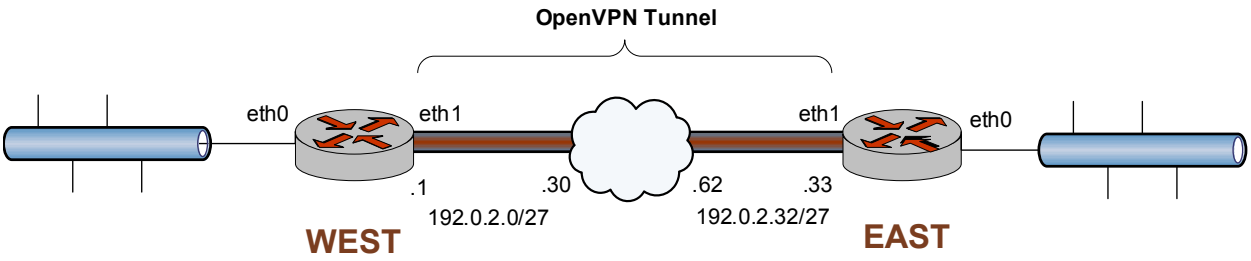
View the modified configuration.	vyatta@EAST# **show vpn ipsec ipsec-interfaces interface** eth1
	vyatta@EAST# **show vpn ipsec site-to-site peer 192.0.2.1**
	authentication
	mode pre-shared-secret
	pre-shared-secret test_key_1
	}
	ike-group IKE-1E
	local-ip 192.0.2.33
	tunnel 1 {
	esp-group ESP-1E
	local-subnet 172.16.0.2/32
	remote-subnet 172.16.0.1/32
	}

Bridging Across a WAN Using Site-to-Site OpenVPN

This example configures an OpenVPN tunnel between WEST and EAST.

When you have finished, WEST and EAST will be configured as shown in Figure 4-4.

Figure 4-4 OpenVPN tunnel



Configure WEST

This section presents the following examples:

- Example 4-9 Defining the bridge and Ethernet interfaces on WEST
- Example 4-10 Defining the OpenVPN tunnel from WEST to EAST
- Example 4-11 Configuration for all interfaces on EAST

Define the Bridge and Ethernet Interfaces on "WEST"

Example 4-9 defines the bridge and Ethernet interfaces on WEST. In this example:

- The bridge interface br0 is created.
- Ethernet interface eth0 is added to the bridge group
- Ethernet interface eth1 is configured with IP address 192.0.2.1/27.

To create the bridge and Ethernet interfaces on WEST, perform the following steps in configuration mode:

Example 4-9 Defining the bridge and Ethernet interfaces on WEST

Step	Command
Create the bridge interface.	`vyatta@WEST# set interfaces bridge br0`
Add eth0 to the bridge group.	`vyatta@WEST# set interfaces ethernet eth0 bridge-group bridge br0`
Configure an address on eth1.	`vyatta@WEST# set interfaces ethernet eth1 address 192.0.2.1/27`

Define the OpenVPN Tunnel on "WEST"

OpenVPN tunnels are explained in detail in the *Vyatta VPN Reference Guide*. Please see that guide for further details.

Example 4-10 defines WEST's end of the OpenVPN tunnel. In this example:

- The OpenVPN tunnel mode is set to **site-to-site**.
- The IP address of the other end of the OpenVPN tunnel (**remote-host**) is assigned the address of the remote system 192.0.2.33.
- The location of the shared secret file is specified.
- The OpenVPN tunnel is added to the bridge group.

To create the OpenVPN interface on WEST, perform the following steps in configuration mode:

Example 4-10 Defining the OpenVPN tunnel from WEST to EAST

Step	Command
Create the OpenVPN tunnel, and specify the mode to be used.	`vyatta@WEST# set interfaces openvpn vtun0 mode site-to-site`
Specify the remote host address for the OpenVPN tunnel.	`vyatta@WEST# set interfaces openvpn vtun0 remote-host 192.0.2.33`

Example 4-10 Defining the OpenVPN tunnel from WEST to EAST

Specify the file containing the shared secret.	`vyatta@WEST# set interfaces openvpn vtun0 shared-secret-key-file /root/secret`
Add vtun0 to the bridge group.	`vyatta@WEST# set interfaces openvpn vtun0 bridge-group bridge br0`
Commit the configuration.	`vyatta@WEST# commit`

View the modified configuration.	`vyatta@WEST# show interfaces`
	`bridge br0 {`
	`}`
	`ethernet eth0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`}`
	`ethernet eth1 {`
	`address 192.0.2.1/27`
	`}`
	`openvpn vtun0 {`
	`bridge-group {`
	`bridge br0`
	`}`
	`mode site-to-site`
	`remote-host 192.0.2.33`
	`shared-secret-key-file /root/secret`
	`}`

Configure EAST

EAST is configured similarly to WEST. The differences in the interface configuration are:

- The address assigned to eth1.
- The **remote-host** address.

Example 4-11 shows the completed interfaces configuration.

Example 4-11 Configuration for all interfaces on EAST

Step	Command

View the modified configuration.	vyatta@EAST# **show interfaces**
	bridge br0 {
	}
	ethernet eth0 {
	bridge-group {
	bridge br0
	}
	}
	ethernet eth1 {
	address 192.0.2.33/27
	}
	openvpn vtun0 {
	bridge-group {
	bridge br0
	}
	mode site-to-site
	remote-host 192.0.2.1
	shared-secret-key-file /root/secret
	}

Bridging Commands

This chapter contains the following commands.

Configuration Commands	
Bridge Groups	
interfaces bridge <brx>	Defines a bridge group.
interfaces bridge <brx> address <address>	Assigns an address to a bridge group.
interfaces bridge <brx> aging <age>	Specifies the MAC address aging timeout for a bridge group.
interfaces bridge <brx> description <desc>	Specifies a description for a bridge group.
interfaces bridge <brx> disable	Disables a bridge group without discarding configuration.
interfaces bridge <brx> disable-link-detect	Directs a bridge group not to detect physical link-state changes.
interfaces bridge <brx> forwarding-delay <delay>	Specifies the amount of time a bridge group keeps listening after a topology change.
interfaces bridge <brx> hello-time <interval>	Specifies the hello packet interval for a bridge group.
interfaces bridge <brx> max-age <interval>	Specifies how long a bridge group waits for a hello packet from the spanning tree root.
interfaces bridge <brx> priority <priority>	Specifies the forwarding priority of a bridge group in the spanning tree.
interfaces bridge <brx> stp <state>	Enables IEEE 802.1D Spanning Tree Protocol on a bridge group.
Ethernet Interfaces	
interfaces ethernet <ethx> bridge-group bridge <group-id>	Assigns an Ethernet interface to a bridge group.
interfaces ethernet <ethx> bridge-group cost <cost>	Specifies a path cost for a specific Ethernet interface within a bridge group.
interfaces ethernet <ethx> bridge-group priority <priority>	Specifies a path priority for an Ethernet interface within a bridge group.

Ethernet Vifs	
interfaces ethernet <ethx> vif <vlan-id> bridge-group bridge <group-id>	Assigns an Ethernet vif to a bridge group.
interfaces ethernet <ethx> vif <vlan-id> bridge-group cost <cost>	Specifies a path cost for an Ethernet vif within a bridge group.
interfaces ethernet <ethx> vif <vlan-id> bridge-group priority <priority>	Specifies a path priority for an Ethernet vif within a bridge group.
Ethernet Link Bonding Interfaces	
interfaces bonding <bondx> bridge-group bridge <group-id>	Assigns an Ethernet link bonding interface to a bridge group.
interfaces bonding <bondx> bridge-group cost <cost>	Specifies a path cost for a specific Ethernet link bonding interface within a bridge group.
interfaces bonding <bondx> bridge-group priority <priority>	Specifies a path priority for an Ethernet link bonding interface within a bridge group.
Ethernet Link Bonding Interface Vifs	
interfaces bonding <bondx> vif <vlan-id> bridge-group bridge <group-id>	Assigns an Ethernet link bonding interface vif to a bridge group.
interfaces bonding <bondx> vif <vlan-id> bridge-group cost <cost>	Specifies a path cost for a specific Ethernet link bonding interface vif within a bridge group.
interfaces bonding <bondx> vif <vlan-id> bridge-group priority <priority>	Specifies a path priority for an Ethernet link bonding interface vif within a bridge group.
Openvpn Interfaces	
interfaces openvpn <vtunx> bridge-group bridge <group-id>	Assigns an openvpn interface to a bridge group.
interfaces openvpn <vtunx> bridge-group cost <cost>	Specifies a path cost for a specific openvpn interface within a bridge group.
interfaces openvpn <vtunx> bridge-group priority <priority>	Specifies a path priority for a openvpn interface within a bridge group.
Tunnel Interfaces	
interfaces tunnel <tunx> bridge-group bridge <group-id>	Assigns a tunnel interface to a bridge group.
interfaces tunnel <tunx> bridge-group cost <cost>	Specifies a path cost for a specific tunnel interface within a bridge group.
interfaces tunnel <tunx> bridge-group priority <priority>	Specifies a path priority for a tunnel interface within a bridge group.

Wireless Interfaces	
interfaces wireless <wlanx> bridge-group bridge <group-id>	Assigns a wireless interface to a bridge group.
interfaces wireless <wlanx> bridge-group cost <cost>	Specifies a path cost for a specific wireless interface within a bridge group.
interfaces wireless <wlanx> bridge-group priority <priority>	Specifies a path priority for a wireless interface within a bridge group.
Operational Commands	
clear interfaces bridge counters	Clears bridge interface statistics.
show bridge	Displays the information for active bridge groups.
show interfaces bridge	Shows bridge interface information.

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

Related Commands Documented Elsewhere	
show interfaces ethernet	Displays information and statistics about Ethernet interfaces. See page 33.
Firewall	Commands for configuring firewall on Bridge interfaces are described in the *Vyatta Firewall Reference Guide*.
IPv6	Commands for configuring IPv6 on Bridge interfaces are described in the *Vyatta Guide to IPv6 Support*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Bridge interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on Bridge interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring qulaity of service on Bridge interfaces are described in the *Vyatta QoS Reference Guide*.
ARP commands	Commands for working with Address Resolution Protocol are described in *Vyatta Basic System Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the *Vyatta Basic System Reference Guide*.

clear interfaces bridge counters

Clears bridge interface statistics.

Syntax

clear interfaces bridge [*if-name*] **counters**

Command Mode

Operational mode.

Parameters

if-name	The identifier for the interface whose bridging counters you wish to clear. This may be an Ethernet interface, an Ethernet link bonding interface, or an Ethernet VLAN interface (a vif, specified as **ethx.vify**).

Default

Statistics are cleared on all bridge interfaces.

Usage Guidelines

Use this command to clear bridge statistics on Ethernet interfaces.

If no Ethernet interface is specified then statistics are cleared on all bridge interfaces.

interfaces bonding <bondx> bridge-group bridge <group-id>

Assigns an Ethernet link bonding interface to a bridge group.

Syntax

set interfaces bonding *bondx* **bridge-group bridge** *group-id*

delete interfaces bonding *bondx* **bridge-group bridge**

show interfaces bonding *bondx* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        bridge-group {
            bridge group-id
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign an Ethernet link bonding interface to a bridge group.

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

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

Use the **show** form of this command to view the bridge group membership information for an Ethernet link bonding interface.

interfaces bonding <bondx> bridge-group cost <cost>

Specifies a path cost for a specific Ethernet link bonding interface within a bridge group.

Syntax

set interfaces bonding *bondx* **bridge-group cost** *cost*

delete interfaces bonding *bondx* **bridge-group cost**

show interfaces bonding *bondx* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        bridge-group {
            cost cost
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for an Ethernet link bonding interface within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces bonding <bondx> bridge-group priority <priority>

Specifies a path priority for an Ethernet link bonding interface within a bridge group.

Syntax

set interfaces bonding *bondx* **bridge-group priority** *priority*

delete interfaces bonding *bondx* **bridge-group priority**

show interfaces bonding *bondx* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        bridge-group {
            priority priority
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for an Ethernet link bonding interface within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

interfaces bonding <bondx> vif <vlan-id> bridge-group bridge <group-id>

Assigns an Ethernet link bonding interface vif to a bridge group.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **bridge-group bridge** *group-id*

delete interfaces bonding *bondx* **vif** *vlan-id* **bridge-group bridge**

show interfaces bonding *bondx* **vif** *vlan-id* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            bridge-group {
                bridge group-id
            }
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign an Ethernet link bonding interface vif to a bridge group.

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

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

Use the **show** form of this command to view the bridge group membership information for an Ethernet link bonding interface vif.

interfaces bonding <bondx> vif <vlan-id> bridge-group cost <cost>

Specifies a path cost for a specific Ethernet link bonding interface vif within a bridge group.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **bridge-group cost** *cost*

delete interfaces bonding *bondx* **vif** *vlan-id* **bridge-group cost**

show interfaces bonding *bondx* **vif** *vlan-id* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            bridge-group {
                cost cost
            }
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for an Ethernet link bonding interface vif within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces bonding <bondx> vif <vlan-id> bridge-group priority <priority>

Specifies a path priority for an Ethernet link bonding interface vif within a bridge group.

Syntax

set interfaces bonding *bondx* **vif** *vlan-id* **bridge-group priority** *priority*

delete interfaces bonding *bondx* **vif** *vlan-id* **bridge-group priority**

show interfaces bonding *bondx* **vif** *vlan-id* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bonding bondx {
        vif vlan-id {
            bridge-group {
                priority priority
            }
        }
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
vlan-id	The VLAN ID for the vif. The range is 0 to 4094.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for an Ethernet link bonding interface vif within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

interfaces bridge <brx>

Defines a bridge group.

Syntax

set interfaces bridge *brx*

delete interfaces bridge *brx*

show interfaces bridge *brx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
    }
}
```

Parameters

brx	Multi-node. The identifier for the bridge group. Supported identifiers are **br0** through **br999**. You can define multiple bridge groups by creating more than one **bridge** configuration node.

Default

None.

Usage Guidelines

Use this command to define a bridge group. Note that you must create the bridge group (using this command) before you can assign interfaces to it.

Use the **set** form of this command to create the bridge group and define bridge settings.

Use the **delete** form of this command to remove all configuration for a bridge group.

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

interfaces bridge <brx> address <address>

Assigns an address to a bridge group.

Syntax

set interfaces bridge *brx* **address** *address*

delete interfaces bridge *brx* **address** *address*

show interfaces bridge *brx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        address address
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
address	Multi-node. The IP address and network prefix for the interface. The address must either be in the form *ip-address/prefix* or **dhcp**. If it is **dhcp**, then the IP address and network prefix are set using the Dynamic Host Configuration Protocol (DHCP). You can assign multiple addresses to a bridge group by creating multiple **address** configuration nodes.

Default

None.

Usage Guidelines

Use this command to assign an address to a bridge group.

Use the **set** form of this command to set the address for the bridge group.

Use the **delete** form of this command to remove address configuration for the bridge group

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

interfaces bridge <brx> aging <age>

Specifies the MAC address aging timeout for a bridge group.

Syntax

set interfaces bridge *brx* **aging** *age*

delete interfaces bridge *brx* **aging**

show interfaces bridge *brx* **aging**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        aging age
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
age	The length of time, in seconds, that a MAC address is kept before being aged out. The range is 1 to 4294967295. The default is 300.

Default

MAC addresses are aged out of the forwarding database after 300 seconds (5 minutes).

Usage Guidelines

Use this command to specify the length of time that a dynamic MAC address entry is kept in a bridge's forwarding database. If this interval expires without the entry being updated, the entry is aged out of the table.

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

Use the **delete** form of this command to restore the default MAC address aging timeout.

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

interfaces bridge <brx> description <desc>

Specifies a description for a bridge group.

Syntax

set interfaces bridge *brx* **description** *desc*

delete interfaces bridge *brx* **description**

show interfaces bridge *brx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        description desc
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
desc	A brief description for the bridge group.

Default

None.

Usage Guidelines

Use this command to specify a description for the bridge group.

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

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

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

interfaces bridge <brx> disable

Disables a bridge group without discarding configuration.

Syntax

set interfaces bridge *brx* **disable**

delete interfaces bridge *brx* **disable**

show interfaces bridge *brx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        disable
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
disable	Disables bridging on this bridge group.

Default

Bridging is enabled.

Usage Guidelines

Use this command to disable a bridge group.

Use the **set** form of this command to specify whether to disable bridging on the interface.

Use the **delete** form of this command to restore the default value for the bridge group.

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

interfaces bridge <brx> disable-link-detect

Directs a bridge group not to detect physical link-state changes.

Syntax

set interfaces bridge *brx* **disable-link-detect**

delete interfaces bridge *brx* **disable-link-detect**

show interfaces bridge *brx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        disable-link-detect
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.

Default

The interface detects physical link state changes.

Usage Guidelines

Use this command to direct a bridge group to not detect physical state change to the 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 bridge group configuration.

interfaces bridge <brx> forwarding-delay <delay>

Specifies the amount of time a bridge group keeps listening after a topology change.

Syntax

set interfaces bridge *brx* **forwarding-delay** *delay*

delete interfaces bridge *brx* **forwarding-delay**

show interfaces bridge *brx* **forwarding-delay**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        forwarding-delay delay
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
delay	The amount of time, in seconds, the bridge keeps learning about the topology of the spanning tree after a topology change. The range is 1 to 4294967295. The default is 15.

Default

The the bridge listens for 15 seconds before transitioning to Forwarding state.

Usage Guidelines

Use this command to specify the amount of time the bridge will keep listening after a topology change.

After a topology change, the bridge remains in a listening state for the forward delay period, learning about the topology of the spanning tree for this interval. During this period, no traffic is forwarded. After the forward delay interval has passed, the bridge transitions to the forwarding state and begins to forward traffic again.

Use the **set** form of this command to specify the amount of time the bridge will keep listening after a topology change.

Use the **delete** form of this command to restore the forwarding-delay to its default.

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

interfaces bridge <brx> hello-time <interval>

Specifies the hello packet interval for a bridge group.

Syntax

set interfaces bridge *brx* **hello-time** *interval*

delete interfaces bridge *brx* **hello-time**

show interfaces bridge *brx* **hello-time**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        hello-time interval
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
interval	The interval in seconds at which this bridge will transmit hello packets. The range is 1 to 4294967295. The default is 2.

Default

The default is 2.

Usage Guidelines

Use this command to specify the “hello packet” interval.

Hello packets are Bridge Protocol Data Units (BPDUs) used as messages to communicate the state of the spanning tree topology. On a spanning tree, hello packets are sent by the bridge that assumes itself to be the root bridge.

Use the **set** form of this command to specify the hello packet interval.

Use the **delete** form of this command to restore the hello packet interval to the default value.

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

interfaces bridge <brx> max-age <interval>

Specifies how long a bridge group waits for a hello packet from the spanning tree root.

Syntax

set interfaces bridge *brx* **max-age** *interval*

delete interfaces bridge *brx* **max-age**

show interfaces bridge *brx* **max-age**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        max-age interval
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
interval	The interval a bridge group waits to receive a hello packet before recomputing the spanning-tree topology. The range is 1 to 4294967295. The default is 20.

Default

The bridge group waits 20 seconds for a hello packet before recomputing the spanning-tree topology.

Usage Guidelines

Use this command to specify the interval a bridge group will wait to receive a hello packet from the spanning tree root. If this interval expires without the bridge group having received the hello packet, the bridge group considers the network topology to have changed and recomputes the spanning-tree topology.

Use the **set** form of this command to specify the maximum age interval.

Use the **delete** form of this command to restore the maximum age interval to its default value.

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

interfaces bridge <brx> priority <priority>

Specifies the forwarding priority of a bridge group in the spanning tree.

Syntax

set interfaces bridge *brx* **priority** *priority*

delete interfaces bridge *brx* **priority**

show interfaces bridge *brx* **priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        priority priority
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
priority	The forwarding priority of this bridge in the spanning tree. The higher the number, the lower the priority. The default is 0, which is the highest priority.

Default

The default is 0.

Usage Guidelines

Use this command to specify the forwarding priority of this bridge in the spanning tree.

The Spanning Tree Protocol uses the bridge priority to determine the spanning tree root. The lower the number assigned to the bridge group, the higher its priority, and the more likely it is to be selected as the root of the spanning tree.

Use the **set** form of this command to specify the forwarding priority of this bridge in the spanning tree.

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

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

interfaces bridge <brx> stp <state>

Enables IEEE 802.1D Spanning Tree Protocol on a bridge group.

Syntax

set interfaces bridge *brx* **stp** *state*

delete interfaces bridge *brx* **stp**

show interfaces bridge *brx* **stp**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    bridge brx {
        stp state
    }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br999**.
stp	Allows you to enable or disable the Spanning Tree Protocol on a per-bridge basis. Supported values are as follows: **true**: Enables Spanning Tree Protocol on this bridge. **false**: Disables Spanning Tree Protocol on this bridge. The default is **false**.

Default

Spanning Tree Protocol is disabled.

Usage Guidelines

Use this command to specify whether or not the IEEE 802.1D Spanning Tree Protocol (STP) is enabled on a bridge group. When STP is enabled on bridge group, it is enabled for all interfaces and vifs assigned to the bridge group.

Use the **set** form of this command to specify whether or not the Spanning Tree Protocol is enabled on the interface.

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

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

interfaces ethernet <ethx> bridge-group bridge <group-id>

Assigns an Ethernet interface to a bridge group.

Syntax

set interfaces ethernet *ethx* **bridge-group bridge** *group-id*

delete interfaces ethernet *ethx* **bridge-group bridge**

show interfaces ethernet *ethx* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        bridge-group {
            bridge group-id
        }
    }
}
```

Parameters

ethx	The Ethernet interface you are adding to the bridge group. Supported values are **eth0** through **eth23**. The interface must already be defined.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign an Ethernet interface to a bridge group.

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

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

Use the **show** form of this command to view the bridge group membership information for an Ethernet interface.

interfaces ethernet <ethx> bridge-group cost <cost>

Specifies a path cost for a specific Ethernet interface within a bridge group.

Syntax

set interfaces ethernet *ethx* **bridge-group cost** *cost*

delete interfaces ethernet *ethx* **bridge-group cost**

show interfaces ethernet *ethx* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        bridge-group {
            cost cost
        }
    }
}
```

Parameters

ethx	The Ethernet interface you are adding to the bridge group. Supported values are **eth0** through **eth23**. The interface must already be defined.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for an Ethernet interface within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces ethernet <ethx> bridge-group priority <priority>

Specifies a path priority for an Ethernet interface within a bridge group.

Syntax

set interfaces ethernet *ethx* **bridge-group priority** *priority*

delete interfaces ethernet *ethx* **bridge-group priority**

show interfaces ethernet *ethx* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        bridge-group {
            priority priority
        }
    }
}
```

Parameters

ethx	The Ethernet interface you are adding to the bridge group. Supported values are **eth0** through **eth23**. The interface must already be defined.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority for is 128.

Usage Guidelines

Use this command to specify a path priority for an Ethernet interface within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

interfaces ethernet <ethx> vif <vlan-id> bridge-group bridge <group-id>

Assigns an Ethernet vif to a bridge group.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group bridge** *group-id*

delete interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group bridge**

show interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            bridge-group {
                bridge group-id
            }
        }
    }
}
```

Parameters

ethx	The Ethernet interface on which the vif resides. Supported values are **eth0** through **eth23**. The interface must already be defined.
vlan-id	The VLAN ID for the vif you are adding to the bridge group. The range is 0 to 4095. The vif must already be defined.
group-id	The bridge group you are adding the vif to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to add an Ethernet vif to a bridge group.

Use the **set** form of this command to add the vif to the bridge group.

Use the **delete** form of this command to remove the Ethernet vif from the bridge group.

Use the **show** form of this command to view the bridge group membership information for an Ethernet vif.

interfaces ethernet <ethx> vif <vlan-id> bridge-group cost <cost>

Specifies a path cost for an Ethernet vif within a bridge group.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group cost** *cost*

delete interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group cost**

show interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            bridge-group {
                cost cost
            }
        }
    }
}
```

Parameters

ethx	The Ethernet interface on which the vif resides. Supported values are **eth0** through **eth23**. The interface must already be defined.
vlan-id	The VLAN ID for the vif you are adding to the bridge group. The range is 0 to 4095.
cost	The path cost for the vif within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for an Ethernet vif within a bridge group.

Use the **set** form of this command to specify the path cost.

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

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

interfaces ethernet <ethx> vif <vlan-id> bridge-group priority <priority>

Specifies a path priority for an Ethernet vif within a bridge group.

Syntax

set interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group priority** *priority*

delete interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group priority**

show interfaces ethernet *ethx* **vif** *vlan-id* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    ethernet ethx {
        vif vlan-id {
            bridge-group {
                priority priority
            }
        }
    }
}
```

Parameters

ethx	The Ethernet interface on which the vif resides. Supported values are **eth0** through **eth23**. The interface must already be defined.
vlan-id	The VLAN ID for the vif you are adding to the bridge group. The range is 0 to 4095.
priority	The path priority for the vif within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for a bridge group on a virtual interface.

Use the **set** form of this command to set the path priority.

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

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

interfaces openvpn <vtunx> bridge-group bridge <group-id>

Assigns an openvpn interface to a bridge group.

Syntax

set interfaces openvpn *vtunx* **bridge-group bridge** *group-id*

delete interfaces openvpn *vtunx* **bridge-group bridge**

show interfaces openvpn *vtunx* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    openvpn vtunx {
        bridge-group {
            bridge group-id
        }
    }
}
```

Parameters

vtunx	The identifier for the openvpn interface. Supported values are **vtun0** through **vtunx**, where *x* is a non-negative integer.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign a openvpn interface to a bridge group.

Use the **set** form of this command to add a openvpn interface to the bridge group.

Use the **delete** form of this command to remove a openvpn interface from the bridge group.

Use the **show** form of this command to view the bridge group membership information for a openvpn interface.

interfaces openvpn <vtunx> bridge-group cost <cost>

Specifies a path cost for a specific openvpn interface within a bridge group.

Syntax

set interfaces openvpn *vtunx* **bridge-group cost** *cost*

delete interfaces openvpn *vtunx* **bridge-group cost**

show interfaces openvpn *vtunx* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    openvpn vtunx {
        bridge-group {
            cost cost
        }
    }
}
```

Parameters

vtunx	The identifier for the openvpn interface. Supported values are **vtun0** through **vtunx**, where *x* is a non-negative integer.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for a openvpn interface within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces openvpn <vtunx> bridge-group priority <priority>

Specifies a path priority for a openvpn interface within a bridge group.

Syntax

set interfaces openvpn *vtunx* **bridge-group priority** *priority*

delete interfaces openvpn *vtunx* **bridge-group priority**

show interfaces openvpn *vtunx* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    openvpn vtunx {
        bridge-group {
            priority priority
        }
    }
}
```

Parameters

vtunx	The identifier for the openvpn interface. Supported values are **vtun0** through **vtunx**, where *x* is a non-negative integer.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for a openvpn interface within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

interfaces tunnel <tunx> bridge-group bridge <group-id>

Assigns a tunnel interface to a bridge group.

Syntax

set interfaces tunnel *tunx* **bridge-group bridge** *group-id*

delete interfaces tunnel *tunx* **bridge-group bridge**

show interfaces tunnel *tunx* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    tunnel tunx {
        bridge-group {
            bridge group-id
        }
    }
}
```

Parameters

tunx	The identifier for the tunnel interface. Supported values are **tun0** through **tun23**.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign a tunnel interface to a bridge group. A tunnel can only be added to a bridge group if it's encapsulation type is **gre-bridge**.

Use the **set** form of this command to add a tunnel interface to the bridge group.

Use the **delete** form of this command to remove a tunnel interface from the bridge group.

Use the **show** form of this command to view the bridge group membership information for a tunnel interface.

interfaces tunnel <tunx> bridge-group cost <cost>

Specifies a path cost for a specific tunnel interface within a bridge group.

Syntax

set interfaces tunnel *tunx* **bridge-group cost** *cost*

delete interfaces tunnel *tunx* **bridge-group cost**

show interfaces tunnel *tunx* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    tunnel tunx {
        bridge-group {
            cost cost
        }
    }
}
```

Parameters

tunx	The identifier for the tunnel interface. Supported values are **tun0** through **tun23**.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for a tunnel interface within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces tunnel <tunx> bridge-group priority <priority>

Specifies a path priority for a tunnel interface within a bridge group.

Syntax

set interfaces tunnel *tunx* **bridge-group priority** *priority*

delete interfaces tunnel *tunx* **bridge-group priority**

show interfaces tunnel *tunx* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    tunnel tunx {
        bridge-group {
            priority priority
        }
    }
}
```

Parameters

tunx	The identifier for the tunnel interface. Supported values are **tun0** through **tun23**.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for a tunnel interface within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

interfaces wireless <wlanx> bridge-group bridge <group-id>

Assigns a wireless interface to a bridge group.

Syntax

set interfaces wireless *wlanx* **bridge-group bridge** *group-id*

delete interfaces wireless *wlanx* **bridge-group bridge**

show interfaces wireless *wlanx* **bridge-group bridge**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        bridge-group {
            bridge group-id
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. Supported values are **wlan0** through **wlan999**.
group-id	The bridge group you are adding the interface to. Supported identifiers are **br0** through **br999**.

Default

None.

Usage Guidelines

Use this command to assign a wireless interface to a bridge group.

Use the **set** form of this command to add a wireless interface to the bridge group.

Use the **delete** form of this command to remove a wireless interface from the bridge group.

Use the **show** form of this command to view the bridge group membership information for a wireless interface.

interfaces wireless <wlanx> bridge-group cost <cost>

Specifies a path cost for a specific wireless interface within a bridge group.

Syntax

set interfaces wireless *wlanx* **bridge-group cost** *cost*

delete interfaces wireless *wlanx* **bridge-group cost**

show interfaces wireless *wlanx* **bridge-group cost**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        bridge-group {
            cost cost
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. Supported values are **wlan0** through **wlan999**.
cost	The path cost for the interface within its bridge group. The range is 0 to 2147483647. The default is 19.

Default

The path cost is 19.

Usage Guidelines

Use this command to specify a path cost for a wireless interface within a bridge group. The Spanning Tree Protocol (STP) uses this value to calculate the shortest path from this bridge group to the spanning tree root.

Use the **set** form of this command to specify the path cost.

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

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

interfaces wireless <wlanx> bridge-group priority <priority>

Specifies a path priority for a wireless interface within a bridge group.

Syntax

set interfaces wireless *wlanx* **bridge-group priority** *priority*

delete interfaces wireless *wlanx* **bridge-group priority**

show interfaces wireless *wlanx* **bridge-group priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
    wireless wlanx {
        bridge-group {
            priority priority
        }
    }
}
```

Parameters

wlanx	The identifier for the wireless interface. Supported values are **wlan0** through **wlan999**.
priority	The path priority for the interface within its bridge group. The range is 0 to 255. The default is 128.

Default

The path priority is 128.

Usage Guidelines

Use this command to specify a path priority for a wireless interface within a bridge group.

Use the **set** form of this command to specify the path priority.

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

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

show bridge

Displays the information for active bridge groups.

Syntax

show bridge [*bridge-group* [**macs** | **spanning-tree**]]

Command Mode

Operational mode.

Parameters

bridge-group	Displays information for the specified bridge group: one of **br0** through **br999**.
macs	Shows the MAC table for the specified bridge group.
spanning-tree	Shows spanning tree information for the specified bridge groups.

Usage Guidelines

Use this command to display information about configured bridge groups.

When used with no option, this command displays information about all active bridge groups. When the identifier of a bridge group is provided, this command displays information for the specified bridge group. You can display the media access control (MAC) table and Spanning Tree Protocol information for a bridge group.

show interfaces bridge

Shows bridge interface information.

Syntax

show interfaces bridge [*bridge-group* [**brief**] | **detail**]

Command Mode

Operational mode.

Parameters

bridge-group	Displays information for the specified bridge group: one of **br0** through **br999**.
brief	Shows a summary of information for a given bridge group.
detail	Shows detailed bridge interface information.

Usage Guidelines

Use this command to display information about configured bridge interfaces.

When used with no option, this command displays information about all active bridge interfaces. When the identifier of a bridge group is provided, this command displays information for the specified bridge group.

Chapter 5: Ethernet Link Bonding

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

This chapter presents the following topics:

Ethernet Link Bonding Configuration

This section 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 (now called IEEE 802.1ax) specification 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 may vary, even between either end of the bonded virtual link. This could 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.

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.

Ethernet Bonding Configuration Example

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.

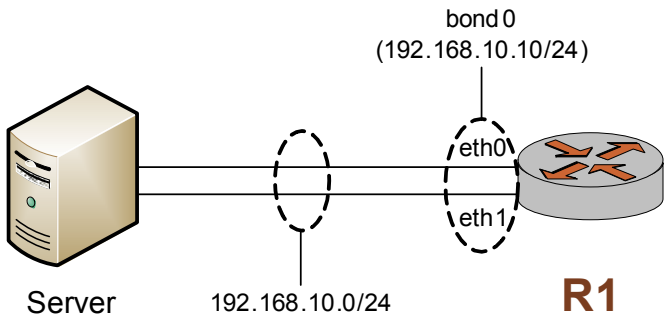
Figure 5-1 shows a simple Ethernet link bonding scenario, with an Ethernet bonded link consisting of two physical Ethernet links. In this example:

- 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 5-1 Creating a bond group with two Ethernet interfaces



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

Example 5-1 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 Configuration Example 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 5-2 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`

Example 5-2 Adding a VLAN to an existing bonding interface.

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
	}

Ethernet Link Bonding Commands

This chapter contains the following commands.

Configuration Commands	
Bond Group	
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> description <desc>	Specifies a description for an Ethernet link bond group.
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> 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.
Bond Group	
interfaces ethernet <ethx> bond-group <bondx>	Adds an Ethernet interface to a bonding group.
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	Commands for configuring bonded Ethernet links within bridge groups are described in "Chapter 4: Bridging."

Firewall	Commands for configuring firewall on bonded Ethernet links are described in the *Vyatta Firewall Reference Guide*.
QoS	Commands for configuring quality of service on bonded Ethernet links are described in the *Vyatta QoS Reference Guide*.
VLAN Interfaces	Commands for defining VLAN interfaces (vifs) on bonded Ethernet links are described in "Chapter 3: VLAN Interfaces."
VRRP	Commands for configuring Virtual Router Redundancy Protocol on bonded Ethernet links 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**}

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

show interfaces bonding *bondx* **address**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

bondx	Multi-node. 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.

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.

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> 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> 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> 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.

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. 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 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 bonding <bondx> 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.

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 5-3 shows the output for **show interfaces bonding**.

Example 5-3 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 5-4 shows the output for **show interfaces bonding slaves**.

Example 5-4 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 6: Pseudo-Ethernet Interfaces

This chapter describes explains 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 Configuration

This section presents the following topics:

Pseudo-Ethernet Interface Overview

A pseudo-Ethernet interface is a means of creating multiple virtual Ethernet devices, each with a difference 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 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 the limit applies to the 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 connected to a pseudo-Ethernet interface internal to a system from 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.
- Pseudo-Ethernet interfaces do not support VLANs, and it is not possible to link a pseudo-Ethernet interface to a VLAN.
- 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

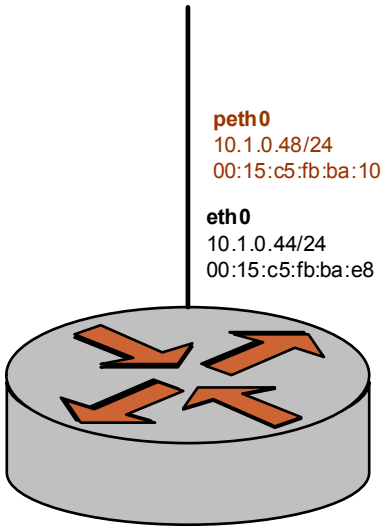
Pseudo-Ethernet Interface Configuration Examples

Figure 6-1 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 6-1 Creating a pseudo-Ethernet interface



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

Example 6-1 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`

Example 6-1 Creating a pseudo-Ethernet interface

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`

Pseudo-Ethernet Interface Commands

This chapter contains the following 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> 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> 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.
Operational Commands	
All operational commands applying to Ethernet interfaces can be used with pseudo-Ethernet interfaces. For these commands, see "Chapter 1: Ethernet Interfaces."	

All features that apply to Ethernet interfaces also apply to pseudo-Ethernet interfaces. Commands for using other system features with Ethernet interfaces can be found in the following locations.

Related Commands Documented Elsewhere	
Firewall	Commands for configuring firewall on Ethernet interfaces are described in the *Vyatta Firewall Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Ethernet interfaces are described in the *Vyatta OSPF Reference Guide*.
RIP and RIPng	Commands for configuring the Routing Information Protocol on Ethernet interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring quality of service on Ethernet interfaces are described in the *Vyatta QoS Reference Guide*.
System interfaces	Commands for showing the physical interfaces available on your system are described in the *Vyatta Basic System Reference Guide*.

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 ethernet *pethx* **address** {*ipv4* | *ipv6* | **dhcp**}

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

show interfaces ethernet *pethx* **address**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

pethx	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.
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.

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 ethernet *pethx* **description** *descr*

delete interfaces ethernet *pethx* **description**

show interfaces ethernet *pethx* **description**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

pethx	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.
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> 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	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.

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	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.

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> link <ethx>

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

Syntax

set interfaces ethernet *pethx* **link** *ethx*

delete interfaces ethernet *pethx* **link**

show interfaces ethernet *pethx* **link**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

pethx	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.
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 ethernet *pethx* **mac** *mac-addr*

delete interfaces ethernet *pethx* **mac**

show interfaces ethernet *pethx* **mac**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

pethx	Multi-node. An identifier for the pseudo-Ethernet interface you are defining; for example **peth0**.
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.

Chapter 7: Wireless Interfaces

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

This chapter presents the following topics:

Wireless Interface Configuration

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

The two primary modes of operation for a wireless interface are as a Wireless Access Point (WAP) and as a Station.

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.

Configuring a Wireless Access Point

The example in this section creates a Wireless Access Point (WAP). The WAP 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.

NOTE *When configuring multiple Wireless Access Point interfaces, unique IP addresses, channels, Network IDs (SSIDs), and MAC addresses must be specified.*

To create this WAP, perform the following steps:

Example 7-1 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`

Example 7-1 Configuring an Access Point

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 Wireless Access Point defined in Example 7-1 Configuring an Access Point.

In this case the default physical device (phy0) is used.

To configure a wireless interface as a Station, perform the following steps:

Example 7-2 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`

Example 7-2 Configuring a Station

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
	}

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> 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> 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	
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> capture	Captures and traffic on 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	Commands for configuring Wireless interfaces within bridge groups are described in "Chapter 4: Bridging."
Firewall	Commands for configuring firewall on Wireless interfaces are described in the *Vyatta Firewall Reference Guide*.
OSPF	Commands for configuring the Open Shortest Path First routing protocol on Wireless interfaces are described in the *Vyatta OSPF Reference Guide*.
QoS	Commands for configuring quality of service on Wireless interfaces are described in the *Vyatta QoS Reference Guide*.
RIP	Commands for configuring the Routing Information Protocol on Wireless interfaces are described in the *Vyatta RIP 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	Commands for configuring vifs on Wireless interfaces are described in "Chapter 3: 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. This may be **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 show interfaces wireless 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**}

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

show interfaces wireless *wlanx* **address**

Command Mode

Configuration mode.

Configuration Statement

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

Parameters

wlanx	The identifier for the wireless interface. This may be **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.

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. This may be **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. This may be **wlan0** to **wlan999**.
country	Indicates the country that the wireless interface is deployed in. The value is a two-letter country code as defined in ISO standard 639. 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. This may be **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> 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. This may be **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. This may be **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> 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. This may be **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. This may be **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. This may be **wlan0** to **wlan999**.
device	A identifier representing physical device to associate with the wireless interface. The range is **phy0** to **phy9**.

Default

None.

Usage Guidelines

Use this command to specify the physical device associated with the wireless interface.

This value is optional for the first wireless interface on a device but is required when there is more than one physical device.

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> 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. This may be **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. This may be **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. This may be **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. This may be **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.

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 7-3 shows information for all wireless interfaces.

Example 7-3 Displaying wireless interface information

```
vyatta@vyatta:~$ show interfaces wireless
Interface    IP Address         State       Link   Description
wlan0        192.168.40.1/24    up          up
```

Example 7-4 shows detailed information for all wireless interfaces.

Example 7-4 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 7-5 shows wireless-specific information for all wireless interfaces.

Example 7-5 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. This may be **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to view status and statistics on the specified wireless interface.

Examples

Example 7-6 shows status and statistics on interface wlan0.

Example 7-6 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. This may be **wlan0** to **wlan999**.

Default

None.

Usage Guidelines

Use this command to view brief status and statistics on the specified wireless interface.

Examples

Example 7-7 shows a brief status on interface wlan0.

Example 7-7 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> capture

Captures and traffic on a wireless interface.

Syntax

show interfaces wireless *wlanx* **capture**

Command Mode

Operational mode.

Parameters

wlanx	The identifier for the wireless interface. This may be **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 7-8 shows captured data on interface wlan0.

Example 7-8 Displaying captured data

```
vyatta@vyatta:~$ show interfaces wireless wlan0 capture
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 <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. This may be **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 7-9 shows queue information for interface wlan0.

Example 7-9 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. This may be **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 7-10 shows scan information on interface wlan0.

Example 7-10 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 7-11 shows detailed scan information on interface wlan0.

Example 7-11 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. This may be **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 7-12 shows station data on interface wlan0.

Example 7-12 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 8: Input Interfaces

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

This chapter presents the following topics:

Input Interface Configuration

An Input interface is a special-purpose interface that, used in conjunction with interface input redirection, is typically used 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

Note that 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).

Configuration examples showing Input interfaces used in conjunction with QoS are provided in the *Vyatta QoS Reference Guide*.

Configuring an Input Interface

The example in this section creates an Input interface.

To create an Input interface, perform the following steps:

Example 8-1 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 {`
	`}`

Input Interface Commands

Configuration Commands	
interfaces input <ifbx>	Defines an input interface.
interfaces input <ifbx> description <descr>	Specifies a description for an input interface.

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

Related Commands Documented Elsewhere	
Firewall	Commands for configuring firewall on Input interfaces are described in the *Vyatta Firewall Reference Guide*.
QoS	Commands for configuring quality of service on Input interfaces 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	Mandatory. Multi-node. The identifier for the input interface. This may be **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. This may be **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.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
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
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
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
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
ISP	Internet Service Provider
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
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
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
PKI	Public Key Infrastructure

PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN

VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access