VYATTA, INC. | Vyatta System

LAN Interfaces

REFERENCE GUIDE

Ethernet Interfaces
Loopback Interface
VLAN Interfaces
Bridging
Ethernet Link Bonding

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

COPYRIGHT

Vyatta reserves the right to make changes to software, hardware, and documentation without notice. For the most recent version of documentation, visit the Vyatta web site at vyatta.com.

PROPRIETARY NOTICES

ISSUE DATE: February 2009

DOCUMENT REVISION. VC5 v03

RELEASED WITH: VC5.0.2

PART NO. A0-0111-10-0002

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 guide explains how to use Vyatta features for high availability. It describes the available commands and provides configuration examples.

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

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

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

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 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.

This guide has the following chapters:

Document Conventions

This guide contains advisory paragraphs and uses typographic conventions.

Advisory Paragraphs

This guide uses the following advisory paragraphs:

Warnings alert you to situations that may pose a threat to personal safety, as in the following example:

WARNING *Risk of injury. Switch off power at the main breaker before attempting to connect the remote cable to the service power at the utility box.*

Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service, as in the following example:

CAUTION *Risk of loss of service. Restarting a running system will interrupt service.*

Notes provide information you might need to avoid problems or configuration errors:

NOTE *You must create and configure network interfaces before enabling them for routing protocols.*

Typographic Conventions

This document uses the following typographic conventions:

`Courier`	Examples, command-line output, and representations of configuration nodes.
`boldface Courier`	In an example, your input: something you type at a command line.
boldface	In-line commands, keywords, and file names .
italics	Arguments and variables, where you supply a value.
<key>	A key on your keyboard. Combinations of keys are joined by plus signs (“+”). An example is <Ctrl>+<Alt>+<Del>.
[*arg1* | *arg2*]	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.
arg1..*argN*	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, and eth3.
arg [*arg* ...] *arg*,[*arg*,...]	A value that can optionally represent a list of elements (a space-separated list in the first case, and a comma-separated list in the second case).

Vyatta Publications

More information about the Vyatta system is available in the Vyatta technical library, and on www.vyatta.com and www.vyatta.org.

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

Chapter 1: Ethernet Interfaces

This chapter describes basic configuration for Ethernet interfaces.

This chapter presents the following topics:

- Ethernet Interface Commands

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> bond-group <bondx>	Adds an Ethernet interface to a bonding group.
interfaces ethernet <ethx> description <descr>	Specifies a description for an Ethernet interface.
interfaces ethernet <ethx> 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> mac <mac-addr>	Sets the MAC address of an Ethernet interface.
interfaces ethernet <ethx> mtu <mtu>	Specifies the MTU for an Ethernet interface.
interfaces ethernet <ethx> smp_affinity <mask>	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> 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 physical layer information for 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 Security 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 Encapsulation and Tunnels Reference Guide.*
RIP	Commands for configuring the Routing Information Protocol on Ethernet interfaces are described in the *Vyatta RIP Reference Guide*.
QoS	Commands for configuring qulaity of service on Ethernet interfaces are described in the *Vyatta Policy and QoS 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. The range is **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.

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 eth0..eth23 {
   }
}
```

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. There will be as many Ethernet interface configuration nodes created as there are physical Ethernet interfaces 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** {*ipv4net* | *ipv6net* | **dhcp**}

delete interfaces ethernet *ethx* **address** {*ipv4net* | *ipv6net* | **dhcp**}

show interfaces ethernet *ethx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      address [ipv4net|ipv6net|dhcp]
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
ipv4net	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.
ipv6net	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.

Default

None.

Usage Guidelines

Use this command to set the IP address and network prefix for an 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 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 eth0..eth23 {
      bond-group bond0..bond99
   }
}
```

Parameters

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

Default

None.

Usage Guidelines

Use this command to add an Ethernet interface to an Ethernet link bonding group. An Ethernet interface can only be a member of one Ethernet link bonding interface and the bonding interface must first be defined using the **interfaces bonding <bondx>** command (see page 119). 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.

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

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

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

interfaces ethernet <ethx> description <descr>

Specifies a description for an Ethernet interface.

Syntax

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

delete interfaces ethernet *ethx* **description**

show interfaces ethernet *ethx* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
         description text
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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> 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 eth0..eth23 {
         disable
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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 eth0..eth23 {
      disable-link-detect
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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 eth0..eth23 {
      duplex [auto|half|full]
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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.

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 eth0..eth23 {
      hw-id mac-addr
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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 the Ethernet interface.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      ip {
         enable-proxy-arp
      }
   }
}
```

Parameters

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

Default

Proxy ARP is disabled.

Usage Guidelines

Proxy ARP is a technique by which the system will answer ARP queries for a network address that it does not have configured on the receiving interface. Traffic that is sent to the network address will be routed to the host via another interface.

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

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> 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 eth0..eth23 {
        mac mac-addr
    }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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> 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 eth0..eth23 {
      mtu u32
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. 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> smp_affinity <mask>

Sets the SMP affinity for an Ethernet interface.

Syntax

set interfaces ethernet *ethx* **smp_affinity** *mask*

delete interfaces ethernet *ethx* **smp_affinity** *mask*

show interfaces ethernet *ethx* **smp_affinity**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      smp_affinity hex-mask
   }
}
```

Parameters

ethx	Multi-node. An identifier for the Ethernet interface you are defining. The range is **eth0** to **eth23**.
mask	Multi-node. Up to four hex digits that identify the processor(s) that this interface will interrupt; for example, 0x0001 represents CPU 0 and 0x0080 represents CPU 7. You can distribute the interrupts from an interface among multiple processors by creating multiple **smp_affinity** configuration nodes.

Default

Interrupts are serviced by any available CPU.

Usage Guidelines

Use this command to configure and display the SMP affinity mask 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 can be recruited to process any task. By default, hardware interrupts are processed by whichever processor is available. Setting SMP affinity 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.

Optimal performance generally achieved when each interface interrupts only one processor interrupts from the most heavily loaded interfaces are evenly distributed among available CPUs.

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 eth0..eth23 {
      speed [auto|10|100|1000]
   }
}
```

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 router 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.

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 1-1 shows information for all Ethernet interfaces.

Example 1-1 Displaying information for all Ethernet interfaces

```
root@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 1-2 shows information for interface eth2.

Example 1-2 Displaying information for one Ethernet interface

```
root@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 1-3 shows the first screen of output for **show interfaces ethernet detail**.

Example 1-3 Displaying detailed Ethernet interface information

```
root@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 specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.

Default

None.

Usage Guidelines

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

Examples

Example 1-4 shows brief status for interface eth2.

Example 1-4 Displaying brief Ethernet interface status

```
root@vyatta> show interfaces ethernet eth2 brief
Interface    IP Address        State       Link   Description
eth2         10.1.0.66/24      up          up
```

show interfaces ethernet <ethx> identify

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

Syntax

show interfaces ethernet *ethx* **identify**

Command Mode

Operational mode.

Parameters

ethx	The specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.

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 1-5 shows the output for **show interfaces ethernet** *ethx* **identify**.

Example 1-5 Identifying an Ethernet interface by blinking its LED

```
root@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 specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.

Default

None.

Usage Guidelines

Use this command to view physical layer command and operational status of Ethernet interfaces.

Examples

Example 1-6 shows output for **show interfaces ethernet** *ethx* **physical**.

Example 1-6 Displaying physical line characteristics for an Ethernet interface.

```
root@vyatta> show interfaces ethernet eth0 physical
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: pumbg
        Wake-on: d
        Current message level: 0x00000001 (1)
        Link detected: yes
root@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 specified Ethernet interface. This may be **eth0** to **eth23**, depending on what Ethernet interfaces that are actually available on the system.
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 1-7 shows queue information for interface eth0.

Example 1-7 Displaying Ethernet queue information

```
root@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
```

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

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.

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

Use this command to define the loopback interface.

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

The loopback interface provides a number of advantages:

- As long as the router is functioning, the loopback interface is always up, and so is very reliable. As long as there is even one functioning link to the router, the loopback interface can be accessed. The loopback interface thus eliminates the need to try each IP address of the router until you find one that is still up.
- Because the loopback interface is always up, a routing session (such as a BGP session) can continue even if the outbound interface fails.

- You can simplify collection of management information by specifying the loopback interface as the interface for sending and receiving management information such as logs and SNMP traps.
- The loopback interface can be used 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.

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 {*ipv4net* | **dhcp**}

delete interfaces loopback lo address {*ipv4net* | **dhcp**}

show interfaces loopback lo address

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   loopback lo {
      address: [ipv4net|dhcp]
   }
}
```

Parameters

ipv4net	The IPv4 address and network prefix for this vif. 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.
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

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

- The router's hostname should be mapped to the loopback interface address, rather than 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.

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: text
   }
}
```

Parameters

descr	The 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.

```
root@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.

```
root@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

```
root@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.

```
root@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

VLAN Interface Commands

This chapter contains the following commands.

Configuration Commands	
Vifs on Ethernet Interfaces	
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.
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.
Operational Commands	
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 4.
show interfaces ethernet detail	Displays detailed information about Ethernet interfaces. See page 30

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 Security 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 Encapsulation and Tunnels 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 Policy and 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 bond0..bond99 {
      vif 0-4094 {
      }
   }
}
```

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** {*ipv4net* | *ipv6net* | **dhcp**}

delete interfaces bonding *bondx* **vif** *vlan-id* **address** {*ipv4net* | *ipv6net* | **dhcp**}

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
      vif 0-4094 {
         address [ipv4net|ipv6net|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.
ipv4net	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.
ipv6net	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 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 bond0..bond99 {
      vif 0-4094 {
         description text
      }
   }
}
```

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 bond0..bond99 {
      vif 0-4094 {
         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 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 eth0..eth23 {
      vif 0-4094 {
      }
   }
}
```

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** {*ipv4net* | *ipv6net* | **dhcp**}

delete interfaces ethernet *ethx* **vif** *vlan-id* **address** {*ipv4net* | *ipv6net* | **dhcp**}

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   ethernet eth0..eth23 {
      vif 0-4094 {
         address [ipv4net|ipv6net|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.
ipv4net	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.
ipv6net	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 eth0..eth23 {
      vif 0-4094 {
         description: text
      }
   }
}
```

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 eth0..eth23 {
      vif 0-4094 {
         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 eth0..eth23 {
        vif 0-4094 {
            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.

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 3-1 shows information for vif 11 on interface eth0

Example 3-1 Displaying Ethernet vif information

```
root@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
root@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 3-2 shows brief status for interface eth2.6.

Example 3-2 Displaying brief status for a vif.

```
root@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 3-3 shows queue information for interface eth0.6.

Example 3-3 Displaying VLAN interface queue information

```
root@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 Commands

Bridging Commands

This chapter contains the following commands.

Configuration Commands	
Bridge Groups	
interfaces bonding <bondx> bridge-group <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 <state>	Enables or disables a bridge group without discarding configuration.
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 <brx>	Assigns an Ethernet link bonding interface to a bridge group.
Ethernet Link Bonding Interface Vifs	
interfaces bonding <bondx> vif <vlan-id> bridge-group <brx>	Assigns an Ethernet link bonding interface vif to 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 28.
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 **eth***x***.vif***y*).

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 <brx>

Assigns an Ethernet link bonding interface to a bridge group.

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
         bridge-group br0..br9
   }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
brx	The bridge group to which you are adding the bonding interface. Supported values are **br0** through **br9**.

Default

None.

Usage Guidelines

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

A bonding interface can only be a member of one bridge group and the bridge group must already be defined.

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

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

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

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

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

Syntax

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

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

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

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
      vif 0-4094 {
         bridge-group brx
      }
   }
}
```

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.
brx	The identifier of the bridge group you are adding the vif to. Supported values are **br0** through **br9**.

Default

None.

Usage Guidelines

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

A vif on an Ethernet bonding link interface can only be a member of one bridge group and the bridge group must first be defined.

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 vif from the bridge group.

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

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 br0..br9 {
   }
}
```

Parameters

brx	Multi-node. The identifier for the bridge group. Supported identifiers are **br0** through **br9**. 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 br0..br9 {
      address: text
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      aging: u32
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      description: text
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 <state>

Enables or disables a bridge group without discarding configuration.

Syntax

set interfaces bridge *brx* **disable** *state*

delete interfaces bridge *brx* **disable**

show interfaces bridge *brx* **disable**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bridge br0..br9 {
      disable: [true|false]
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
state	Enables or disables bridging on this bridge group. Supported values are as follows: **true**—Disables this bridge group without discarding configuration. **false**—Enables this bridge group. The default is **false**.

Default

Bridging is disabled.

Usage Guidelines

Use this command to enable or disable a bridge group.

Use the **set** form of this command to specify whether or not bridging is enabled 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> 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 br0..br9 {
      forwarding-delay: u32
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      hello-time: u32
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      max-age: u32
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      priority: u32
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 br0..br9 {
      stp: [true|false]
   }
}
```

Parameters

brx	The identifier for the bridge group. Supported identifiers are **br0** through **br9**.
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 eth0..eth23 {
      bridge-group {
         bridge br0..br9
      }
   }
}
```

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 **br9**.

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 eth0..eth23 {
      bridge-group {
         cost: [0-2147483647]
      }
   }
}
```

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 eth0..eth23 {
      bridge-group {
         priority: [0-255]
      }
   }
}
```

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 eth0..eth23 {
      vif 0-4095 {
         bridge-group {
            bridge: br0..br9
         }
      }
   }
}
```

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 **br9**.

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 eth0..eth23 {
      vif 0-4095 {
         bridge-group {
            cost: [0-2147483647]
         }
      }
   }
}
```

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 eth0..eth23 {
      vif 0-4095 {
         bridge-group {
            priority 0-255
         }
      }
   }
}
```

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 srange 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.

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 **br9**.
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 **br9**.
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
- Ethernet Link Bonding Commands

Ethernet Link Bonding Configuration

This section presents the following topics:

- Ethernet Link Bonding Overview
- Ethernet Bonding Configuration Example

Ethernet Link Bonding Overview

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

Many implementations of Ethernet link bonding have been non-standard. The IEEE 802.3ad specification 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, within each Ethernet interfaces that you wish to add as member links of the bonded link, specify a “bond-group” that points to the bonding interface you created. Repeat this for all Ethernet interfaces to be part of the bonded link.

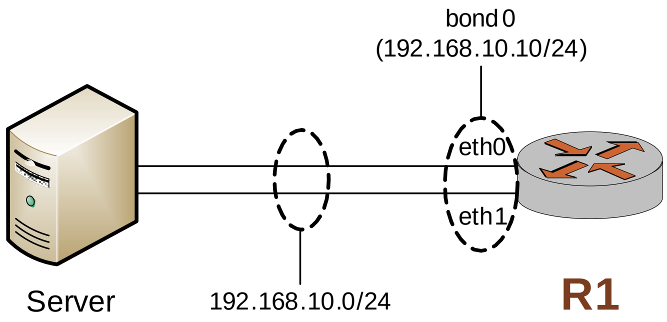
Figure 5-1 shows a simple Ethernet link bonding scenario with Ethernet link bonding interface consisting of two physical Ethernet links. In this example:

- The bond group bond0 is created using the default bonding mode (802.3ad).
- Interfaces eth0 and eth1 are both added to bond0.

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

Use the **show interfaces** and **show interface 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** [edit]
Set the IP address of the bonding group.	vyatta@R1# **set interfaces bonding bond0 address 192.168.10.10/24** [edit]
Set the bonding mode of the bonding group.	vyatta@R1# **set interfaces bonding bond0 mode 802.3ad** [edit]
Add eth0 to the bond0 bonding group.	vyatta@R1# **set interfaces ethernet eth0 bonding-group bond0** [edit]
Add eth1 to the bond0 bonding group.	vyatta@R1# **set interfaces ethernet eth1 bonding-group bond0** [edit]
Commit the change.	vyatta@R1# **commit** [edit]
Show the bonding group configuration.	vyatta@R1# **show interfaces bonding bond0** address 192.168.10.10/24 [edit]
Show the eth0 configuration.	vyatta@R1# **show interfaces ethernet eth0** bonding-group bond0 [edit]
Show the eth1 configuration.	vyatta@R1# **show interfaces ethernet eth1** bonding-group bond0 [edit]

Ethernet Link Bonding Commands

This chapter contains the following commands.

Configuration Commands	
interfaces bonding <bondx>	Defines the characteristics of an Ethernet link bonding interface.
interfaces bonding <bondx> address	Assigns a network address to an Ethernet link bonding interface.
interfaces bonding <bondx> description <desc>	Specifies a description for an Ethernet link bonding interface.
interfaces bonding <bondx> disable	Disables an Ethernet link bonding interface without discarding configuration.
interfaces bonding <bondx> mac <mac-addr>	Sets the MAC address of an Ethernet link bonding interface.
interfaces bonding <bondx> mode	Sets the bonding mode for an Ethernet link bonding interface.
interfaces bonding <bondx> mtu <mtu>	Specifies the MTU for an Ethernet link bonding interface.
interfaces bonding <bondx> primary <ethx>	Specifies the primary Ethernet interface within the Ethernet link bonding interface.
Operational Commands	
show interfaces bonding	Shows Ethernet link bonding interface information.

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

Related Commands Documented Elsewhere	
Firewall	Commands for configuring firewall on bonded Ethernet links are described in the *Vyatta Security Reference Guide*.
VLAN Interfaces	Commands for defining VLAN interfaces (vifs) on bonded Ethernet links are described in “Chapter 3: VLAN Interfaces.”
QoS	Commands for configuring quality of service on bonded Ethernet links are described in the *Vyatta Policy and QoS Reference Guide*.
Bridging	Commands for adding Ethernet bonding link interfaces are described in “Chapter 4: Bridging.”

interfaces bonding <bondx>

Defines the characteristics of an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx*

delete interfaces bonding *bondx*

show interfaces bonding *bondx*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
   }
}
```

Parameters

bondx	Multi-node. The identifier of the bonding interface you are defining. Supported values are **bond0** through **bond99**.

Default

None.

Usage Guidelines

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

Note that you must create the bonding interface (using this command) before you can assign Ethernet interfaces to it.

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

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

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

interfaces bonding <bondx> address

Assigns a network address to an Ethernet link bonding interface.

Syntax

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

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

show interfaces bonding *bondx* **address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond99 {
      address [ipv4net|dhcp]
   }
}
```

Parameters

bondx	Multi-node. The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
ipv4net	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.
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 an Ethernet link bonding 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 bonding <bondx> description <desc>

Specifies a description for an Ethernet link bonding interface.

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 bond0..bond99 {
      description text
   }
}
```

Parameters

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

Default

None.

Usage Guidelines

Use this command to specify a description for the bonding interface.

Use the **set** form of this command to specify a description for the bonding interface.

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 bonding interface 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 bond0..bond99 {
      disable
   }
}
```

Parameters

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

Default

None.

Usage Guidelines

Use this command to disable 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 the configuration.

interfaces bonding <bondx> mac <mac-addr>

Sets the MAC address of an Ethernet link bonding interface.

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 bond0..bond23 {
        mac mac-addr
    }
}
```

Parameters

bondx	The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
mac-addr	The MAC address for the Ethernet link bonding interface. The format should be appropriate for the interface type. For an Ethernet interface, this is 6 colon-separated 8-bit numbers in hexadecimal; for example, 00:0a:59:9a:f2:ba.

Default

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 interface.

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.

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

interfaces bonding <bondx> mode

Sets the bonding mode for an Ethernet link bonding interface.

Syntax

set interfaces bonding *bondx* **mode** {**802.3ad** | **active-backup** | **balance-alb** | **balance-rr** | **balance-tlb** | **balance-xor** | **broadcast**}

delete interfaces bonding *bondx* **mode**

show interfaces bonding *bondx* **mode**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces {
   bonding bond0..bond23 {
      mode [802.3ad | active-backup | balance-alb | balance-rr |
         balance-tlb | balance-xor | broadcast]
   }
}
```

Parameters

bondx	The identifier for the bonding interface. 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.
balance-alb	Uses adaptive load balancing as the bonding mode. This mode includes both adaptive transmit load balancing plus receive load balancing, and does not require any special switch support. The receive load balancing is achieved by ARP negotiation.

balance-rr	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.
balance-tlb	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.
balance-xor	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 bonding interface.

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

Use the **delete** form of this command to remove the bonding mode for the interface.

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

interfaces bonding <bondx> mtu <mtu>

Specifies the MTU for an Ethernet link bonding interface.

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 bond0..bond99 {
      mtu u32
   }
}
```

Parameters

bondx	The identifier for the bonding interface. 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 underlying device.

Usage Guidelines

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

Note that changing the MTU will change the MTU on the Ethernet links within the bond. Also, the MTU of the Ethernet links within the bond 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 bonding <bondx> primary <ethx>

Specifies the primary Ethernet interface within the Ethernet link bonding interface.

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 bond0..bond99 {
      primary ethx
   }
}
```

Parameters

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

Default

None.

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 slave is preferred over another; for example, when one slave 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.

show interfaces bonding

Shows Ethernet link bonding interface information.

Syntax

show interfaces bonding

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display information about configured Ethernet link bonding interfaces.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
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
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
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
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus

TCP	Transmission Control 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