VYATTA, INC. | Vyatta System

High Availability

REFERENCE GUIDE

WAN Load Balancing
VRRP
Clustering
RAID 1

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: April 2010

DOCUMENT REVISION. R6.0 v03

RELEASED WITH: R6.0

PART NO. A0-0221-10-0007

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 and appendixes:

Chapter	Description	Page
Chapter 1: WAN Load Balancing	This chapter describes how to use the wide area network (WAN) load balancing feature of the Vyatta system.	1
Chapter 2: VRRP	This chapter explains how to use Virtual Router Redundancy Protocol (VRRP) on the Vyatta system.	68
Chapter 3: Clustering	This chapter explains clustering for high availability on the Vyatta system.	137
Chapter 4: RAID 1	This chapter describes how to set up hard drives in a Redundant Array of Independent Disks (RAID) 1 deployment using the Vyatta system.	180
Glossary of Acronyms		198

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 *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 *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:

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs ("+"), as in <Ctrl>+c.
[*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, inclusive.
arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[*arg*...] *arg*[,*arg*...]	A value that can optionally represent a list of elements (a space-separated list 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 *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Chapter 1: WAN Load Balancing

This chapter describes how to use the wide area network (WAN) load balancing feature of the Vyatta system.

This chapter presents the following topics:

- WAN Load Balancing Configuration
- WAN Load Balancing Commands

WAN Load Balancing Configuration

This section describes how to configure WAN load balancing on the Vyatta system.

This section presents the following topics:

- WAN Load Balancing Overview
- Configuration Examples

WAN Load Balancing Overview

This section presents the following topics:

- What Is Load Balancing
- Balancing Algorithm
- Load Balancing Rules
- Flow-Based vs. Packet-Based Balancing
- Health Checking
- Failover
- Excluding Traffic from WAN Load Balancing
- Rate Limiting
- Script Execution on Interface State Changes
- Steps for Configuring WAN Load Balancing

What Is Load Balancing

The Vyatta system supports automatic load balancing for outbound traffic across two or more outbound interfaces. In addition to balancing traffic loads across interfaces, this feature also provides for path redundancy should a path fail, as traffic will be balanced across the remaining healthy paths. When the failed path recovers it will be added to the list of healthy paths so that the load balancing system can use it again. The system determines path health through periodic health checking to a remote target or targets.

Load balancing is supported for outbound traffic only. Load balancing is not performed on packets sourced from the system itself unless it is explicitly confgured to do so (using e the **load-balancing wan enable-local-traffic** command (see page 26)).

For load balancing to occur, at least two paths need to be available in the routing table, and these paths must egress through the interfaces being load balanced. The WAN load balancing process automatically installs the default routes you configure for each path, and balances traffic according to path health and the weights you apply to each interface. You can see which paths are installed in the routing table using the **show ip route** command.

Balancing Algorithm

Outbound packets are load balanced using a weighted random distribution load balancing algorithm. If no weights are assigned, each interface has an equal chance of being picked, which, on average, results in each interface receiving approximately the same number of packets. If an interface has a higher weight, it will tend to be picked more often; for example, if interface A has a weight of 2 and interface B has a weight of 1, interface A will tend to be picked 67% of the time.

Load Balancing Rules

The kind of traffic to be balanced, the set of interfaces, and the relative weight for each interface is specified in a load balancing rule. A load balancing rule contains a set of match criteria and a set of interfaces with weights attached. Outgoing packets are matched against the criteria specified in the rule. If the packet is a match for the rule, the load balancing algorithm determines to which interface in the specified set the packet is sent.

Rules are executed in numeric order until a successful match is achieved. If a match is achieved, the packet is sent to one of the interfaces specified by the rule, unless none of the interfaces is active. In this case, the next rule is executed until a matching rule has at least one active interface. If no rules are matched, then the main system routing table is used.

Once configured, rule numbers cannot be changed. For this reason, it is good practice to configure rules at intervals (for example, rule 5, rule 10, rule 15, and so on) in case a rule must be inserted later on.

Flow-Based vs. Packet-Based Balancing

In general, traffic is load balanced on a per-flow basis, not on a per-packet basis. Any connection-oriented traffic remains appropriately associated with the interface assigned for load balancing. Flows are tracked by means of an entry in the connection tracking table. Flows are identified by a tuple consisting of source address, destination address, and port.

If source-based routing is enabled, traffic is load balanced on a per-packet basis and the source address is considered when choosing the egress interface. Also, if the connection tracking table is flushed, previously established flows are balanced on a per-packet basis until a new connection is established on the flow.

Health Checking

A load balanced WAN interface is considered an active member of the interface pool so long as it passes health checks. The health of the interface is monitored by having it send an ICMP Echo Request (“ping”) message at intervals to some remote destination.

Successful receipt of the ICMP Echo Reply message from the destination shows that the interface can both transmit to the Internet and receive packets from the Internet. If the interface fails the health check, it is removed from the pool of active interfaces.

NOTE *A time-to-live (ttl) test is also available which sends a udp packet with a ttl limit to the target.*

For each interface to be load balanced, the interface health criteria must be configured, including the number of missed health checks that cause an interface to be declared unhealthy and the successes required to declare its health restored. Configuring more than one target for health check tests means that one does not have to rely on a single target which might be non-responsive for reasons other than path failure. Multi-targets will be tested until a test is successful or the list of tests is exhausted.

Health check configuration consists of the following:

- The remote destination to be tested for accessibility. Use the **load-balancing wan interface-health <if-name> test <test-no> target <address>** command (see page 39).
- The nexthop on the path to the target destination. Use the **load-balancing wan interface-health <if-name> nexthop <ipv4>** command (see page 33)
- The type of test to perform (either **ping** or **ttl**). Use the **load-balancing wan interface-health <if-name> test <test-no> type <type>** command (see page 43)
- The maximum response time to the ping message that can be considered a success. Use the **load-balancing wan interface-health <if-name> test <test-no> resp-time <seconds>** command (see page 37). Or the ttl limit for **ttl** tests. Use the **load-balancing wan interface-health <if-name> test <test-no> ttl-limit <limit>** command (see page 41)
- The number of health check failures that can occur before the interface is considered unavailable. Use the **load-balancing wan interface-health <if-name> failure-count <num>** command (see page 32).
- The number of successful health checks that must occur before the interface can be added back into the pool of active interfaces. Use the **load-balancing wan interface-health <if-name> success-count <num>** command (see page 45).

Implicit NAT

One potential issue with multiple external interfaces is that return traffic can enter through a different interface than it was sent. This is referred to as asymmetric routing and, in general, it is undesirable for a number of reasons, including a potential load imbalance on incoming traffic and difficulties troubleshooting. In order to prevent asymmetric routing, the WAN load balancer replaces the source address of all IP packets egressing through an interface with the address of that interface (i.e. Masquerade NAT) so that reply traffic will return through the same interface that it was sent on. If this default behavior is not desirable it can be disabled using the the **load-balancing wan disable-source-nat** command (see page 24).

Failover

Normally, all balanced interfaces are utilized to balance traffic. However, the Vyatta system supports a “failover” mode for WAN load balancing (using the **load-balancing wan rule <rule> failover** command (see page 54)). When in “failover” mode, one interface is selected by the system as the primary and the remaining interfaces are designated secondary or spare.

The primary interface is selected based on its configured weight and the reachability of the target. If connectivity via the primary interface is interrupted, the next secondary interface is selected from the set of secondary interfaces.

In addition to weight-based primary interface selection, the primary interface can be selected based on rule order. Because traffic is directed out the first healthy interface specified in a matching rule, rules can be ordered based on exgress interface preference. For example, if eth0 is to be used as the primary interface unless it becomes unhealthy in which case eth1 is to be used, then eth0 would be included in the first rule and eth1 in the following rule.

When a link fails over and a new primary interface is selected, existing sessions do not automatically failover to the new path. The end user will experience a session timeout. To avoid this delay in session failure, the session table can be flushed if there is a link state change using the **load-balancing wan flush-connections** command (see page 27).

Excluding Traffic from WAN Load Balancing

There are cases where it is not desirable to load balance all traffic sourced from a specified interface. For example, in a scenario where there are two local LANs whose upstream traffic is to be load balanced, traffic from one LAN to the other will also be routed upstream - which is not desired. To avoid this, the intra-LAN traffic needs to “excluded” from being load balanced. Traffic exclusion is configured using the **load-balancing wan rule <rule> exclude** command (see page 53).

Rate Limiting

The rate limit feature sets a rate limit at which a rule will be active. For example, it may be desirable to configure a system to send all packets out one interface up to a given limit. All packets in excess of that limit would be sent out a second interface. This scenario would require two rules. The first rule, with a rate limit specified, sends packets out interface 1. The second rule, with no limits specified, sends packets out interface 2. Rate limiting is configured using the **load-balancing wan rule <rule> limit** command (see page 58).

Script Execution on Interface State Changes

To provide flexibility in actions that can be taken on an interface state change the “hook” feature is provided. This feature allows a script to be run when an interface state changes from acive to failed or failed to active. Script execution on interface state changes is configured using the **load-balancing wan hook <script-name>** command (see page 29).

Steps for Configuring WAN Load Balancing

There are three steps for setting up WAN load balancing:

1. Define a target (or targets) for each interface being load balanced that is (are) accessible to the interface. The target is used by the load balancing service to determine the health of the interface.
2. Configure a nexthop address for each target, for each interface to be load balanced. The load balancing service uses this address to access the target.
3. Configure one static host route entry for each target to provide routing for accessing the target via the desired interface.

Configuration Examples

This section presents the following topics:

- Basic WAN Load Balancing
- Failover using Interface Weights
- Failover using Rule Order
- Failover using Rule Order - Priority Traffic
- Excluding Traffic from Load Balancing

Basic WAN Load Balancing

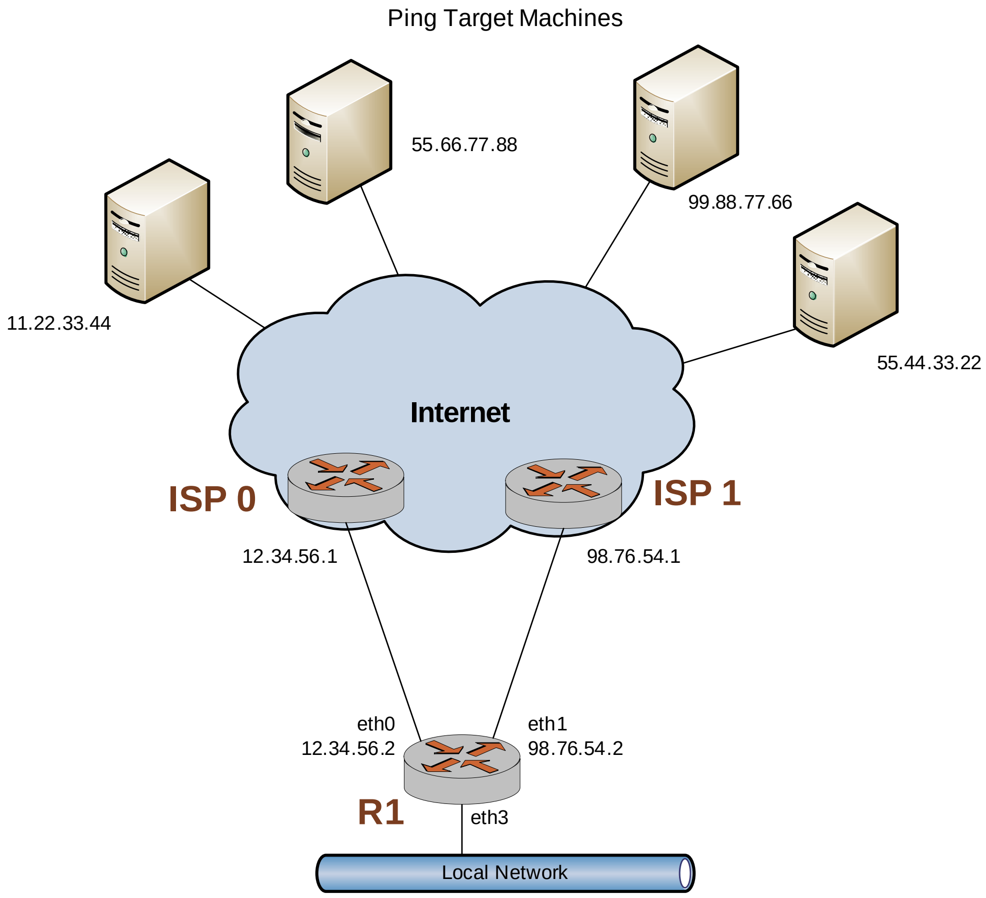
In this example, a sample configuration is presented for a basic WAN load balancing scenario. In this configuration:

- All traffic incoming through interface eth3 is balanced between interfaces eth0 and eth1 on R1.
- The outgoing interfaces eth0 and eth1 are tested for interface health by pinging remote ping targets via these interfaces. The remote targets in this example are: 11.22.33.44, 55.66.77.88, 99.88.77.66, and 55.44.33.22.
- Outgoing packets are assigned the primary source address of the assigned interface.

- Interface eth1 would be removed from the active pool after four consecutive ping failures and interface eth0 after five consecutive failures.

When you are finished, R1 will be configured as shown in Figure 1-1.

Figure 1-1 WAN load balancing



This section includes the following examples:

- Example 1-1 Creating static routes to ping targets
- Example 1-2 Creating load balancing configuration

Example 1-1 creates static routes directed towards the two ISPs that the load will be balanced between: 12.34.56.1 and 98.76.54.1. To create these static routes, perform the following steps in configuration mode:

Example 1-1 Creating static routes to ping targets

Step	Command

Create a static route to a ping target for testing the health of eth0.	`vyatta@R1#` **`set protocols static route 11.22.33.44/32 next-hop 12.34.56.1`**
	`[edit]`

Create a static route to a second ping target for testing the health of eth0.	`vyatta@R1#` **`set protocols static route 55.66.77.88/32 next-hop 12.34.56.1`**
	`[edit]`

Create a static route to a ping target for testing the health of eth1.	`vyatta@R1#` **`set protocols static route 99.88.77.66/32 next-hop 98.76.54.1`**
	`[edit]`

Create a static route to a second ping target for testing the health of eth1.	`vyatta@R1#` **`set protocols static route 55.44.33.22/32 next-hop 98.76.54.1`**
	`[edit]`

Commit the configuration.	`vyatta@R1#` **`commit`**
	`[edit]`

Display the configuration	`vyatta@R1#` **`show protocols static`**
	` route 11.22.33.44/32 {`
	`      next-hop 12.34.56.1 {`
	`      }`
	` }`
	` route 55.66.77.88/32 {`
	`      next-hop 12.34.56.1 {`
	`      }`
	` }`
	` route 99.88.77.66/32 {`
	`      next-hop 98.76.54.1 {`
	`      }`
	` }`
	` route 55.44.33.22/32 {`
	`      next-hop 98.76.54.1 {`
	`      }`
	` }`
	`[edit]`

Example 1-2 sets up a basic WAN load balancing configuration on R1. To create the load balancing configuration, perform the following steps in configuration mode:

Example 1-2 Creating load balancing configuration

Step	Command
Set the failure count for eth0.	vyatta@R1# **set load-balancing wan interface-health eth0 failure-count 5** [edit]
Set the next hop for eth0.	vyatta@R1# **set load-balancing wan interface-health eth0 nexthop 12.34.56.1** [edit]
Set the test type for eth0.	vyatta@R1# **set load-balancing wan interface-health eth0 test 10 type ping** [edit]
Set a ping target for eth0.	vyatta@R1# **set load-balancing wan interface-health eth0 test 10 target 11.22.33.44** [edit]
Set a second ping target for eth0.	vyatta@R1# **set load-balancing wan interface-health eth0 test 10 target 55.66.77.88** [edit]
Set the failure count for eth1.	vyatta@R1# **set load-balancing wan interface-health eth1 failure-count 4** [edit]
Set the nexthop for eth1.	vyatta@R1# **set load-balancing wan interface-health eth1 nexthop 98.76.54.1** [edit]
Set the test type for eth1.	vyatta@R1# **set load-balancing wan interface-health eth1 test 10 type ping** [edit]
Set a ping target for eth1.	vyatta@R1# **set load-balancing wan interface-health eth1 test 10 target 99.88.77.66** [edit]
Set a second ping target for eth1.	vyatta@R1# **set load-balancing wan interface-health eth1 test 10 target 55.44.33.22** [edit]
Define eth3 as the inbound interface.	vyatta@R1# **set load-balancing wan rule 10 inbound-interface eth3** [edit]
Define eth0 as one of the interfaces to be load balanced.	vyatta@R1# **set load-balancing wan rule 10 interface eth0** [edit]

Example 1-2 Creating load balancing configuration

Define eth1 as another interface to be load balanced.

```
vyatta@R1# set load-balancing wan rule 10 interface eth1
[edit]
```

Commit the configuration.

```
vyatta@R1# commit
[edit]
```

Display the configuration

```
vyatta@R1# show load-balancing
wan {
      interface-health eth0 {
         failure-count 5
         nexthop 12.34.56.1
         test 10 {
            target 11.22.33.44
            target 55.66.77.88
            type ping
         }
      }
      interface-health eth1 {
         failure-count 4
         nexthop 98.76.54.1
         test 10 {
            target 99.88.77.66
            target 55.44.33.22
            type ping
         }
      }
      rule 10 {
         inbound-interface eth3
         interface eth0 {
         }
         interface eth1 {
         }
      }
}
[edit]
```

Failover using Interface Weights

In the previous example the system was configured to balance the traffic load between eth0 and eth1. In this example, instead of balancing the traffic, eth0 will be configured as the primary interface and eth1 as a backup interface to be used if eth0 fails. The only change will be to Rule 10.

In this configuration:

- All traffic incoming through interface eth3 is sent out interface eth0.
- If eth0 fails, all traffic incoming through interface eth3 will be sent out eth1.

Example 1-3 sets up a failover configuration rule on R1. To modify the previous example to create the failover configuration using interface weights, perform the following steps in configuration mode:

Example 1-3 Creating failover configuration using interface weights

Step	Command

Remove the existing Rule 10 from the previous example.	`vyatta@R1#` **`delete load-balancing wan rule 10`**
	`[edit]`

Enable failover mode.	`vyatta@R1#` **`set load-balancing wan rule 10 failover`**
	`[edit]`

Define eth3 as the inbound interface.	`vyatta@R1#` **`set load-balancing wan rule 10 inbound-interface eth3`**
	`[edit]`

Define eth0 as the primary interface as it will have the largest weight value.	`vyatta@R1#` **`set load-balancing wan rule 10 interface eth0 weight 10`**
	`[edit]`

Define eth1 as the secondary interface as it will have the smallest weight value.	`vyatta@R1#` **`set load-balancing wan rule 10 interface eth1 weight 1`**
	`[edit]`

Commit the configuration.	`vyatta@R1#` **`commit`**
	`[edit]`

Example 1-3 Creating failover configuration using interface weights

Display the configuration

```
vyatta@R1# show load-balancing
wan {
      interface-health eth0 {
         failure-count 5
         nexthop 12.34.56.1
         test 10 {
            target 11.22.33.44
            target 55.66.77.88
            type ping
         }
      }
      interface-health eth1 {
         failure-count 4
         nexthop 98.76.54.1
         test 10 {
            target 99.88.77.66
            target 55.44.33.22
            type ping
         }
      }
      rule 10 {
         failover
         inbound-interface eth3
         interface eth0 {
            weight 10
         }
         interface eth1 {
            weight 1
         }
      }
}
[edit]
```

Failover using Rule Order

In the previous example the system was configured to send all incoming traffic from eth3 out eth0 unless the health check on eth0 failed. If the health check on eth0 failed all incoming traffic from eth3 would be sent out eth1. This was accomplished using the “failover” command and assigning different weights to the two outbound interfaces. In this example, failover is accomplished based on rule order. Again, all traffic from eth3 will be sent out eth0 unless its health check fails, at which point the traffic will be sent out eth1. This is accomplished using two rules. One rule directs all trffic from eth3 to eth0 and the other directs all traffic from eth3 to eth1. When eth0 is healthy all traffic from eth3 will match the first rule and be sent out eth0. If eth0 fails, the first rule will be bypassed due to path health and the second rule will take effect - directing all traffic from eth3 to eth1. Once eth0 becomes healthy, traffic will again match the first rule and will be sent out eth0.

Example 1-4 sets up failover configuration rules on R1. To modify the previous example to create the failover configuration using rule order, perform the following steps in configuration mode:

Example 1-4 Creating failover configuration using rule order

Step	Command
Remove the existing Rule 10 from the previous example.	vyatta@R1# **delete load-balancing wan rule 10** [edit]
Define eth3 as the inbound interface for this rule.	vyatta@R1# **set load-balancing wan rule 10 inbound-interface eth3** [edit]
Define eth0 as the primary egress interface.	vyatta@R1# **set load-balancing wan rule 10 interface eth0** [edit]
Define eth3 as the inbound interface for this rule.	vyatta@R1# **set load-balancing wan rule 20 inbound-interface eth3** [edit]
Define eth1 as the secondary egress interface.	vyatta@R1# **set load-balancing wan rule 20 interface eth1** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Example 1-4 Creating failover configuration using rule order

Display the configuration

```
vyatta@R1# show load-balancing
wan {
      interface-health eth0 {
         failure-count 5
         nexthop 12.34.56.1
         test 10 {
            target 11.22.33.44
            target 55.66.77.88
            type ping
         }
      }
      interface-health eth1 {
         failure-count 4
         nexthop 98.76.54.1
         test 10 {
            target 99.88.77.66
            target 55.44.33.22
            type ping
         }
      }
      rule 10 {
         inbound-interface eth3
         interface eth0 {
         }
      }
      rule 20 {
         inbound-interface eth3
         interface eth1 {
         }
      }
}
[edit]
```

Failover using Rule Order - Priority Traffic

One of the advantages of using rule order for failover configuration is the additional flexibility it provides. For example, in situations where the backup link is a lower speed link than the primary and is only able to pass priority traffic, a rule order configuration is required.

In the previous example the system was configured to send all incoming traffic from eth3 out eth0 unless the health check on eth0 failed. If the health check on eth0 failed all incoming traffic from eth3 would be sent out eth1. This was accomplished using rule order.

In this example, we assume that the backup link is lower speed and unable to accomodate all traffic, so we select only high priority traffic (in this case Voice over IP traffic) to send out the backup link if eth0 fails the health check.

Example 1-5 sets up failover configuration rules on R1. To modify the previous example to create the failover configuration using rule order where only a subset of traffic is sent out the secondary interface when the primary interface fails a health check, perform the following steps in configuration mode:

Example 1-5 Creating failover configuration using rule order and a lower speed backup link

Step	Command

Remove the existing Rule 20 from the previous example.	`vyatta@R1# delete load-balancing wan rule 20`
	`[edit]`

Define eth3 as the inbound interface for this rule.	`vyatta@R1# set load-balancing wan rule 20 inbound-interface eth3`
	`[edit]`

Define eth1 as the secondary egress interface.	`vyatta@R1# set load-balancing wan rule 20 interface eth1`
	`[edit]`

Identify the traffic to match by port (VoIP traffic is on the sip port) ...	`vyatta@R1# set load-balancing wan rule 20 destination port sip`
	`[edit]`

... and protocol.	`vyatta@R1# set load-balancing wan rule 20 protocol tcp`
	`[edit]`

Allow traffic that falls through the load-balancing rules to try to exit via eth0 as the health check failure may be due to issues with the health check target rather than the local link.	`vyatta@R1# set protocols static route 0.0.0.0/0 next-hop 12.34.56.1`
	`[edit]`

Commit the configuration.	`vyatta@R1# commit`
	`[edit]`

Example 1-5 Creating failover configuration using rule order and a lower speed backup link

Display the load-balancing configuration

```
vyatta@R1# show load-balancing
wan {
      interface-health eth0 {
         failure-count 5
         nexthop 12.34.56.1
         test 10 {
            target 11.22.33.44
            target 55.66.77.88
            type ping
         }
      }
      interface-health eth1 {
         failure-count 4
         nexthop 98.76.54.1
         test 10 {
            target 99.88.77.66
            target 55.44.33.22
            type ping
         }
      }
      rule 10 {
         inbound-interface eth3
         interface eth0 {
         }
      }
      rule 20 {
         destination {
            port sip
         }
         inbound-interface eth3
         interface eth1 {
         }
         protocol tcp
      }
}
[edit]
```

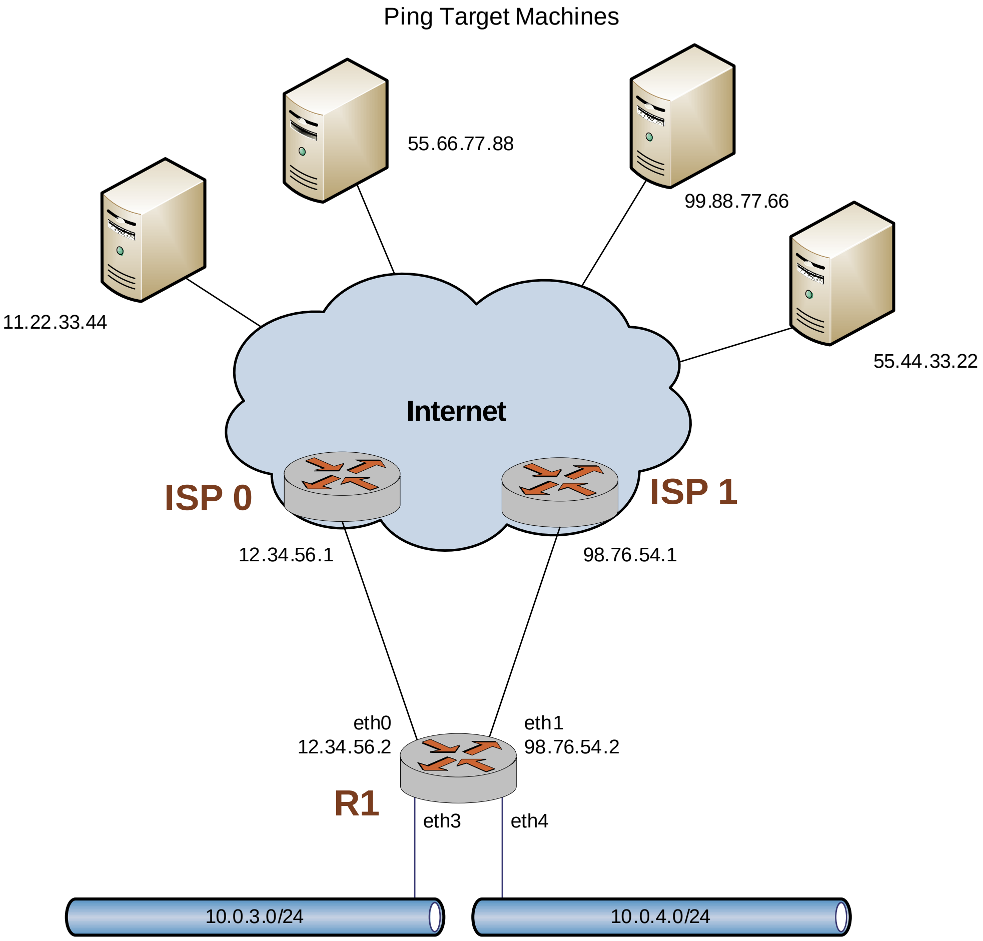
Display the static route configuration.

```
vyatta@R1# show protocols static
 route 0.0.0.0/0 next-hop 12.34.56.1
[edit]
```

Excluding Traffic from Load Balancing

In this example, an almost identical configuration to the first example is used. The only difference is that there are two local LANs rather than one, as shown in Figure 1-2.

Figure 1-2 WAN load balancing with two local networks



One would assume that simply adding another rule to account for traffic from the second source would suffice. For example, rule 20 is added to load balance traffic from eth4:

Example 1-6 Adding a rule for a second traffic source

Step	Command

Display the configuration	`vyatta@R1#` **`show load-balancing wan rule 20`**
	`inbound-interface eth4`
	`interface eth0 {`
	`}`
	`interface eth1 {`
	`}`
	`[edit]`

But this does not account for intra-LAN traffic. Traffic that is meant to be routed between LANs will also be sent out either eth0 or eth1 in this configuration. In order to prevent this, an additional rule is required to exclude intra-LAN traffic from being load balanced. For example, rule 5 is added to exclude all traffic that is destined to either of the LANs:

Example 1-7 Adding a rule to exclude traffic destined for the LANs

Step	Command

Display the configuration	`vyatta@R1#` **`show load-balancing wan rule 5`**
	`destination {`
	`      address 10.0.0.0/8`
	`}`
	`exclude`
	`inbound-interface eth+`
	`[edit]`

NOTE *eth+ means all ethernet interfaces.*

NOTE *This rule will also exclude any locally-sourced packets destined for 10.0.0.0/8. This is required when using services like Web Caching with WAN Load Balancing.*

The configuration looks as follows when complete:

Example 1-8 Configuration used with two LANs

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Display the configuration</td><td><pre>
vyatta@R1# <b>show load-balancing</b>
wan {
      interface-health eth0 {
         failure-count 5
         nexthop 12.34.56.1
         test 10 {
            target 11.22.33.44
            target 55.66.77.88
            type ping
         }
      }
      interface-health eth1 {
         failure-count 4
         nexthop 98.76.54.1
         test 10 {
            target 99.88.77.66
            target 55.44.33.22
            type ping
         }
      }
      rule 5 {
         destination {
            address 10.0.0.0/8
         }
         exclude
         inbound-interface eth+
      }
      rule 10 {
         inbound-interface eth3
         interface eth0 {
         }
         interface eth1 {
         }
      }
      rule 20 {
         inbound-interface eth4
         interface eth0 {
         }
         interface eth1 {
         }
      }
}
[edit]
</pre></td></tr>
</table>

WAN Load Balancing Commands

This section presents the following commands.

Configuration Commands	
Processing Directives	
load-balancing wan	Enables WAN load balancing on the system.
load-balancing wan disable-source-nat	Disables source NAT for balanced traffic.
load-balancing wan enable-local-traffic	Enables WAN load balancing for locally sourced traffic.
load-balancing wan flush-connections	Directs the system to flush the connection tracking table when a connection changes state.
load-balancing wan hook <script-name>	Specifies a script to be run on interface status changes.
Interface Health	
load-balancing wan interface-health <if-name>	Sets the characteristics for health checking for a load-balanced interface.
load-balancing wan interface-health <if-name> failure-count <num>	Sets the failure count for interface health checks.
load-balancing wan interface-health <if-name> nexthop <ipv4>	Sets the next-hop address for interface health checks.
load-balancing wan interface-health <if-name> test <test-no>	Defines an interface health test.
load-balancing wan interface-health <if-name> test <test-no> resp-time <seconds>	Sets the maximum response time before declaring a health check message failed.
load-balancing wan interface-health <if-name> test <test-no> target <address>	Specifies the address of the health check target.
load-balancing wan interface-health <if-name> test <test-no> ttl-limit <limit>	Specifies the hop count limit for a udp test.
load-balancing wan interface-health <if-name> test <test-no> type <type>	Specifies the interface health check test type.
load-balancing wan interface-health <if-name> success-count <num>	Sets the number of successful health checks required for an interface to be considered healthy.
Load Balancing Rules	
load-balancing wan rule <rule>	Defines a WAN load balancing rule.

load-balancing wan rule <rule> description <desc>	Specifies a description for a WAN load balancing rule.
load-balancing wan rule <rule> destination	Specifies a destination as a match criterion for a WAN load balancing rule.
load-balancing wan rule <rule> enable-source-based-routing	Enables source-based routing for a WAN load balancing rule.
load-balancing wan rule <rule> exclude	Excludes traffic matching a WAN load balancing rule from being load balanced.
load-balancing wan rule <rule> failover	Puts the load balancing process into “failover” mode, where one load balancing interface as active and the remaining links are spare.
load-balancing wan rule <rule> inbound-interface <if-name>	Specifies the interface that traffic to be load-balanced will come from.
load-balancing wan rule <rule> interface <if-name>	Adds an interface to the set of interfaces to be load-balanced in a WAN load balancing rule.
load-balancing wan rule <rule> limit	Specifies the traffic rate limiting parameters for a WAN load balancing rule.
load-balancing wan rule <rule> protocol <protocol>	Specifies an IP protocol as a match criterion for a WAN load balancing rule.
load-balancing wan rule <rule> source	Specifies a source as a match criterion for a WAN load balancing rule.
Operational Commands	
clear wan-load-balance process	Restarts the WAN load balancing process.
show wan-load-balance	Displays information about WAN load-balanced interfaces.
show wan-load-balance connection	Displays connection data generated by load balanced traffic.
show wan-load-balance status	Displays information about the status of WAN load balancing.

clear wan-load-balance process

Restarts the WAN load balancing process.

Syntax

clear wan-load-balance process

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to restart the WAN load balancing process.

load-balancing wan

Enables WAN load balancing on the system.

Syntax

set load-balancing wan

delete load-balancing wan

show load-balancing wan

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {}
}
```

Parameters

None.

Default

None.

Usage Guidelines

Use this command to enable wide area networking (WAN) load balancing on the system.

Use the **set** form of this command to create the configuration node for WAN load balancing.

Use the **delete** form of this command to remove the WAN load balancing configuration and disable WAN load balancing on the system.

Use the **show** form of this command to display the configuration node.

load-balancing wan disable-source-nat

Disables source NAT for balanced traffic.

Syntax

set load-balancing wan disable-source-nat

delete load-balancing wan disable-source-nat

show load-balancing wan

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      disable-source-nat
   }
}
```

Parameters

None.

Default

Source NAT rules are automatically generated when the source address is changed.

Usage Guidelines

Use this command to disable the automatic generation of source network address translation (source NAT or SNAT) rules for load balanced traffic.

Normally, the WAN load balanced interface replaces the source IP address of outbound traffic with its own IP address to ensure that reply traffic arrives back on the egressing interface. To do this, the WAN load balancing process automatically generates SNAT rules. These SNAT rules are specific to WAN load balancing; they rewrite the source address only for balanced traffic (that is, traffic matching the WAN load balancing rule).

In some scenarios—for example, cases where you are connecting private networks and do not require NAT, or cases where you want to employ other SNAT rules instead of relying on the rules generated by the WAN load balancing process. When the **load-balancing wan disable-source-nat** command is issued, WAN load balancing SNAT is not performed on balanced traffic.

Use the **set** form of this command to disable SNAT on balanced traffic.

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

Use the **show** form of this command to display WAN load balancing configuration.

load-balancing wan enable-local-traffic

Enables WAN load balancing for locally sourced traffic.

Syntax

set load-balancing wan enable-local-traffic

delete load-balancing wan enable-local-traffic

show load-balancing wan

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      enable-local-traffic
   }
}
```

Parameters

None.

Default

Traffic that originates on the Vyatta system does not participate in WAN load balancing..

Usage Guidelines

Use this command to enable WAN load balancing for traffic that originates on the Vyatta system itself.

Normally, only traffic that passes through the Vyatta system can be WAN load balanced. Setting this parameter allows both through traffic and traffic that originates from the Vyatta system to be WAN load balanced.

Use the **set** form of this command to enable WAN load balancing for locally sourced traffic..

Use the **delete** form of this command to restore the default behavior for locally sourced traffic.

Use the **show** form of this command to display WAN load balancing configuration.

load-balancing wan flush-connections

Directs the system to flush the connection tracking table when a connection changes state.

Syntax

set load-balancing wan flush-connections

delete load-balancing wan flush-connections

show load-balancing wan

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      flush-connections
   }
}
```

Parameters

None.

Default

The connection tracking table is not flushed when connections change state.

Usage Guidelines

Use this command to direct the system to flush the connection tracking table when any connection changes state. The complete connection tracking table is flushed.

The connection tracking table can accumulate stale entries—for example, when a DHCP advertisement causes a change of address for a balanced interface or where a balanced link fails. Enabling connection tracking directs the system to flush the table when a connection changes state in this way.

Note that enabling this option causes the entire connection tracking table to be flushed when any connection changes state. This removes entries for other flows than the changed flow, including entries for established and active flows. Subsequently, the system does not create a new entry in the connection tracking table until a new connection is established on the flow (where a flow is a tuple consisting of source address, destination address, IP address, and port). Until then, previously established flows are balanced on a per-packet, rather than a per-flow, basis.

Use the **set** form of this command to enable connection tracking table flushing.

Use the **delete** form of this command to restore the default behavior for the connection tracking table.

Use the **show** form of this command to display WAN load balancing configuration.

load-balancing wan hook <script-name>

Specifies a script to be run on interface status changes.

Syntax

set load-balancing wan hook *script-name*

delete load-balancing wan hook

show load-balancing wan hook

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      hook text
   }
}
```

Parameters

script-name	The path and name of the script to be executed on an interface state change.

Default

None.

Usage Guidelines

Use this command to specify the name of a script that will be run on a per interface state change basis. There are two environment variables available to the script:

WLB_INTERFACE_NAME=[interfacename] - specifies the interface to monitor for state change.

WLB_INTERFACE_STATE=[ACTIVE|FAILED] - specifies the interface state.

NOTE *This is a blocking call, so if the script doesn't return, the WAN load balancing process will wait forever - effectively hanging it.*

Use the **set** form of this command to specify the name of the script to be run when an interface changes state.

Use the **delete** form of this command to remove the specified script name.

Use the **show** form of this command to display the script name configured.

load-balancing wan interface-health <if-name>

Sets the characteristics for health checking for a load-balanced interface.

Syntax

set load-balancing wan interface-health *if-name*

delete load-balancing wan interface-health *if-name*

show load-balancing wan interface-health *if-name*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text
   }
}
```

Parameters

if-name	Mandatory. Multi-node. The name of a physical or logical interface. This is the load-balanced interface whose health is to be monitored. You can define health checks for all load-balanced interfaces by creating multiple **interface-health** configuration nodes.

Default

None.

Usage Guidelines

Use this command set the health-checking characteristics on a load-balanced outbound interface.

Use the **set** form of this command to enable health checking on an interface.

Use the **delete** form of this command to remove health checking on an interface.

Use the **show** form of this command to display health checking configuration.

load-balancing wan interface-health <if-name> failure-count <num>

Sets the failure count for interface health checks.

Syntax

set load-balancing wan interface-health *if-name* **failure-count** *num*

delete load-balancing wan interface-health *if-name* **failure-count**

show load-balancing wan interface-health *if-name* **failure-count**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         failure-count u32
      }
   }
}
```

Parameters

if-name	Mandatory. The name of a physical or logical interface.
num	The maximum number of failed health checks that can occur before the interface is considered unavailable. The range is 1 to 10. The default is 1.

Default

If an interface fails to respond to one health check, it is considered unavailable.

Usage Guidelines

Use this command to set the failure count for interface health checks. The failure count is the number of consecutive failed pings required to remove an interface from the pool of active load balanced interfaces.

Use the **set** form of this command to specify the failure count.

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

Use the **show** form of this command to display failure count configuration.

load-balancing wan interface-health <if-name> nexthop <ipv4>

Sets the next-hop address for interface health checks.

Syntax

set load-balancing wan interface-health *if-name* **nexthop** *ipv4*

delete load-balancing wan interface-health *if-name* **nexthop**

show load-balancing wan interface-health *if-name* **nexthop**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
    wan {
        interface-health text {
            nexthop ipv4
        }
    }
}
```

Parameters

if-name	Mandatory. The name of a physical or logical interface.
ipv4	The IPv4 address of the next hop for interface health checks, or the keyword **dhcp**, which directs the system to obtain the IP address from the DHCP server.

Default

None.

Usage Guidelines

Use this command to set the IPv4 address of the next hop for interface health checks.

If the next-hop address is specified using the **dhcp** keyword, the next-hop IP address is obtained from the DHCP advertisement. In this case, the applicable source NAT rule and the routing table are automatically updated when the new DHCP assignment is made.

Use the **set** form of this command to specify the IPv4 address of the next hop.

Use the **delete** form of this command to remove the IPv4 address of the next hop.

Use the **show** form of this command to display the next hop configuration.

load-balancing wan interface-health <if-name> test <test-no>

Defines an interface health test.

Syntax

set load-balancing wan interface-health *if-name* **test** *test-no*

delete load-balancing wan interface-health *if-name* **test**

show load-balancing wan interface-health *if-name* **test**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         test u32 {
         }
      }
   }
}
```

Parameters

if-name	The name of a physical or logical interface.
test-no	The test identifier.

Default

None.

Usage Guidelines

Use this command to define an interface health test. Multiple health test targets provide more than one target for a single interface. Rather than relying on a single target which might be non-responsive for reasons other than path failure, this allows multiple endpoints to be tested. Multi-targets will test until either the list of tests for that interface are exhausted or until the first successful response is received.

Use the **set** form of this command to specify the test configuration node.

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

Use the **show** form of this command to display test configuration.

load-balancing wan interface-health <if-name> test <test-no> resp-time <seconds>

Sets the maximum response time before declaring a health check message failed.

Syntax

set load-balancing wan interface-health *if-name* **test** *test-no* **resp-time** *seconds*

delete load-balancing wan interface-health *if-name* **test** *test-no* **resp-time**

show load-balancing wan interface-health *if-name* **test** *test-no* **resp-time**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         test u32 {
            resp-time u32
         }
      }
   }
}
```

Parameters

if-name	The name of a physical or logical interface.
test-no	The test identifier.
seconds	The number of seconds to wait for a ping response before declaring the ping to have failed. The range is 1 to 30. The default is 5.

Default

If an ICMP Echo Reply message is not received within 5 seconds, the ping test is considered to have failed.

Usage Guidelines

Use this command to configure the number of seconds to wait for a ping response before considering the health check to have failed. Pings are sent to test interface health when **type** is set to **ping**.

Use the **set** form of this command to set the maximum response time.

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

Use the **show** form of this command to display response time configuration.

load-balancing wan interface-health <if-name> test <test-no> target <address>

Specifies the address of the health check target.

Syntax

set load-balancing wan interface-health *if-name* **test** *test-no* **target** *address*

delete load-balancing wan interface-health *if-name* **test** *test-no* **target**

show load-balancing wan interface-health *if-name* **test** *test-no* **target**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         test u32 {
            target text
         }
      }
   }
}
```

Parameters

if-name	The name of a physical or logical interface.
test-no	The test identifier.
address	The IPv4 address or hostname of an interface health check target.

Default

None.

Usage Guidelines

Use this command to configure the destination for ping messages that test the interface health.

Use the **set** form of this command to set the destination for ping messages that test the interface health.

Use the **delete** form of this command to remove the destination for ping messages that test the interface health.

Use the **show** form of this command to display target configuration.

load-balancing wan interface-health <if-name> test <test-no> ttl-limit <limit>

Specifies the hop count limit for a udp test.

Syntax

set load-balancing wan interface-health *if-name* **test** *test-no* **ttl-limit** *limit*

delete load-balancing wan interface-health *if-name* **test** *test-no* **ttl-limit**

show load-balancing wan interface-health *if-name* **test** *test-no* **ttl-limit**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         test u32 {
            ttl-limit u32
         }
      }
   }
}
```

Parameters

if-name	The name of a physical or logical interface.
test-no	The test identifier.
limit	The hop count limit used when the test **type** is **udp**. The default value is 1.

Default

The limit is one hop.

Usage Guidelines

Use this command to configure the hop count limit for use with udp health check tests.

In addition to ping tests, it is possible to execute udp tests. In these, the target is sent a udp packet with a time-to-live (ttl) limit to the target. For the test to be successful the ttl limit must be shorter than the path length to the target as the test requires an icmp time expired message to be returned for a successful test.

Use the **set** form of this command to specify the hop count limit for use with udp health check tests.

Use the **delete** form of this command to remove the hop count limit.

Use the **show** form of this command to display ttl-limit configuration.

load-balancing wan interface-health <if-name> test <test-no> type <type>

Specifies the interface health check test type.

Syntax

set load-balancing wan interface-health *if-name* **test** *test-no* **type** [**ping** | **udp**]

delete load-balancing wan interface-health *if-name* **test** *test-no* **type**

show load-balancing wan interface-health *if-name* **test** *test-no* **type**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         test u32 {
            type [ping|udp]
         }
      }
   }
}
```

Parameters

if-name	The name of a physical or logical interface.
test-no	The test identifier.
type	The type of test to perform. Supported values are: **ping**: Run a ping test. **udp**: Run a udp test.

Default

None.

Usage Guidelines

Use this command to configure the type of health check test to be performed.

Use the **set** form of this command to specify the type of health check test to be performed.

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

Use the **show** form of this command to display type configuration.

load-balancing wan interface-health <if-name> success-count <num>

Sets the number of successful health checks required for an interface to be considered healthy.

Syntax

set load-balancing wan interface-health *if-name* **success-count** *num*

delete load-balancing wan interface-health *if-name* **success-count**

show load-balancing wan interface-health *if-name* **success-count**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      interface-health text {
         success-count u32
      }
   }
}
```

Parameters

if-name	Mandatory. The name of a physical or logical interface.
num	The number of consecutive successful pings required for the interface to be considered healthy. The range is 1 to 10. The default is 1.

Default

If an interface completes one successful ping, it is added back to the pool of active load-balanced interfaces.

Usage Guidelines

Use this command to set the number of consecutive successful ICMP Echo Request (ping) messages required to add an interface back into the pool of active load-balanced interfaces.

Use the **set** form of this command to specify the success count.

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

Use the **show** form of this command to display success count configuration.

load-balancing wan rule <rule>

Defines a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule*

delete load-balancing wan rule *rule*

show load-balancing wan rule *rule*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
      }
   }
}
```

Parameters

rule	Mandatory. Multi-node. A unique number identifying the rule. The range is 1 to 4294967295. You can define multiple load balancing rules by creating multiple **rule** configuration nodes.

Default

None.

Usage Guidelines

Use this command to define a WAN load balancing rule.

Once configured, rule numbers cannot be changed. For this reason, it is good practice to configure rules at intervals (for example, Rule 5, Rule 10, Rule 15, and so on) in case a rule must be inserted later on.

Use the **set** form of this command to create the load balancing rule. Note that you cannot use **set** to change the number of an existing rule. To change a rule’s number, delete the rule and re-create it.

Use the **delete** form of this command to remove a load balancing rule.

Use the **show** form of this command to display load balancing rule configuration.

load-balancing wan rule <rule> description <desc>

Specifies a description for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **description** *desc*

delete load-balancing wan rule *rule* **description**

show load-balancing wan rule *rule* **description**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         description text
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
desc	A description for the WAN load balancing rule.

Default

None.

Usage Guidelines

Use this command to provide a description for the WAN load balancing rule.

Use the **set** form of this command to specify a description for the WAN load balancing rule.

Use the **delete** form of this command to remove a description for the WAN load balancing rule.

Use the **show** form of this command to display the WAN load balancing rule description.

load-balancing wan rule <rule> destination

Specifies a destination as a match criterion for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **destination** {**address** *address* | **port** *port*}

delete load-balancing wan rule *rule* **destination** [**address** | **port**]

show load-balancing wan rule *rule* **destination**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         destination {
            address text
            port text
         }
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
address	The destination address to match. Supported formats are as follows: *ip-address*: An IPv4 address. *ip-address/prefix*: A network address, where 0.0.0.0/0 matches any network. *ip-address–ip-address*: A range of contiguous IP addresses; for example, 192.168.1.1–192.168.1.150. **!***ip-address*: Matches all IP addresses except the one specified. **!***ip-address/prefix*: Matches all network addresses except the one specified. **!***ip-address–ip-address*: Matches all IP addresses except those in the specified range. Exactly one of **address** or **port** must be specified.

port	Applicable only when the protocol is TCP or UDP. The destination port to match. Supported formats are as follows: *port-name*: Matches the name of an IP service; for example, **http**. You can specify any service name in the file /**etc/services**. *port-num*: Matches a port number. The range is 1 to 65535. start–end: Matches the specified range of ports; for example, 1001–1005. You can use a combination of these formats in a comma-separated list. You can also negate the entire list by prepending it with an exclamation mark (“!”); for example,**!22,telnet,http,123,1001-1005**. Exactly one of **address** or **port** must be specified.

Default

If not set, or if the **destination** configuration node is created with no attributes, the packet matches any destination.

Usage Guidelines

Use this command to define a match criterion based on destination address for a load balancing rule.

You can match packets based on a destination represented by an IP address or port.

Use the **set** form of this command to specify a destination to be matched.

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

Use the **show** form of this command to display destination configuration.

load-balancing wan rule <rule> enable-source-based-routing

Enables source-based routing for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **enable-source-based-routing**

delete load-balancing wan rule *rule* **enable-source-based-routing**

show load-balancing wan rule *rule*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         enable-source-based-routing
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.

Default

None.

Usage Guidelines

Use this command to enable source-based routing (also called policy-based routing) during load balancing.

Normally, the Vyatta system balances traffic flows based on a tuple consisting ot source address, destination address, and port. The first packet of the flow initiates the connection and subsequent packets matching the flow tuple are directed to the same interface. Flow-based load balancing helps prevent problems associated with out-of-order packets; for example, in cases where one link in the set is much faster or slower than others in the set. (Out-of-order packets can affect performance because of the processing required to reassemble the flow.)

For some scenarios—for example, scenarios where out-of-order packets are not a concern—flow-based balancing is not required; packet-based balancing (that is, simple statistically random distribution) is sufficient. Disabling flow-based balancing can afford efficiencies in forwarding packets; in addition, a better balance of packets can be achieved.

Use the **set** form of this command to enable source-based routing.

Use the **delete** form of this command to disable source-based routing.

Use the **show** form of this command to display WAN load balancing configuration.

load-balancing wan rule <rule> exclude

Excludes traffic matching a WAN load balancing rule from being load balanced.

Syntax

set load-balancing wan rule *rule* **exclude**

delete load-balancing wan rule *rule* **exclude**

show load-balancing wan rule *rule*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         exclude
      }
   }
}
```

Parameters

None.

Default

Traffic matching the characteristics specified in this rule is load balanced.

Usage Guidelines

Use this command to define an “exclusion” rule excluding traffic from being load balanced.

Traffic matching the characteristics specified in the load balancing rule is not load balanced, but is routed normally.

Use the **set** form of this command to exclude traffic matching this rule from being load balanced.

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

Use the **show** form of this command to display WAN load balancing rule configuration.

load-balancing wan rule <rule> failover

Puts the load balancing process into “failover” mode, where one load balancing interface as active and the remaining links are spare.

Syntax

set load-balancing wan rule *rule* **failover**

delete load-balancing wan rule *rule* **failover**

show load-balancing wan rule *rule*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         failover
      }
   }
}
```

Parameters

None.

Default

All configured load balancing links are used to balance traffic.

Usage Guidelines

Use this command to direct the system to put the load balancing process into “failover” mode. In “failover” mode, one load balancing link is selected by the system as the active link; the remaining load balancing links are reserved as “standby” or spare links to be used in case the connection to the active link is interrupted.

The active link is selected by the system based on its configured weight and the reachability of the target from the interface. Only the active link is used to forward traffic. If the active link becomes inoperable, the interface with the next highest combination of weight and reachability becomes the active link.

Use the **set** form of this command to enable “failover” mode.

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

Use the **show** form of this command to display WAN load balancing rule configuration.

load-balancing wan rule <rule> inbound-interface <if-name>

Specifies the interface that traffic to be load-balanced will come from.

Syntax

set load-balancing wan rule *rule* **inbound-interface** *if-name*

delete load-balancing wan rule *rule* **inbound-interface** *if-name*

show load-balancing wan rule *rule* **inbound-interface**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         inbound-interface text
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
if-name	Mandatory. The interface that traffic to be load-balanced will come from.

Default

None.

Usage Guidelines

Use this command to specify the interface that traffic to be load-balanced will come from.

Use the **set** form of this command to specify the interface that traffic to be load-balanced will come from.

Use the **delete** form of this command to remove the inbound interface from the load balancing rule.

Use the **show** form of this command to display inbound interface configuration in a load balancing rule.

load-balancing wan rule <rule> interface <if-name>

Adds an interface to the set of interfaces to be load-balanced in a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **interface** *if-name* [**weight** *num*]

delete load-balancing wan rule *rule* **interface** *if-name* [**weight**]

show load-balancing wan rule *rule* **interface** *if-name* [**weight**]

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         interface text {
            weight 1-255
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
if-name	Mandatory. The name of a physical or logical interface.
weight	The weight to be associated with the interface, where weight represents the relative distribution of packets to this interface.The range is 1 to 255. The default is 1.

Default

Each interface is assigned a weight of 1.

Usage Guidelines

Use this command to add an interface to the set of interfaces to be load-balanced in a WAN load balancing rule. When a load balancing rule is matched, the outgoing packet is sent out through one of the interfaces specified in this set, as determined by the load balancing algorithm.

Use the **set** form of this command to add an interface to the load balancing rule or to modify an interface’s load balancing weight.

Use the **delete** form of this command to remove the interface from the load balancing rule or to restore the default weight of an interface.

Use the **show** form of this command to display interface configuration in a load balancing rule.

load-balancing wan rule <rule> limit

Specifies the traffic rate limiting parameters for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **limit** {**burst** *burst* | **period** [**second**|**minute**|**hour**] | **rate** *rate* | **threshold** [**above**|**below**]}

delete load-balancing wan rule *rule* **limit** [**burst** | **period** | **rate** | **threshold**]

show load-balancing wan rule *rule* **limit** [**burst** | **period** | **rate** | **threshold**]

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         limit {
            burst u32
            period [second|minute|hour]
            rate u32
            threshold [above|below]
         }
      }
   }
```

Parameters

rule	The number of the rule being configured.
burst	The number of packets over the limit that are allowed within the specified **period**. The default value is 5.
period	The time window for rate calculation. Supported values are: **second**: One second **minute**: One minure **hour**: One hour The default value is **second**.
rate	The number of packets used for the rate limit. The default value is 5.

threshold	The threshold behavior for limit. Supported values are: **above**: The rule applies to packets above the limit **below**: The rule applies to packets below the limit The default value is **below**.

Default

No limits are applied.

Usage Guidelines

Use this command to set the rate limit at which the rule will be active. It can be interpreted as “Apply this rule to traffic below (or above) this rate. Allow bursts of x packets per time period above (or below) this rate as well.”

Use the **set** form of this command to specify the traffic rate limiting parameters for a WAN load balancing rule.

Use the **delete** form of this command to remove the traffic rate limiting parameters for a WAN load balancing rule.

Use the **show** form of this command to display the traffic rate limiting parameters for a WAN load balancing rule.

load-balancing wan rule <rule> protocol <protocol>

Specifies an IP protocol as a match criterion for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **protocol** *protocol*

delete load-balancing wan rule *rule* **protocol** *protocol*

show load-balancing wan rule *rule* **protocol** *protocol*

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         protocol text
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
protocol	The protocol(s) on which to perform load balancing. Any protocol literals or numbers listed in **/etc/protocols** can be used. The keyword **all** is also supported. Prefixing the protocol name with the exclamation mark character (“!”) matches every protocol except the specified protocol. For example, **!tcp** matches all protocols except TCP.

Default

All protocols are matched.

Usage Guidelines

Use this command to specify the protocol(s) on which to define a match.

Note that you should take care in using more than one “exclusion” rule (that is, a rule using the negation operation (“!”) in combination. Locad balancing rules are evaluated sequentially, and a sequence of exclusion rules could result in unexpected behavior.

Use the **set** form of this command to specify a protocol to be matched.

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

Use the **show** form of this command to display protocol match configuration.

load-balancing wan rule <rule> source

Specifies a source as a match criterion for a WAN load balancing rule.

Syntax

set load-balancing wan rule *rule* **source** {**address** *address* | **port** *port*}

delete load-balancing wan rule *rule* **source** {**address** | **port**}

show load-balancing wan rule *rule* **source**

Command Mode

Configuration mode.

Configuration Statement

```
load-balancing {
   wan {
      rule u32 {
         source {
            address text
            port text
         }
      }
   }
}
```

Parameters

rule	Mandatory. The number of the rule being configured.
address	The destination address to match. Supported formats are as follows: *ip-address*: An IPv4 address. *ip-address/prefix*: A network address, where 0.0.0.0/0 matches any network. *ip-address–ip-address*: A range of contiguous IP addresses; for example, 192.168.1.1–192.168.1.150. **!***ip-address*: Matches all IP addresses except the one specified. **!***ip-address/prefix*: Matches all network addresses except the one specified. **!***ip-address–ip-address*: Matches all IP addresses except those in the specified range. Exactly one of **address** or **port** must be specified.

port	Applicable only when the protocol is TCP or UDP. The destination port to match. Supported formats are as follows: *port-name*: Matches the name of an IP service; for example, **http**. You can specify any service name in the file /**etc/services**. *port-num*: Matches a port number. The range is 1 to 65535. start–end: Matches the specified range of ports; for example, 1001–1005. You can use a combination of these formats in a comma-separated list. You can also negate the entire list by prepending it with an exclamation mark (“!”); for example,**!22,telnet,http,123,1001-1005**. Exactly one of **address** or **port** must be specified.

Default

If not set, or if the **source** configuration node is created with no attributes, the packet matches any source.

Usage Guidelines

Use this command to define a match criterion based on source address for a load balancing rule.

You can match packets based on a source represented by an IP address or port.

Use the **set** form of this command to specify a source to be matched.

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

Use the **show** form of this command to display source configuration.

show wan-load-balance

Displays information about WAN load-balanced interfaces.

Syntax

show wan-load-balance

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see information about WAN load-balanced interfaces.

The command displays information for each balanced interface and reports on the current status, including the last time the interface changed state from active to failed or failed to active.

The command also shows the test type and the target for the test (in order of configured test number). The character at the beginning of the line represents the state of the test:

+ = the last test was successful

- = the last test failed

* - no test has been performed

Examples

Example 1-9 shows WAN load balancing information where eth0 and eth1 are balanced interfaces.

Example 1-9 Displaying load balanced interface information

```
vyatta@vyatta>show wan-load-balance
Interface:  eth0
  Status:  active
  Last Status Change:  Fri May 15 13:38:39 2009
  -Test:  Ping  Target:  192.168.0.120
  +Test:  ttl   Target:  192.168.0.1
    Last Interface Success:  0s
    Last Interface Failure:  35s
    # Interface Failure(s):  10
```

```
Interface:  eth1
  Status:  active
  Last Status Change:  Fri May 15 13:38:39 2009
  +Test:  Ping  Target:  192.168.0.1
  *Test:  Ping  Target:  192.168.0.120
    Last Interface Success:  10s
    Last Interface Failure:  0s
    # Interface Failure(s):  0
```

show wan-load-balance connection

Displays connection data generated by load balanced traffic.

Syntax

show wan-load-balance connection

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see connection information generated by load balanced traffic.

Examples

Example 1-10 shows WAN load balancing connection information.

Example 1-10 Displaying load balancing connection information

```
   vyatta@vyatta>show wan-load-balance connection
   Type     State     Src                    Dst                 Packets    Bytes
   tcp      estab     172.16.117.1:123       172.16.117.2:123       1          11
   icmp               172.16.117.1           172.16.117.2           1          11
```

show wan-load-balance status

Displays information about the status of WAN load balancing.

Syntax

show wan-load-balance status

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to see information about status of WAN load balancing.

Examples

Example 1-11 shows WAN load balancing information where eth0 and eth1 are balanced interfaces.

Example 1-11 Displaying load balancing status

```
vyatta@vyatta>show wan-load-balance status
Chain PREROUTING (policy ACCEPT 1415 packets, 96338 bytes)
 pkts bytes target prot opt in   out source   destination
 0    0     ISP_1  tcp  --   any any anywhere anywhere multiport dports 223
                state NEW statistic mode random probability 0.500000
 0    0     ISP_2  tcp  --   any any anywhere anywhere multiport dports 223
```

Chapter 2: VRRP

This chapter explains how to use Virtual Router Redundancy Protocol (VRRP) on the Vyatta system.

This chapter presents the following topics:

- VRRP Configuration
- VRRP Commands

VRRP Configuration

This section describes how to configure the Virtual Router Redundancy Protocol on the Vyatta system.

This section presents the following topics:

- VRRP Overview
- VRRP Configuration Examples

VRRP Overview

Virtual Router Redundancy Protocol (VRRP) is a protocol for allowing a cluster of routers to act as one virtual router. VRRP, as specified by RFC 2338 and RFC 3678, was designed to provide router failover services in the event of an interface failure.

On the Vyatta system, VRRP can be run on either a standard Ethernet interface, or it can be run on the vif of an Ethernet interface (that is, a VLAN interface).

This section presents the following topics:

- VRRP Groups
- The Virtual IP Address
- Election of the Master Router
- VRRP Advertisements and Failover
- Preemption
- VRRP Authentication
- VRRP Sync Groups

VRRP Groups

A VRRP group consists of a cluster of interfaces and/or virtual interfaces providing redundancy for a primary, or “master,” interface in the group. Redundancy is managed by the VRRP process on the system.

The VRRP group has a unique numeric identifier and is assigned a single virtual IP address (sometimes called a virtual IP or VIP). The virtual address is linked with the MAC address of the master router. If the master router fails, a new master is elected and the new master notifies the network of its MAC address by issuing a gratuitous ARP.

All interfaces in the group must be assigned the same VRRP group identifier and virtual address; otherwise they cannot provide redundancy for one another. Interfaces being mapped to the virtual address must be on the same subnet as the virtual address, but should not have the same address as the virtual address.

The Virtual IP Address

Routers in a VRRP cluster share a virtual IP address (the VIP) and a virtual MAC address. This provides alternate paths through the network for hosts without explicitly configuring them, and creates redundancy that eliminates any individual router as a single point of failure in the network. This is particularly important for statically configured default routers, the failure of which could otherwise be a catastrophic event on a network.

In VRRP, the IP addresses of interfaces on different real routers are mapped onto a “virtual router”. The virtual router is an abstract object, managed by the VRRP process, that is defined by its virtual router ID (the group identifier of the set of routers forming the virtual router) plus the VIP presented to the network. Hosts on the network are configured to direct packets to the VIP, rather than to the IP addresses of the real interfaces.

The virtual router uses the group identifier to construct a virtual MAC address from a standard MAC prefix (specified in the VRRP standard) plus the group identifier. ARP requests for the VIP are resolved to the virtual MAC address, which “floats” from real router to real router, depending on which is acting as the master router of the virtual router. If the master router fails, the backup router is brought into service using the virtual MAC address and VIP of the virtual router. In this way, service can continue around a failed gateway transparently to hosts on the LAN.

The master router forwards packets for local hosts and responds to ARP requests, ICMP pings, and IP datagrams directed to the VIP. Backup routers remain idle, even if healthy. ARP requests, pings, and datagrams made to the real IP addresses of interfaces are responded to by the interface in the normal way.

Election of the Master Router

VRRP dynamically elects the router that is to be the master. In most cases, the master router is simply the router with the interface that has the highest configured priority. If two interfaces have identical priorities, the router with the one having the highest IP address is elected master.

If the master interface fails, the interface with the next highest priority is elected master and assumes the virtual address of the group. The new master notifies the network of its MAC address by sending out a gratuitous ARP message.

The priority of the master interface is typically set to 255. The backup interface can be left with the default priority; however, if more than one interface is acting as backup, they should be configured with different priorities.

VRRP Advertisements and Failover

To signal that it is still in service, the master interface or vif sends MAC-level multicast “heartbeat” packets called advertisements to the backup routers on the LAN segment, using the IP address 224.0.0.18, which is the IPv4 multicast address assigned to VRRP. These advertisements confirm the health of the master to backup routers and contain other VRRP information, such as the master’s priority.

If the heartbeat stops for a configured period (the “dead interval”), the VRRP process considers the master out of service and triggers failover by electing the backup interface with the highest priority to become the new master router. The new master assumes the virtual address and notifies the network of its MAC address by issuing a gratuitous ARP message.

Preemption

If preemption is enabled, a backup router with a higher priority than the current master will “preempt” the master, and become the master itself. The backup router preempts the master by beginning to send out its own VRRP advertisements. The master router examines these, and discovers that the backup router has a higher priority than itself. The master then stops sending out advertisements, while the backup continues to send, thus making itself the new master.

Preemption is useful in situation where a lower-performance backup router becomes master when a higher-performance router fails. In this case, a new higher-performance router can be brought online, and it will automatically preempt the lower-performance backup.

VRRP Authentication

If a password is set for VRRP authentication, the authentication type must also be defined. If the password is set and authentication type is not defined, the system generates an error when you try to commit the configuration.

Similarly, you cannot delete the VRRP password without also deleting the VRRP authentication type. If you do, the system generates an error when you try to commit the configuration.

If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the vif.

VRRP Sync Groups

Interfaces in a VRRP sync group are synchronized such that, if one of the interfaces in the group fails over to backup, all interfaces in the group fail over to backup.

For example, in many cases, if one interface on a master router fails, the whole router should fail over to a backup router. By assigning all the interfaces on the master to a sync group, the failure of one interface will trigger a failover of all the interfaces in the sync group to the backup configured for the interface.

VRRP Configuration Examples

This section presents the following topics:

- Configuring the First System
- Configuring the Second System

This sequence sets up a basic VRRP configuration between two Vyatta systems.

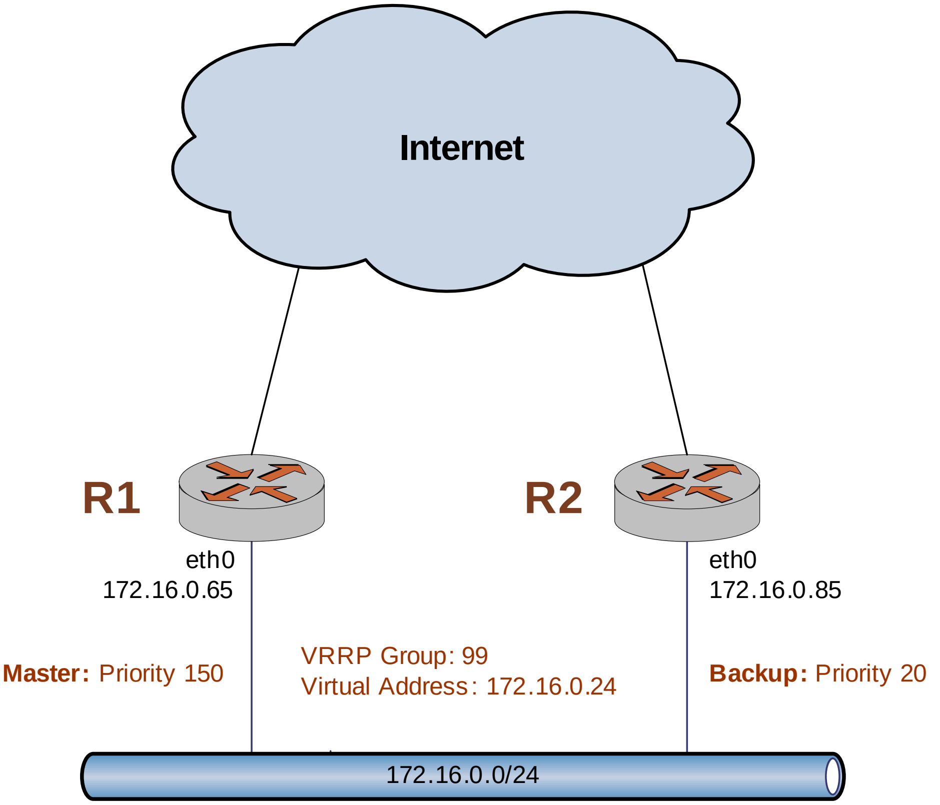
Remember that in VRRP:

- The system configured with the highest priority will initially be elected the master router. If more than one system has the highest priority, then the first active system will be elected the master router.
- Enabling preemption will allow a higher-priority neighbor to preempt the current master and become master itself.

The implementation is currently restricted to one VRRP group per interface, regardless of whether the group is defined at the physical interface level or the vif level.

In this section, sample configurations are presented for VRRP. When you have finished, the system will be configured as shown in Figure 2-1.

Figure 2-1 VRRP



This section includes the following examples:

- Example 2-1 Configuring a first system for VRRP
- Example 2-2 Configuring a backup system for VRRP

Configuring the First System

Example 2-1 enables VRRP on eth0 of the first system (R1) and assigns it to VRRP group 99. The virtual address is 172.16.0.24/24. Preemption is enabled, and R1 is assigned a priority of 150.

To configure the first system for VRRP, perform the following steps in configuration mode:

Example 2-1 Configuring a first system for VRRP

Step	Command
Create the VRRP configuration node for eth0 on R1. This enables VRRP on that interface. Assign the VRRP group.	vyatta@R1# **set interfaces ethernet eth0 vrrp vrrp-group 99** [edit]
Specify the virtual address of the VRRP group.	vyatta@R1# **set interfaces ethernet eth0 vrrp vrrp-group 99 virtual-address 172.16.0.24/24** [edit]
Enable preemption.	vyatta@R1# **set interfaces ethernet eth0 vrrp vrrp-group 99 preempt true** [edit]
Set the priority of this system to 150.	vyatta@R1# **set interfaces ethernet eth0 vrrp vrrp-group 99 priority 150** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Display the configuration	vyatta@R1# **show interfaces ethernet eth0 vrrp**
	 vrrp-group 99 {
	      preempt true
	      priority 150
	      virtual-address 172.16.0.24/24
	 }

Configuring the Second System

Example 2-2 enables VRRP on eth0 of the second system (R2), and assigns it to VRRP group 99. The virtual address is the same as that for R1: 172.16.0.24/24. Preemption is enabled, and R2 is assigned a priority of 20. This is lower than the priority of R1, so R1 will be the master and R2 will be the backup under ordinary circumstances.

To configure the second system for VRRP, perform the following steps in configuration mode:

Example 2-2 Configuring a backup system for VRRP

Step	Command
Create the VRRP configuration node for eth0 of R2. This enables VRRP on that interface. Assign the VRRP group.	vyatta@R2# **set interfaces ethernet eth0 vrrp vrrp-group 99** [edit]
Specify the virtual address of the VRRP group.	vyatta@R2# **set interfaces ethernet eth0 vrrp vrrp-group 99 virtual-address 172.16.0.24/24** [edit]
Enable preemption.	vyatta@R2# **set interfaces ethernet eth0 vrrp vrrp-group 99 preempt true** [edit]
Set the priority of this system to 20. This is a lower priority than that set for R1, so R1 will become the master.	vyatta@R2# **set interfaces ethernet eth0 vrrp vrrp-group 99 priority 20** [edit]
Commit the configuration.	vyatta@R2# **commit** [edit]

Display the configuration	vyatta@R1# **show interfaces ethernet eth0 vrrp**
	vrrp-group 99 {
	preempt true
	priority 20
	virtual-address 172.16.0.24/24
	}

VRRP Commands

This section presents the following commands.

Configuration Commands	
Ethernet Interface VRRP Configuration Commands	
interfaces <interface> vrrp vrrp-group <group-id>	Assigns an Ethernet interface to a VRRP group.
interfaces <interface> vrrp vrrp-group <group-id> advertise-interval <interval>	Sets the advertisement interval for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> authentication password	Sets the VRRP authentication password for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> authentication type	Specifies the VRRP authentication type for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> description <desc>	Specifies a description for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> disable	Disables a VRRP group.
interfaces <interface> vrrp vrrp-group <group-id> hello-source-address <addr>	Specifies the source address for VRRP hello packets.
interfaces <interface> vrrp vrrp-group <group-id> preempt <preempt>	Enables or disables preemption for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> preempt-delay <delay>	Sets the preemption delay for a VRRP group on an interface.
interfaces <interface> vrrp vrrp-group <group-id> priority <priority>	Sets the priority of an interface within a VRRP group.
interfaces <interface> vrrp vrrp-group <group-id> run-transition-scripts	Specify a script to run on VRRP state transition.
interfaces <interface> vrrp vrrp-group <group-id> sync-group <group>	Assigns an interface to a VRRP sync group.
interfaces <interface> vrrp vrrp-group <group-id> virtual-address <addr>	Sets the virtual IP address or network address for a VRRP group on an interface.
Ethernet Vif VRRP Configuration Commands	
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id>	Assigns a vif to a VRRP group.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> advertise-interval <interval>	Sets the advertisement interval for a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> authentication password <pwd>	Sets a VRRP authentication password for a VRRP group on a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> authentication type <type>	Specifies the VRRP authentication type for a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> description <desc>	Specifies a description for a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> disable	Disables a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> hello-source-address <addr>	Specifies the source address for VRRP hello packets.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> preempt <preempt>	Enables or disables preemption for a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> preempt-delay <delay>	Sets the preemption delay for a VRRP group on a vif.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> priority <priority>	Sets the priority of a vif within a VRRP group.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> run-transition-scripts	Specify a script to run on VRRP state transition.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> sync-group <group>	Assigns a vif to a VRRP sync group.
interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> virtual-address <addr>	Sets the virtual IP address or network address for a VRRP group on a vif.
Operational Commands	
clear vrrp master interface <interface> group <group-id>	Force a VRRP state transition to the backup state.
clear vrrp process	Restarts the VRRP process.
show vrrp	Displays information about VRRP groups.

clear vrrp master interface <interface> group <group-id>

Force a VRRP state transition to the backup state.

Syntax

clear vrrp master interface *interface* **group** *group-id*

Command Mode

Operational mode.

Parameters

interface	The interface to force to backup. Ethernet interfaces (eth0..eth23) and Ethernet Link Bonding interfaces (bond0..bond99) are supported. Virtual interfaces on either of these are also supported.
group-id	The group within the interface to force to backup.

Default

None.

Usage Guidelines

Use this command to force a VRRP state transition to the backup state. It will transition the current master to the backup state.

Examples

Example 2-3 shows sample output for the **clear vrrp master interface <interface> group <group-id>** command. Notice that prior to the command running the “State” is “master” and afterwards it is “backup”. Also notice the change in “Master router”.

Example 2-3 Forcing the VRRP master into the backup state

```
vyatta@vyatta:~$ show vrrp interface eth1
Physical interface: eth1.4001, Address 172.16.40.160
  Interface state: up, Group 200, State: master
  Priority: 201, Advertisement interval: 1, Authentication type: none
  Preempt: false, VIP count: 2, VIP: 172.16.40.100
                                     172.16.40.101
  Master router: 172.16.40.160
  Last transition: 51s
```

```
vyatta@vyatta:~$ clear vrrp master interface eth1.4001 group 200
Forcing eth1.4001-200 to BACKUP...
vyatta@vyatta:~$ show vrrp interface eth1
Physical interface: eth1.4001, Address 172.16.40.160
  Interface state: up, Group 200, State: backup
  Priority: 201, Advertisement interval: 1, Authentication type: none
  Preempt: false, VIP count: 2, VIP: 172.16.40.100
                                     172.16.40.101
  Master router: 172.16.40.128 [00:0C:29:11:B2:75], Master Priority: 200
  Last transition: 3s

vyatta@vyatta:~$
```

clear vrrp process

Restarts the VRRP process.

Syntax

clear vrrp process

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to restart the VRRP process.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id>

Assigns a vif to a VRRP group.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id*

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. Multi-node. An integer uniquely identifying a VRRP group. The range is 1 to 255, where the higher the number, the higher the priority. You can assign a vif to multiple VRRP groups by creating multiple **vrrp-group** configuration nodes within the **vif** configuration node.

Default

Vifs are not assigned to a VRRP group.

Usage Guidelines

Use this command to assign a vif to a VRRP group.

An interface or virtual interface can belong to more than one VRRP group.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to assign a vif to a VRRP group.

Use the **delete** form of the command to remove a vif from a VRRP group.

Use the **show** form of the command to view VRRP group configuration settings for a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> advertise-interval <interval>

Sets the advertisement interval for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **advertise-interval** *interval*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **advertise-interval**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **advertise-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            advertise-interval 1-255
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
interval	Optional. The interval in seconds between VRRP advertisement packets. All interfaces in this VRRP group must use the same advertisement interval. The range is 1 to 255. The default is 1.

Default

The master router sends VRRP advertisements at 1-second intervals.

Usage Guidelines

Use this command to set the interval between VRRP advertisements on a vif VRRP group.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to set the VRRP advertise interval for a vif VRRP group.

Use the **delete** form of the command to restore the default value for VRRP advertise interval for a vif VRRP group.

Use the **show** form of the command to view vif VRRP group advertise interval configuration.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> authentication password <pwd>

Sets a VRRP authentication password for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication password** *pwd*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication password**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication password**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            authentication {
               password text
            }
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
pwd	Mandatory. The password the interface will use to authenticate itself as a member of the VRRP group.

Default

Interfaces are not required to authenticate themselves to the VRRP group.

Usage Guidelines

Use this command to set a password for VRRP authentication on a vif.

If a password is set for VRRP authentication, the authentication type (AH or plaintext-password) must also be defined. If the password is set and authentication type is not defined, the system will generate an error when you try to commit the configuration.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify a VRRP authentication password for a vif VRRP group.

Use the **delete** form of the command to delete the VRRP authentication password.

- You cannot delete the VRRP password without also deleting the VRRP authentication type. If you do, the system will generate an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the vif.

Use the **show** form of the command to view the VRRP authentication password for a vif VRRP group.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> authentication type <type>

Specifies the VRRP authentication type for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication type** *type*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication type**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **authentication type**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
    vif [0-4095] {
        vrrp {
            vrrp-group [1-255] {
                authentication {
                    type {
                        ah
                        plaintext-password
                    }
                }
            }
        }
    }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
pwd	Mandatory. The plaintext password the interface will use to authenticate itself as a member of the group.

type	The type of authentication to be used. Supported values are as follows: **ah**: The IP Authentication Header (AH) protocol is used. **plaintext-password**: Plain-text password authentication is used.

Default

Interfaces are not required to authenticate themselves to the VRRP group.

Usage Guidelines

Use this command to set the authentication type for VRRP authentication on a vif.

If the authentication type is set for VRRP authentication, a password must also be specified. If the authentication type is defined and a password is not set, the system will generate an error when you try to commit the configuration.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the VRRP authentication type for a vif VRRP group.

Use the **delete** form of the command to delete the authentication type.

- You cannot delete the VRRP authentication type without also deleting the VRRP password. If you do, the system will generate an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the vif.

Use the **show** form of the command to view the VRRP authentication password for a vif VRRP group.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> description <desc>

Specifies a description for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **description** *desc*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **description**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            description text
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
desc	A description for the VRRP group on a vif.

Default

Preemption is enabled.

Usage Guidelines

Use this command to provide a description for a VRRP group on a vif.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to provide a description for a vif VRRP group.

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

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

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> disable

Disables a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **disable**

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **disable**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            disable
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.

Default

The VRRP group is enabled.

Usage Guidelines

Use this command to disable a VRRP group on a vif.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to disable a VRRP group on a vif.

Use the **delete** form of the command to reenable a VRRP group on a vif.

Use the **show** form of the command to view VRRP group configuration on a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> hello-source-address <addr>

Specifies the source address for VRRP hello packets.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **hello-source-address** *addr*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **hello-source-address** *addr*

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **hello-source-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
    vif [0-4095] {
        vrrp {
            vrrp-group [1-255] {
                hello-source-address ipv4
            }
        }
    }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
addr	Mandatory. The IP address to use as the VRRP source address when sending VRRP hello packets. The format is *ipv4-addr*. The address must already be defined on an interface.

Default

The IP address of the vif is used as the source for VRRP hello packets.

Usage Guidelines

Use this command to specify the source address for VRRP hello packets. This is typically used when an address other than the default address for the interface is required. Note that the address must be defined on an interface already

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the source address for VRRP hello packets.

Use the **delete** form of the command to restore the default source address.

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

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> preempt <preempt>

Enables or disables preemption for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt** *preempt*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            preempt [true|false]
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.

preempt	Optional. Allows a higher-priority VRRP backup router to assert itself as master over a lower-priority master router. Supported values are as follows: **true**: Allow the master router to be preempted by a backup router with higher priority. **false**: Do not allow the master router to be preempted by a backup router with higher priority. The default is **true**; that is, the master router can be preempted by a backup router with higher priority.

Default

Preemption is enabled.

Usage Guidelines

Use this command to enable or disable preemption on a vif.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to enable or disable VRRP preemption on a vif.

Use the **delete** form of the command to restore the default value for VRRP preemption on a vif.

Use the **show** form of the command to view VRRP preemption configuration on a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> preempt-delay <delay>

Sets the preemption delay for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt-delay** *delay*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt-delay**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **preempt-delay**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif 0-4095 {
      vrrp {
         vrrp-group 1-255 {
            preempt-delay 0-3600
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	The VRRP group being configured. The range is 1 to 255.
delay	The amount of time to postpone preemption, in seconds. The range is 0 to 3600 (1 hour), where 0 means no delay. The default is 0.

Default

A router preempting another router does not wait.

Usage Guidelines

Use this command to set the preemption delay on a vif. The preemption delay is the amount of time a router must wait before preempting a lower-priority VRRP router and becoming the master.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to set the preemption delay.

Use the **delete** form of the command to restore the default value preemption delay.

Use the **show** form of the command to view preemption delay configuration on a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> priority <priority>

Sets the priority of a vif within a VRRP group.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **priority** *priority*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **priority**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            priority [1-255]
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
priority	Mandatory. The priority with which this interface should be considered for election as master within the VRRP group. The higher the configured number, the higher the priority. The range for a VRRP backup router is from 1 to 254. The VRRP master router must have the highest priority, and typically has a priority of 255. The default is 1.

Default

The default is 1.

Usage Guidelines

Use this command to set the VRRP priority of a vif on a real router. This determines the likelihood of its being elected the master router in a cluster of VRRP routers.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the VRRP group priority for a vif.

Use the **delete** form of the command to restore the default VRRP group priority to a vif.

Use the **show** form of the command to view vif VRRP group priority configuration.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> run-transition-scripts

Specify a script to run on VRRP state transition.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**] *script*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**]

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            run-transition-scripts {
               backup text
               fault text
               master text
            }
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.

backup *script*	The name of the executable script to run during VRRP state transition to the backup state.
fault *script*	The name of the executable script to run during VRRP state transition to the fault state.
master *script*	The name of the executable script to run during VRRP state transition to the master state.

Default

None.

Usage Guidelines

Use this command to specify a script to run when the VRRP group on the specified interface changes state. The state is either backup, fault, or master.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify a script to run when the VRRP group on the specified interface changes state.

Use the **delete** form of the command to stop the script from being run when the specifed state transition occurs.

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

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> sync-group <group>

Assigns a vif to a VRRP sync group.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **sync-group** *group*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **sync-group**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **sync-group**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            sync-group text
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
group	A text string defining the name of a sync group.

Default

None.

Usage Guidelines

Use this command to define a VRRP sync group for a vif on a router.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to assign a vif to a sync group.

Use the **delete** form of the command to remove a vif from a sync group.

Use the **show** form of the command to view sync group configuration for a vif.

interfaces <interface> vif <vlan-id> vrrp vrrp-group <group-id> virtual-address <addr>

Sets the virtual IP address or network address for a VRRP group on a vif.

Syntax

set interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **virtual-address** *addr*

delete interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **virtual-address**

show interfaces *interface* **vif** *vlan-id* **vrrp vrrp-group** *group-id* **virtual-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vif [0-4095] {
      vrrp {
         vrrp-group [1-255] {
            virtual-address text
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
vlan-id	Mandatory. The VLAN ID of a defined vif. The range is 0 to 4095.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
addr	Mandatory. The virtual IP address or network address of the VRRP group. The format is either *ipv4-addr* or *ipv4-addr/prefix*.

Default

None.

Usage Guidelines

Use this command to set the virtual IP address or network address for a VRRP group. Every VRRP group must have a virtual address, and all interfaces and vifs in the VRRP group must be configured with the same virtual address.

The virtual address is “shared” by the VRRP group and is dynamically assigned to the master interface in the group. The master links the virtual address to its own MAC address in the network by issuing a gratuitous ARP to the LAN segment. If the master fails, the group elects a new master, to whom the virtual address is then assigned. The new master notifies the network of the changed MAC address by issuing another gratuitous ARP.

In general, a real interface or vif should not be configured with the virtual address of the VRRP group. In practice, if a real interface is configured with the virtual address, the interface is said to “own” the virtual address. The VRRP standard (RFC 2338) prescribes that a router owning the virtual address should be assigned a priority of 255, which automatically elects the router owning the VIP as master. If you do assign a virtual address to a real interface, set the priority of the interface to 255.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the virtual address of a VRRP group for a vif.

Use the **delete** form of the command to remove the virtual address from the vif. However, note that the virtual address is mandatory in VRRP configuration.

Use the **show** form of the command to view the virtual address configured for a VRRP group on a vif.

interfaces <interface> vrrp vrrp-group <group-id>

Assigns an Ethernet interface to a VRRP group.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id*

delete interfaces *interface* **vrrp vrrp-group** *group-id*

show interfaces *interface* **vrrp vrrp-group** *group-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. Multi-node. An integer uniquely identifying a VRRP group. The range is 1 to 255. The default is 1. You can assign an interface to multiple VRRP groups by creating multiple **vrrp-group** configuration nodes within the **interfaces ethernet** configuration node.

Default

The default is 1.

Usage Guidelines

Use this command to assign a virtual interface to a VRRP group on an interface.

A VRRP group consists of a cluster of interfaces and/or vifs providing redundancy for the primary, or “master,” interface in the group. Redundancy is managed by the VRRP process on the system.

The VRRP group has a unique numeric identifier and is assigned a single virtual IP address (sometimes called a virtual IP or VIP). The virtual address is linked with the MAC address of the master router. If the master router fails, a new master is elected and the new master notifies the network of its MAC address by issuing a gratuitous ARP.

All interfaces in the group must be assigned the same VRRP group identifier and virtual address; otherwise they cannot provide redundancy for one another. Interfaces being mapped to the virtual address must be on the same subnet as the virtual address, but should not have the same address as the virtual address.

An interface or virtual interface can belong to more than one VRRP group.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to assign an interface to a VRRP group.

Use the **delete** form of the command to remove an interface from a VRRP group.

Use the **show** form of the command to view VRRP group configuration settings for an interface

interfaces <interface> vrrp vrrp-group <group-id> advertise-interval <interval>

Sets the advertisement interval for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **advertise-interval** *interval*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **advertise-interval**

show interfaces *interface* **vrrp vrrp-group** *group-id* **advertise-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         advertise-interval 1-255
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
interval	Optional. The interval in seconds between VRRP advertisement packets. All interfaces in this VRRP group must use the same advertisement interval. The range is 1 to 255. The default is 1.

Default

The master router sends VRRP advertisements 1-second intervals.

Usage Guidelines

Use this command to set the interval between VRRP advertisements.

To signal that it is still in service, the master interface or vif sends MAC-level multicast "heartbeat" packets called advertisements to the LAN segment, using the IP address 224.0.0.18, , which is the IPv4 multicast address assigned to VRRP. These advertisements confirm the health of the master to backup routers and contain other VRRP information, such as the master's priority.

If the master fails to send advertisements for some number of intervals, the master is declared out of service, and the VRRP process triggers failover to the backup interface. In this case, the backup interface with the highest priority is elected as the new master. The new master assumes the virtual address and notifies the network of its MAC address by issuing a gratuitous ARP message.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to set the VRRP advertise interval for a VRRP group on an interface.

Use the **delete** form of the command to restore the default value for VRRP advertise interval for a VRRP group on an interface.

Use the **show** form of the command to view VRRP advertise interval configuration.

interfaces <interface> vrrp vrrp-group <group-id> authentication password

Sets the VRRP authentication password for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **authentication password** *pwd*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **authentication password**

show interfaces *interface* **vrrp vrrp-group** *group-id* **authentication password**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         authentication {
            password text
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
pwd	Mandatory. The password the interface will use to authenticate itself as a member of the VRRP group.

Default

Interfaces are not required to authenticate themselves to the VRRP group.

Usage Guidelines

Use this command to set a password for VRRP authentication on an interface.

If a password is set for VRRP authentication, the authentication type (AH or plaintext-password) must also be defined. If the password is set and authentication type is not defined, the system will generate an error when you try to commit the configuration.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify a VRRP authentication password for a VRRP group on an interface.

Use the **delete** form of the command to delete the VRRP authentication password.

- You cannot delete the VRRP password without also deleting the VRRP authentication type. If you do, the system will generate an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the interface.

Use the **show** form of the command to view the VRRP authentication password for a VRRP group on an interface.

interfaces <interface> vrrp vrrp-group <group-id> authentication type

Specifies the VRRP authentication type for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **authentication type** *type*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **authentication type**

show interfaces *interface* **vrrp vrrp-group** *group-id* **authentication type**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         authentication {
            type {
               ah
               plaintext-password
            }
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
pwd	Mandatory. The plaintext password the interface will use to authenticate itself as a member of the group.

type	The type of authentication to be used. Supported values are as follows: **ah**: The IP Authentication Header (AH) protocol is used. **plaintext-password**: Plain-text password authentication is used.

Default

Interfaces are not required to authenticate themselves to the VRRP group.

Usage Guidelines

Use this command to set the authentication type for VRRP authentication on an interface.

If the authentication type is set for VRRP authentication, a password must also be specified. If the authentication type is defined and a password is not set, the system will generate an error when you try to commit the configuration.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the VRRP authentication type for a VRRP group on an interface.

Use the **delete** form of the command to delete the authentication type.

- You cannot delete the VRRP authentication type without also deleting the VRRP password. If you do, the system will generate an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the interface.

Use the **show** form of the command to view the VRRP authentication password for a VRRP group on an interface.

interfaces <interface> vrrp vrrp-group <group-id> description <desc>

Specifies a description for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **description** *desc*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **description**

show interfaces *interface* **vrrp vrrp-group** *group-id* **description**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         description text
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
desc	A description for the VRRP group on a vif.

Default

Preemption is enabled.

Usage Guidelines

Use this command to provide a description for the VRRP group.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to provide a description for the VRRP.

Use the **delete** form of the command to provide a description for the VRRP.

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

interfaces <interface> vrrp vrrp-group <group-id> disable

Disables a VRRP group.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **disable**

delete interfaces *interface* **vrrp vrrp-group** *group-id* **disable**

show interfaces *interface* **vrrp vrrp-group** *group-id*

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         disable
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.

Default

The VRRP group is enabled.

Usage Guidelines

Use this command to disable a VRRP group.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to disable a VRRP group.

Use the **delete** form of the command to reenable a VRRP group.

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

interfaces <interface> vrrp vrrp-group <group-id> hello-source-address <addr>

Specifies the source address for VRRP hello packets.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **hello-source-address** *addr*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **hello-source-address** *addr*

show interfaces *interface* **vrrp vrrp-group** *group-id* **hello-source-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         hello-source-address ipv4
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
addr	Mandatory. The IP address to use as the VRRP source address when sending VRRP hello packets. The format is *ipv4-addr*. The address must already be defined on an interface.

Default

The IP address of the interface is used as the source for VRRP hello packets.

Usage Guidelines

Use this command to specify the source address for VRRP hello packets. This is typically used when an address other than the default address for the interface is required. Note that the address must be defined on an interface already

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the source address for VRRP hello packets.

Use the **delete** form of the command to restore the default source address.

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

interfaces <interface> vrrp vrrp-group <group-id> preempt <preempt>

Enables or disables preemption for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **preempt** *preempt*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **preempt**

show interfaces *interface* **vif vrrp vrrp-group** *group-id* **preempt**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         preempt [true|false]
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
preempt	Optional. Allows a higher-priority VRRP backup router to assert itself as master over a lower-priority master router. Supported values are as follows: **true**: Allow the master router to be preempted by a backup router with higher priority. **false**: Do not allow the master router to be preempted by a backup router with higher priority. The default is **true**; that is, the master router can be preempted by a backup router with higher priority.

Default

Preemption is enabled.

Usage Guidelines

Use this command to enable or disable preemption on an interface.

If preemption is enabled, a backup router with a higher priority than the current master will “preempt” the master, and become the master itself.

A backup router preempts the master by beginning to send out its own VRRP advertisements. The master router examines these, and discovers that the backup router has a higher priority than itself. The master then stops sending out advertisements, while the backup continues to send, thus making itself the new master.

Preemption is useful in situation where a lower-performance backup router becomes master when a higher-performance router fails. In this case, a new higher-performance router can be brought online, and it will automatically preempt the lower-performance backup.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to enable or disable VRRP preemption on an interface.

Use the **delete** form of the command to restore the default value for VRRP preemption on an interface.

Use the **show** form of the command to view VRRP preemption configuration on an interface.

interfaces <interface> vrrp vrrp-group <group-id> preempt-delay <delay>

Sets the preemption delay for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **preempt-delay** *delay*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **preempt-delay**

show interfaces *interface* **vif vrrp vrrp-group** *group-id* **preempt-delay**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         preempt-delay 0-3600
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
delay	The amount of time to postpone preemption, in seconds. The range is 0 to 3600 (1 hour), where 0 means no delay. The default is 0.

Default

A router preempting another router does not wait.

Usage Guidelines

Use this command to set the preemption delay on an interface. The preemption delay is the amount of time a router must wait before preempting a lower-priority VRRP router and becoming the master.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to set the preemption delay.

Use the **delete** form of the command to restore the default value preemption delay

Use the **show** form of the command to view preemption delay configuration on an interface.

interfaces <interface> vrrp vrrp-group <group-id> priority <priority>

Sets the priority of an interface within a VRRP group.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **priority** *priority*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **priority**

show interfaces *interface* **vrrp vrrp-group** *group-id* **priority**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         priority [1-255]
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
priority	Mandatory. The priority with which this interface should be considered for election as master within the VRRP group. The higher the configured number, the higher the priority. The range for a VRRP backup router is from 1 to 254. The VRRP master router must have the highest priority, and typically has a priority of 255. The default is 1.

Default

The default is 1.

Usage Guidelines

Use this command to set the VRRP priority of a real interface. This determines the likelihood of its being elected the master router in a cluster of VRRP routers.

The master interface in the VRRP group is elected master based on its priority, where the higher the configured number, the higher the priority. If the master interface fails, the interface with the next highest priority is elected master and assumes the virtual address of the group. The new master notifies the network of its MAC address by sending out a gratuitous ARP message.

The priority of the master interface is typically set to 255. The backup interface can be left with the default priority; however, if more than one interface is acting as backup, they should be configured with different priorities.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the VRRP group priority for the interface.

Use the **delete** form of the command to remove the VRRP group priority from the interface.

Use the **show** form of the command to view the VRRP group priority for the interface.

interfaces <interface> vrrp vrrp-group <group-id> run-transition-scripts

Specify a script to run on VRRP state transition.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**] *script*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**]

show interfaces *interface* **vrrp vrrp-group** *group-id* **run-transition-scripts [backup** | **fault** | **master**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         run-transition-scripts {
            backup text
            fault text
            master text
         }
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
backup *script*	The name of the executable script to run during VRRP state transition to the backup state.
fault *script*	The name of the executable script to run during VRRP state transition to the fault state.

master *script*	The name of the executable script to run during VRRP state transition to the master state.

Default

None.

Usage Guidelines

Use this command to specify a script to run when the VRRP group on the specified interface changes state. The state is either backup, fault, or master.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify a script to run when the VRRP group on the specified interface changes state.

Use the **delete** form of the command to stop the script from being run when the specifed state transition occurs.

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

interfaces <interface> vrrp vrrp-group <group-id> sync-group <group>

Assigns an interface to a VRRP sync group.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **sync-group** *group*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **sync-group**

show interfaces *interface* **vrrp vrrp-group** *group-id* **sync-group**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         sync-group text
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
group	A text string defining the name of a sync group.

Default

None.

Usage Guidelines

Use this command to define a VRRP sync group for an interface on a router.

Interfaces in a sync group are synchronized such that, if one of the interfaces in the group fails over to backup, all interfaces in the group fail over to backup.

For example, in many cases, if one interface on a master router fails, the whole router should fail over to a backup router. By assigning all the interfaces on the master to a sync group, the failure of one interface will trigger a failover of all the interfaces in the sync group to the backup configured for the interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to assign an interface to a sync group.

Use the **delete** form of the command to remove an interface from a sync group.

Use the **show** form of the command to view sync group configuration for an interface.

interfaces <interface> vrrp vrrp-group <group-id> virtual-address <addr>

Sets the virtual IP address or network address for a VRRP group on an interface.

Syntax

set interfaces *interface* **vrrp vrrp-group** *group-id* **virtual-address** *addr*

delete interfaces *interface* **vrrp vrrp-group** *group-id* **virtual-address**

show interfaces *interface* **vrrp vrrp-group** *group-id* **virtual-address**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   vrrp {
      vrrp-group [1-255] {
         virtual-address text
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in the Usage Guidelines below.
group-id	Mandatory. The VRRP group being configured. The range is 1 to 255.
addr	Mandatory. The virtual IP address or network address of the VRRP group. The format is either *ipv4-addr* or *ipv4-addr/prefix*.

Default

None.

Usage Guidelines

Use this command to set the virtual IP address or network address for a VRRP group. Every VRRP group must have a virtual address, and all interfaces and vifs in the VRRP group must be configured with the same virtual address.

The virtual address is “shared” by the VRRP group and is dynamically assigned to the master interface in the group. The master links the virtual address to its own MAC address in the network by issuing a gratuitous ARP to the LAN segment. If the master fails, the group elects a new master, to whom the virtual address is then assigned. The new master notifies the network of the changed MAC address by issuing another gratuitous ARP.

In general, a real interface or vif should not be configured with the virtual address of the VRRP group. In practice, if a real interface is configured with the virtual address, the interface is said to “own” the virtual address. The VRRP standard (RFC 2338) prescribes that a router owning the virtual address should be assigned a priority of 255, which automatically elects the router owning the VIP as master. If you do assign a virtual address to a real interface, set the priority of the interface to 255.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*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.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Use the **set** form of the command to specify the virtual address of a VRRP group for an interface.

Use the **delete** form of the command to remove the virtual address from the interface. However, note that the virtual address is mandatory in VRRP configuration.

Use the **show** form of the command to view the virtual address configured for a VRRP group on an interface.

show vrrp

Displays information about VRRP groups.

Syntax

show vrrp [**interface** *eth0..eth23* [**group** *group-name*] | **summary**]

Command Mode

Operational mode.

Parameters

eth0..eth23	Shows VRRP information for the specified interface.
group-name	Shows VRRP information for the specified interface and group.
summary	Shows a summary of VRRP information.

Default

Displays information about all groups on all interfaces.

Usage Guidelines

Use this command to see information about VRRP groups, including current VRRP elections and statistics.

Examples

Example 2-4 shows output for **show vrrp summary** command.

Example 2-4 “show vrrp summary”: Showing VRRP summary information

```
vyatta@R1> show vrrp summary
VRRP    Addr                    Interface       VRRP
Interface       Group   Type    Address         State           State
---------       -----   ----    -------         -----           -----
eth1.10         110     int     192.168.10.128  down            fault
                        vip     192.168.10.100
eth1            10      int     192.168.74.128  down            fault
                        vip     192.168.74.100
eth2.20         220     int     172.16.20.128   up              master
                        vip     172.16.20.100
```

Chapter 3: Clustering

This chapter explains clustering for high availability on the Vyatta system.

This chapter presents the following topics:

- Clustering Configuration
- Clustering Commands

Clustering Configuration

This section presents the following topics:

- Clustering Overview
- Clustering Configuration Examples

Clustering Overview

This section presents the following topics:

- Components of a Cluster
- Failure Detection in a Cluster
- Clustering Heartbeat Mechanism
- IP Addressing in Clusters
- Revertive and Non-Revertive Failover

On the Vyatta system, clustering can be used as a failover mechanism to provide high availability (HA) for mission-critical services. The cluster monitors the nodes providing designated services (for example, an IPsec VPN tunnel) at a designated address. If the system detects that the node has failed, or that the link to the node has failed, the system migrates both the services and the IP addresses to a backup node.

Failover is currently supported between two nodes: a primary node and a secondary node.

Components of a Cluster

There are three types of nodes in a cluster:

- **The primary cluster node.** This is the “active” router in the cluster; it is the router initially providing the service. For example, in a scenario with redundant VPN tunnels, this is the router initially operating as the local endpoint of the VPN tunnel.
- **A secondary cluster node.** This is the “backup” router in the cluster. It is the router to which the cluster fails over if the primary cluster node fails. Currently, only one secondary node is supported.
- **Monitor nodes.** The primary and secondary nodes monitor their own network connectivity by “pinging” devices upstream/downstream on the network. These devices are called “monitor nodes.”

  Monitor nodes themselves do not actively participate in the clustering; the only requirement for a monitor node is that it must respond to ICMP Echo Request messages (ping). Communication between monitor nodes and the cluster devices uses the IP addresses applied to the physical interfaces of the cluster devices. This is distinct from the cluster IP addresses, but must be on the same subnet.

A cluster provides failover for two types of resources:

- **Cluster IP addresses.** These are IP addresses that are “shared” between the redundant nodes. Initially, these IP addresses are assigned to the primary node. If the primary node fails over, the system migrates the cluster IP addresses to the secondary node.

  Note that, in the cluster model, cluster IP addresses are considered “services.” When the system fails over, the IP address “services” are “started up” on the secondary node along with other services.

  In addition to cluster IP addresses, the interfaces used for clustering must be configured with a separate IP address on the same subnet for communicating with monitor nodes.
- **Services.** The set of things to be made redundant. Together with the cluster IP addresses, the currently supported service is ipsec, which provides redundancy for IPsec VPN tunnels.

These cluster nodes and resources are specified as a *resource group*. Currently, only one resource group is supported.

Failure Detection in a Cluster

A cluster can respond to two kinds of failure:

- **Node failure.** The primary and the secondary cluster nodes exchange regular heartbeat messages through their network interfaces. If a cluster node does not receive a heartbeat message from its peer within a certain interval, it considers the peer to be dead. If the secondary node determines that the primary peer is dead, the secondary node triggers the failover process and takes over the cluster resources.
- **Connectivity failure.** The primary and secondary nodes monitor their own network connectivity by “pinging” the specified monitor nodes. Failover is triggered when connectivity is lost. For example, if the primary node can no longer reach one of the monitor nodes, it considers itself down and triggers the failover process so that the secondary node can take over the cluster resources.

Clustering Heartbeat Mechanism

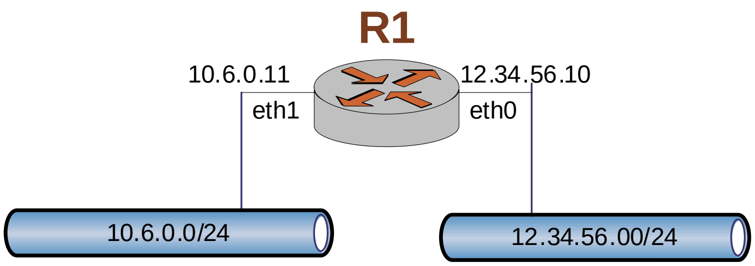
As soon as configuration is committed on a cluster node, the node begins sending heartbeat messages. By default the heartbeat mechanism waits for 120 seconds for the other cluster node to start up.

- If heartbeat messages are received from the other node within this interval, the services listed in the cluster resource group are started on the primary node, and the secondary node becomes an active standby.
- If heartbeat messages are not received from the cluster peer node within this interval, the node with the functioning heartbeat “acquires” the services specified in the resource group configuration and assumes control.

IP Addressing in Clusters

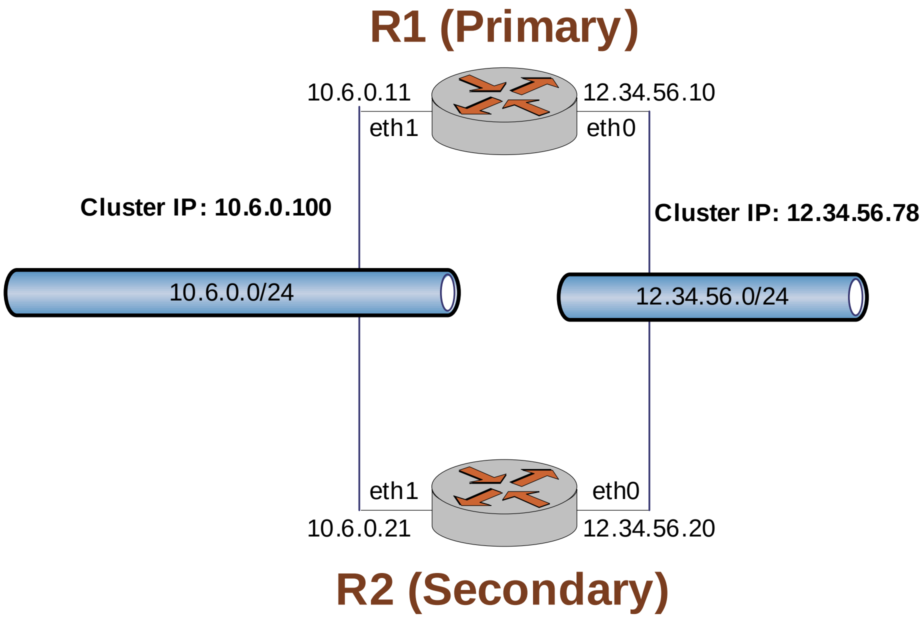
In a non-redundant scenario, IP addresses are assigned to network interfaces or vifs, as shown in Figure 3-1.

Figure 3-1 Explicitly configured IP addresses



In a cluster, cluster IP addresses are “shared” between the two cluster nodes, as shown in Figure 3-2. These are distinct from IP addresses configured for the physical Ethernet interfaces. They must be different from the IP addresses configured for the interface, but d must reside within the same subnet.

Figure 3-2 Clustered IP addresses



Initially, the primary node “owns” the cluster IP addresses. When the heartbeat mechanism starts the services on the primary cluster node, it creates alias interfaces for the cluster IP addresses. For example, on router R1 the heartbeat mechanism would create an alias interface eth0:0 with IP address 12.34.56.78 and an alias interface eth1:0 with IP address 10.6.0.100.

If router R1 fails, the heartbeat mechanism creates the same alias interfaces on the secondary cluster node R2.

NOTE *Cluster IP addresses are started and stopped automatically and dynamically by the system. This means that those addresses must not be explicitly configured for any interfaces on the cluster nodes.*

Revertive and Non-Revertive Failover

Failover can be revertive or non-revertive. If revertive failover (also called “auto-failback”) is configured, the system will fail back from the secondary node to the primary if the primary recovers. If non-revertive failover is configured, the secondary node will remain active even if the primary node recovers.

By default, auto-failback is disabled (that is, failover is non-revertive).

Clustering Configuration Examples

This section presents the following topics:

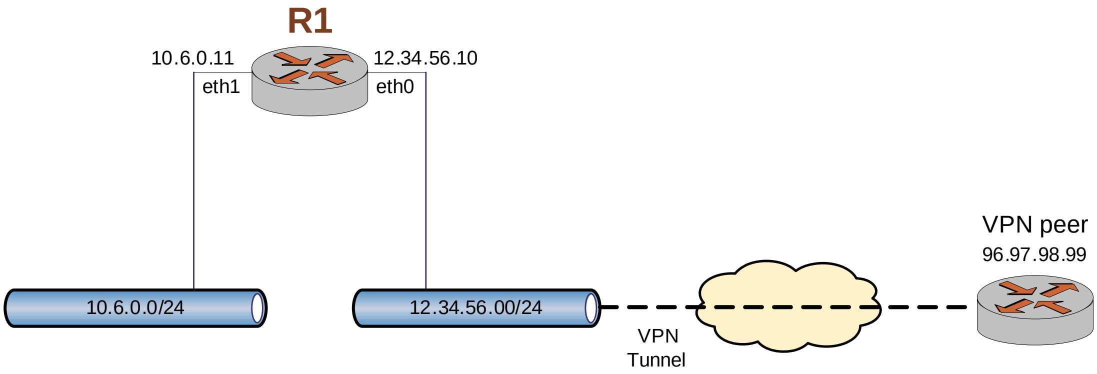
- Defining a Site-to-Site VPN Configuration
- Defining the Cluster on Router R1
- Defining the Cluster on Router R2

This section describes a scenario where failover is required for IPsec VPN tunnels between a local site and a remote VPN peer.

NOTE *The Site-to-site VPN configuration should be set up prior to setting up the cluster configuration.*

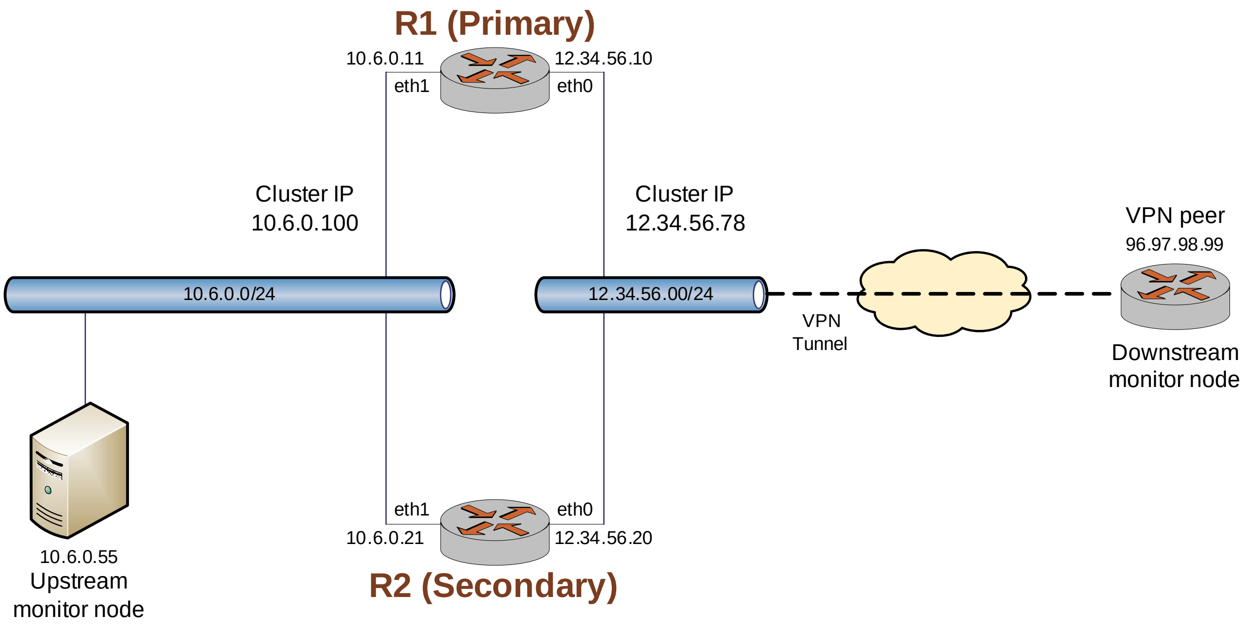
In the non-redundant setup, the VPN tunnel is terminated at the near end by router R1 on interface 12.34.56.10 and at the far end by the remote VPN peer on interface 96.97.98.99, as shown in Figure 3-3.

Figure 3-3 Non-redundant VPN tunnel



To provide redundancy for router R1, we would define the cluster shown Figure 3-4.

Figure 3-4 HA cluster for VPN tunnel failover



In this scenario:

- Routers R1 and R2 are the cluster nodes: R1 is the primary node and R2 is the secondary node.
- The cluster IP addresses are 10.6.0.100 and 12.34.56.78. As in all clusters, these cluster IP addresses are each considered a “service.” The IPsec process managing the VPN tunnels on the router is the third “service” in the cluster.

- The host at 10.6.0.55, which is a reliable host in the upstream network, is the upstream monitor node. This host will be used by the cluster nodes to test upstream connectivity.
- The remote VPN peer is the downstream monitor node. This peer will be used by the cluster to test downstream connectivity.

This deployment allows detection of both node failure and network connectivity failure.

Under normal operational conditions, all three services (the two cluster IP addresses and the IPsec process) run on the primary node, R1. The VPN tunnel is established and maintained between the cluster IP address 12.34.56.78 and the VPN peer on IP address 96.97.98.99. If the primary node fails, or connectivity is lost between the primary node and either of the monitor nodes, the system detects the failure and migrates the two cluster IP addresses and the IPsec process to R2, minimizing service disruption. After failover, router R2 “owns” the cluster IP addresses and establishes and maintains the VPN tunnel with the peer 96.97.98.99.

Defining a Site-to-Site VPN Configuration

When a VPN tunnel is created within a high availability cluster the cluster IP address is used as the local IP address for the peer. This is in contrast to a non-clustered tunnel, where the IP address configured for the physical interface is used as the local IP address for the tunnel.

Note that, in addition to the cluster IP, an IP address must be configured independently for the physical Ethernet interface, so that the cluster node can ping the monitor nodes. (This configuration is not shown in this example.)

Example 3-1 sets up a VPN endpoint for router R1.

To configure the VPN endpoint on R1, perform the following steps in configuration mode:

Example 3-1 Defining a VPN on router R1

Step	Command

Enable VPN on eth0 on R1.	`vyatta@R1# set vpn ipsec ipsec-interfaces interface eth0`
	`[edit]`

Do not copy the ToS byte into the header of the encapsulated packet.	`vyatta@R1# set vpn ipsec copy-tos disable`
	`[edit]`

Create the configuration node for proposal 1 of IKE group VYATTA.	`vyatta@R1# set vpn ipsec ike-group VYATTA proposal 1`
	`[edit]`

Set the encryption cipher for proposal 1.	`vyatta@R1# set vpn ipsec ike-group VYATTA proposal 1 encryption 3des`
	`[edit]`

Example 3-1 Defining a VPN on router R1

Set the hash algorithm for proposal 1.	vyatta@R1# **set vpn ipsec ike-group VYATTA proposal 1 hash sha1** [edit]
Set the lifetime for the whole IKE group.	vyatta@R1# **set vpn ipsec ike-group VYATTA lifetime 28800** [edit]
Use IKE main mode.	vyatta@R1# **set vpn ipsec ike-group VYATTA aggressive-mode disable** [edit]
Set IKE keep-alive message interval.	vyatta@R1# **set vpn ipsec ike-group VYATTA dead-peer-detection interval 30** [edit]
Non-response timeout before **action** will be taken.	vyatta@R1# **set vpn ipsec ike-group VYATTA dead-peer-detection timeout 90** [edit]
Action to take on timeout.	vyatta@R1# **set vpn ipsec ike-group VYATTA dead-peer-detection action clear** [edit]
Create the configuration node for proposal 1 of ESP group VYATTA.	vyatta@R1# **set vpn ipsec esp-group VYATTA proposal 1** [edit]
Set the encryption cipher for proposal 1.	vyatta@R1# **set vpn ipsec esp-group VYATTA proposal 1 encryption 3des** [edit]
Set the hash algorithm for proposal 1.	vyatta@R1# **set vpn ipsec esp-group VYATTA proposal 1 hash sha1** [edit]
Set IPsec connection mode to tunnel.	vyatta@R1# **set vpn ipsec esp-group VYATTA mode tunnel** [edit]
Set the lifetime for the whole ESP group.	vyatta@R1# **set vpn ipsec esp-group VYATTA lifetime 3600** [edit]
Enable Perfect Forward Secrecy.	vyatta@R1# **set vpn ipsec esp-group VYATTA pfs enable** [edit]
Disable compression.	vyatta@R1# **set vpn ipsec esp-group VYATTA compression disable** [edit]
Create the site-to-site node for R1 and set the authentication mode.	vyatta@R1# **set vpn ipsec site-to-site peer 96.97.98.99 authentication mode pre-shared-secret** [edit]

Example 3-1 Defining a VPN on router R1

Navigate to the node for the peer for easier editing.	vyatta@R1# **edit vpn ipsec site-to-site peer 96.97.98.99** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Provide the string that will be used to generate encryption keys.	vyatta@R1# **set authentication pre-shared-secret vyatta** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Specify the IKE group.	vyatta@R1# **set ike-group VYATTA** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Identify the IP address on this router to be used for this connection. ***NOTE*** *The local IP address specified is the Cluster IP.*	vyatta@R1# **set local-ip 12.34.56.78** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Create a tunnel configuration, and provide the local subnet for this tunnel.	vyatta@R1# **set tunnel 1 local-subnet 10.6.0.0/24** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Provide the remote subnet for the tunnel.	vyatta@R1# **set tunnel 1 remote-subnet 10.5.0.0/24** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Do not allow connection to the private network.	vyatta@R1# **set tunnel 1 allow-nat-networks disable** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Do not allow connections to public networks.	vyatta@R1# **set tunnel 1 allow-public-networks disable** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Specify the ESP group for this tunnel.	vyatta@R1# **set tunnel 1 esp-group VYATTA** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Return to the top of the configuration tree.	vyatta@R1# **top** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

Example 3-1 Defining a VPN on router R1

View the configuration.

```
vyatta@R1# show -all vpn
    ipsec {
        ipsec-interfaces {
            interface eth0
        }
        copy-tos disable
        ike-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            lifetime 28800
            aggressive-mode disable
            dead-peer-detection {
                interval 30
                timeout 90
                action clear
            }
        }
        esp-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            mode tunnel
            lifetime 3600
            pfs enable
            compression disable
        }
```

Example 3-1 Defining a VPN on router R1

```
  site-to-site {
        peer 96.97.98.99 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret "vyatta"
            }
            ike-group VYATTA
            local-ip 12.34.56.78
            tunnel 1 {
                local-subnet 10.6.0.0/24
                remote-subnet 10.5.0.0/24
                allow-nat-networks disable
                allow-public-networks disable
                esp-group VYATTA
            }
        }
    }
}
```

Example 3-2 sets up a VPN endpoint for router R2.

To configure the VPN endpoint on R2, perform the following steps in configuration mode:

Example 3-2 Defining a VPN on router R2

Step	Command
Enable VPN on eth0 on R2.	vyatta@R2# **set vpn ipsec ipsec-interfaces interface eth0** [edit]
Do not copy the ToS byte into the header of the encapsulated packet.	vyatta@R2# **set vpn ipsec copy-tos disable** [edit]
Create the configuration node for proposal 1 of IKE group VYATTA.	vyatta@R2# **set vpn ipsec ike-group VYATTA proposal 1** [edit]
Set the encryption cipher for proposal 1.	vyatta@R2# **set vpn ipsec ike-group VYATTA proposal 1 encryption 3des** [edit]
Set the hash algorithm for proposal 1.	vyatta@R2# **set vpn ipsec ike-group VYATTA proposal 1 hash sha1** [edit]

Example 3-2 Defining a VPN on router R2

Set the lifetime for the whole IKE group.	vyatta@R2# **set vpn ipsec ike-group VYATTA lifetime 28800** [edit]
Use IKE main mode.	vyatta@R2# **set vpn ipsec ike-group VYATTA aggressive-mode disable** [edit]
Set IKE keep-alive message interval.	vyatta@R2# **set vpn ipsec ike-group VYATTA dead-peer-detection interval 30** [edit]
Non-response timeout before **action** will be taken.	vyatta@R2# **set vpn ipsec ike-group VYATTA dead-peer-detection timeout 90** [edit]
Action to take on timeout.	vyatta@R2# **set vpn ipsec ike-group VYATTA dead-peer-detection action clear** [edit]
Create the configuration node for proposal 1 of ESP group VYATTA.	vyatta@R2# **set vpn ipsec esp-group VYATTA proposal 1** [edit]
Set the encryption cipher for proposal 1.	vyatta@R2# **set vpn ipsec esp-group VYATTA proposal 1 encryption 3des** [edit]
Set the hash algorithm for proposal 1.	vyatta@R2# **set vpn ipsec esp-group VYATTA proposal 1 hash sha1** [edit]
Set IPsec connection mode to tunnel.	vyatta@R2# **set vpn ipsec esp-group VYATTA mode tunnel** [edit]
Set the lifetime for the whole ESP group.	vyatta@R2# **set vpn ipsec esp-group VYATTA lifetime 3600** [edit]
Enable Perfect Forward Secrecy.	vyatta@R2# **set vpn ipsec esp-group VYATTA pfs enable** [edit]
Disable compression.	vyatta@R2# **set vpn ipsec esp-group VYATTA compression disable** [edit]
Create the site-to-site node for R2 and set the authentication mode.	vyatta@R2# **set vpn ipsec site-to-site peer 96.97.98.99 authentication mode pre-shared-secret** [edit]
Navigate to the node for the peer for easier editing.	vyatta@R2# **edit vpn ipsec site-to-site peer 96.97.98.99** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]

Example 3-2 Defining a VPN on router R2

Provide the string that will be used to generate encryption keys.	vyatta@R2# **set authentication pre-shared-secret vyatta** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Specify the IKE group.	vyatta@R2# **set ike-group VYATTA** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Identify the IP address on this router to be used for this connection. ***NOTE*** *The local IP address specified is the Cluster IP.*	vyatta@R2# **set local-ip 12.34.56.78** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Create a tunnel configuration, and provide the local subnet for this tunnel.	vyatta@R2# **set tunnel 1 local-subnet 10.6.0.0/24** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Provide the remote subnet for the tunnel.	vyatta@R2# **set tunnel 1 remote-subnet 10.5.0.0/24** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Do not allow connection to the private network.	vyatta@R2# **set tunnel 1 allow-nat-networks disable** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Do not allow connections to public networks.	vyatta@R2# **set tunnel 1 allow-public-networks disable** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Specify the ESP group for this tunnel.	vyatta@R2# **set tunnel 1 esp-group VYATTA** [edit vpn/ipsec/site-to-site/peer/96.97.98.99]
Return to the top of the configuration tree.	vyatta@R2# **top** [edit]
Commit the configuration.	vyatta@R2# **commit** [edit]

Example 3-2 Defining a VPN on router R2

View the configuration.

```
vyatta@R2# show -all vpn
    ipsec {
        ipsec-interfaces {
            interface eth0
        }
        copy-tos disable
        ike-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            lifetime 28800
            aggressive-mode disable
            dead-peer-detection {
                interval 30
                timeout 90
                action clear
            }
        }
        esp-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            mode tunnel
            lifetime 3600
            pfs enable
            compression disable
        }
```

Example 3-2 Defining a VPN on router R2

```
  site-to-site {
        peer 96.97.98.99 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret "vyatta"
            }
            ike-group VYATTA
            local-ip 12.34.56.78
            tunnel 1 {
                local-subnet 10.6.0.0/24
                remote-subnet 10.5.0.0/24
                allow-nat-networks disable
                allow-public-networks disable
                esp-group VYATTA
            }
        }
    }
}
```

Example 3-3 sets up a VPN endpoint on the VPN router VPNPeer.

To configure the VPN on VPNPeer, perform the following steps in configuration mode:

Example 3-3 Defining a VPN on router VPNPeer

Step	Command
Enable VPN on eth1 on VPNPeer.	vyatta@VPNPeer# **set vpn ipsec ipsec-interfaces interface eth1** [edit]
Do not copy the ToS byte into the header of the encapsulated packet.	vyatta@VPNPeer# **set vpn ipsec copy-tos disable** [edit]
Create the configuration node for proposal 1 of IKE group VYATTA.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA proposal 1** [edit]
Set the encryption cipher for proposal 1.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA proposal 1 encryption 3des** [edit]
Set the hash algorithm for proposal 1.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA proposal 1** [edit]

Example 3-3 Defining a VPN on router VPNPeer

Set the lifetime for the whole IKE group.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA lifetime 28800** [edit]
Use IKE main mode.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA aggressive-mode disable** [edit]
Set IKE keep-alive message interval.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA dead-peer-detection interval 30** [edit]
Non-response timeout before **action** will be taken.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA dead-peer-detection timeout 90** [edit]
Action to take on timeout.	vyatta@VPNPeer# **set vpn ipsec ike-group VYATTA dead-peer-detection action clear** [edit]
Create the configuration node for proposal 1 of ESP group VYATTA.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA proposal 1** [edit]
Set the encryption cipher for proposal 1.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA proposal 1 encryption 3des** [edit]
Set the hash algorithm for proposal 1.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA proposal 1 hash sha1** [edit]
Set IPsec connection mode to tunnel.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA mode tunnel** [edit]
Set the lifetime for the whole ESP group.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA lifetime 3600** [edit]
Enable Perfect Forward Secrecy.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA pfs enable** [edit]
Disable compression.	vyatta@VPNPeer# **set vpn ipsec esp-group VYATTA compression disable** [edit]

Example 3-3 Defining a VPN on router VPNPeer

Create the site-to-site node for VPNPeer and set the authentication mode. ***NOTE*** *The peer IP address specified is the Cluster IP.*	vyatta@VPNPeer# **set vpn ipsec site-to-site peer 12.34.56.78 authentication mode pre-shared-secret** [edit]
Navigate to the node for the peer for easier editing.	vyatta@VPNPeer# **edit vpn ipsec site-to-site peer 12.34.56.78** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Provide the string that will be used to generate encryption keys.	vyatta@VPNPeer# **set authentication pre-shared-secret vyatta** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Specify the IKE group.	vyatta@VPNPeer# **set ike-group VYATTA** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Identify the IP address on this router to be used for this connection.	vyatta@VPNPeer# **set local-ip 96.97.98.99** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Create a tunnel configuration, and provide the local subnet for this tunnel.	vyatta@VPNPeer# **set tunnel 1 local-subnet 10.5.0.0/24** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Provide the remote subnet for the tunnel.	vyatta@VPNPeer# **set tunnel 1 remote-subnet 10.6.0.0/24** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Do not allow connection to the private network.	vyatta@VPNPeer# **set tunnel 1 allow-nat-networks disable** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Do not allow connections to public networks.	vyatta@VPNPeer# **set tunnel 1 allow-public-networks disable** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Specify the ESP group for this tunnel.	vyatta@VPNPeer# **set tunnel 1 esp-group VYATTA** [edit vpn/ipsec/site-to-site/peer/12.34.56.78]
Return to the top of the configuration tree.	vyatta@VPNPeer# **top** [edit]
Commit the configuration.	vyatta@VPNPeer# **commit** [edit]

Example 3-3 Defining a VPN on router VPNPeer

View the configuration.

```
vyatta@VPNPeer# show -all vpn
    ipsec {
        ipsec-interfaces {
            interface eth1
        }
        copy-tos disable
        ike-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            lifetime 28800
            aggressive-mode disable
            dead-peer-detection {
                interval 30
                timeout 90
                action clear
            }
        }
        esp-group VYATTA {
            proposal 1 {
                encryption 3des
                hash sha1
            }
            mode tunnel
            lifetime 3600
            pfs enable
            compression disable
        }
```

Example 3-3 Defining a VPN on router VPNPeer

```
  site-to-site {
        peer 12.34.56.78 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret "vyatta"
            }
            ike-group VYATTA
            local-ip 96.97.98.99
            tunnel 1 {
                local-subnet 10.5.0.0/24
                remote-subnet 10.6.0.0/24
                allow-nat-networks disable
                allow-public-networks disable
                esp-group VYATTA
            }
        }
    }
}
```

Defining the Cluster on Router R1

Example 3-4 sets up clustering on router R1. In this example:

- Interfaces eth0 and eth1 on R1 are used to exchange heartbeat messages between R1 and R2.
- The pre-shared key for heartbeat authentication is “!secret!”
- The heartbeat interval is 2 seconds (2000 milliseconds).
- The longest allowable interval between heartbeat messages is 10 seconds (10,000 milliseconds). After that period, the peer cluster node is considered dead.
- R1 is the primary node. (“R1” is the configured host name for the router. It is the name that would be returned when the **show host name** command is issued on R1.)
- R2 is the secondary node. (“R2” is the configured host name for the router. It is the name that would be returned when the **show host name** command is issued on R2.)
- The VPN peer at 96.97.98.99 is a monitor node.
- The reliable host at 10.6.0.55 is a monitor node.
- 10.6.0.100 is a cluster IP address, and therefore a cluster service.
- 12.34.56.78 is a cluster IP address, and therefore a cluster service.
- The IPsec process **ipsec** is the cluster service.

- Failover is to be non-revertive. This is the default, and need not be explicitly configured.

This example assumes that IP addresses have already been configured for the Ethernet interfaces eth0 and eth1 on both R1 and R2. This example focuses on cluster-specific configurations.

To configure this cluster on R1, perform the following steps in configuration mode:

Example 3-4 Defining a cluster on router R1

Step	Command

Specify the interfaces to use for heartbeat messages	`vyatta@R1# set cluster interface eth0`
	`[edit]`
	`vyatta@R1# set cluster interface eth1`
	`[edit]`

Provide the pre-shared key for heartbeat authentication.	`vyatta@R1# set cluster pre-shared-secret !secret!`
	`[edit]`

Set the interval between heartbeats.	`vyatta@R1# set cluster keepalive-interval 2000`
	`[edit]`

Set the length of the interval after which the cluster peer is considered dead.	`vyatta@R1# set cluster dead-interval 10000`
	`[edit]`

Create the resource group.	`vyatta@R1# set cluster group cluster1`
	`[edit]`

Specify the primary node in the cluster.	`vyatta@R1# set cluster group cluster1 primary R1`
	`[edit]`

Specify the secondary node in the cluster.	`vyatta@R1# set cluster group cluster1 secondary R2`
	`[edit]`

Specify the downstream monitor node.	`vyatta@R1# set cluster group cluster1 monitor 96.97.98.99`
	`[edit]`

Specify the upstream monitor node.	`vyatta@R1# set cluster group cluster1 monitor 10.6.0.55`
	`[edit]`

List both cluster IP addresses as services to fail over in the event of failure.	`vyatta@R1# set cluster group cluster1 service 10.6.0.100/24`
	`[edit]`
	`vyatta@R1# set cluster group cluster1 service 12.34.56.78/24`
	`[edit]`

Example 3-4 Defining a cluster on router R1

List the **ipsec** process as a service to fail over in the event of failure.	vyatta@R1# **set cluster group cluster1 service ipsec** [edit]
Commit the configuration.	vyatta@R1# **commit** [edit]

View the configuration.	vyatta@R1# **show cluster**
	      interface eth0
	      interface eth1
	      pre-shared-secret "!secret!"
	      keepalive-interval 2000
	      dead-interval 10000
	      group cluster1 {
	         primary R1
	         secondary R2
	         monitor 96.97.98.99
	         monitor 10.6.0.55
	         service 10.6.0.100/24
	         service 12.34.56.78/24
	         service ipsec
	      }
	vyatta@R1#

Defining the Cluster on Router R2

Example 3-5 sets up clustering on router R2. Note that the commands in this example are identical to those used to configure R1.

To configure this cluster on R2, perform the following steps in configuration mode:

Example 3-5 Defining a cluster on router R2

Step	Command
Specify the interfaces to use for heartbeat messages	vyatta@R2# **set cluster interface eth0** [edit] vyatta@R2# **set cluster interface eth1** [edit]
Provide the pre-shared key for heartbeat authentication.	vyatta@R2# **set cluster pre-shared-secret !secret!** [edit]
Set the interval between heartbeats.	vyatta@R2# **set cluster keepalive-interval 2000** [edit]

Example 3-5 Defining a cluster on router R2

Set the length of the interval after which the cluster peer is considered dead.	vyatta@R2# **set cluster dead-interval 10000** [edit]
Create the resource group.	vyatta@R2# **set cluster group cluster1** [edit]
Specify the primary node in the cluster.	vyatta@R2# **set cluster group cluster1 primary R1** [edit]
Specify the secondary node in the cluster.	vyatta@R2# **set cluster group cluster1 secondary R2** [edit]
Specify the downstream monitor node.	vyatta@R2# **set cluster group cluster1 monitor 96.97.98.99** [edit]
Specify the upstream monitor node.	vyatta@R2# **set cluster group cluster1 monitor 10.6.0.55** [edit]
List both cluster IP addresses as services to fail over in the event of failure.	vyatta@R2# **set cluster group cluster1 service 10.6.0.100/24** [edit] vyatta@R2# **set cluster group cluster1 service 12.34.56.78/24** [edit]
List the **ipsec** process as a service to fail over in the event of failure.	vyatta@R2# **set cluster group cluster1 service ipsec** [edit]
Commit the configuration.	vyatta@R2# **commit** [edit]

View the configuration.	vyatta@R2# **show cluster**
	      interface eth0
	      interface eth1
	      pre-shared-secret: "!secret!"
	      keepalive-interval 2000
	      dead-interval 10000
	      group cluster1 {
	         primary R1
	         secondary R2
	         monitor 96.97.98.99
	         monitor 10.6.0.55
	         service 10.6.0.100/24
	         service 12.34.56.78/24
	         service ipsec
	      }
	vyatta@R2#

Clustering Commands

This section presents the following commands.

Configuration Commands	
Clusters	
cluster	Enables clustering for high availability.
cluster dead-interval <interval>	Defines the time after which a cluster peer is considered dead.
cluster interface <interface>	Defines a interface over which heartbeat messages will be sent.
cluster keepalive-interval <interval>	Defines the time interval between heartbeat messages.
cluster mcast-group <ipv4>	Defines the multicast group for sending and receiving heartbeat messages.
cluster pre-shared-secret <secret>	Defines the shared key for heartbeat authentication.
Cluster Groups	
cluster group <group>	Defines a cluster resource group.
cluster group <group> auto-failback <mode>	Specifies whether or not the system should revert back to the primary node should the primary node become available again.
cluster group <group> monitor <ipv4>	Defines a monitor node for a cluster resource group.
cluster group <group> primary <hostname>	Specifies the host name configured for the primary node in the cluster.
cluster group <group> secondary <hostname>	Specifies the host name configured for the secondary node in the cluster.
cluster group <group> service <service>	Specifies the services that will be started on the primary and secondary nodes.
Operational Commands	
show cluster status	Displays current clustering status.

cluster

Enables clustering for high availability.

Syntax

set cluster

delete cluster

show cluster

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
}
```

Parameters

None.

Default

None.

Usage Guidelines

Use this command to specify a cluster configuration.

Use the **set** form of this command to create the cluster configuration.

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

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

cluster dead-interval <interval>

Defines the time after which a cluster peer is considered dead.

Syntax

set cluster dead-interval *interval*

delete cluster dead-interval

show cluster dead-interval

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   dead-interval u32
}
```

Parameters

interval	The time, in milliseconds, after which if a heartbeat message is not received from the cluster peer node, the peer is considered dead. This triggers the failover procedure and all services are moved to the secondary node.The default is 20000 (20 seconds).

Default

A peer is considered dead after not sending a heartbeat after 20 seconds.

Usage Guidelines

Use this command to specify the dead interval in a cluster configuration.

Use the **set** form of this command to create the dead interval in a cluster configuration.

Use the **delete** form of this command to remove the dead interval in a cluster configuration.

Use the **show** form of this command to view the dead interval in a cluster configuration.

cluster group <group>

Defines a cluster resource group.

Syntax

set cluster group *group*

delete cluster group *group*

show cluster group *group*

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
   }
}
```

Parameters

group	The name of the cluster group.

Default

None.

Usage Guidelines

Use this command to define the resources and clustering behavior associated with a cluster group. Currently only one group is supported.

Use the **set** form of this command to create the cluster resource group configuration.

Use the **delete** form of this command to remove the cluster resource group configuration.

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

cluster group <group> auto-failback <mode>

Specifies whether or not the system should revert back to the primary node should the primary node become available again.

Syntax

set cluster group *group* **auto-failback** *mode*

delete cluster group *group* **auto-failback**

show cluster group *group* **auto-failback**

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
      auto-failback [true | false]
   }
}
```

Parameters

group	The name of the cluster group.
mode	Specifies whether the system should revert back to the primary node, should it become available again. Supported values are as follows: **true**: Failover is revertive. Migrate back to the primary node if it recovers. **false**: Failover is non-revertive. Do not migrate back to the primary node even if it recovers.

Default

The default is **false**.

Usage Guidelines

Use this command to specify the auto failback mode in a cluster resource group configuration.

Use the **set** form of this command to create the auto failback mode in a cluster resource group configuration.

Use the **delete** form of this command to remove the auto failback mode in a cluster resource group configuration.

Use the **show** form of this command to view the auto failback mode in a cluster resource group configuration.

cluster group <group> monitor <ipv4>

Defines a monitor node for a cluster resource group.

Syntax

set cluster group *group* **monitor** *ipv4*

delete cluster group *group* **monitor** *ipv4*

show cluster group *group* **monitor**

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
      monitor ipv4
   }
}
```

Parameters

group	The name of the cluster group.
ipv4	Multi-node. The IP address of a monitor node. Monitor nodes are used within the cluster to confirm network connectivity. Note that communication between monitor nodes and the cluster devices takes place using the IP addresses configured for the physical interfaces in the cluster, not the cluster IP addresses. You can define more than one monitor node by creating multiple **monitor** configuration nodes.

Default

None.

Usage Guidelines

Use this command to specify the IP address of a monitor node in a cluster resource group configuration.

Use the **set** form of this command to create the IP address of a monitor node in a cluster resource group configuration.

Use the **delete** form of this command to remove the IP address of a monitor node in a cluster resource group configuration.

Use the **show** form of this command to view the IP address of a monitor node in a cluster resource group configuration.

cluster group <group> primary <hostname>

Specifies the host name configured for the primary node in the cluster.

Syntax

set cluster group *group* **primary** *hostname*

delete cluster group *group* **primary**

show cluster group *group* **primary**

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
      primary text
   }
}
```

Parameters

group	The name of the cluster group.
hostname	Mandatory. The host name configured for the primary node in the cluster. Enter the host name exactly as configured for the device. You can view this by issuing the **show host name** command on the primary (active) node.

Default

None.

Usage Guidelines

Use this command to specify the host name of for the primary node in a cluster resource group configuration.

Use the **set** form of this command to create the host name of for the primary node in a cluster resource group configuration.

Use the **delete** form of this command to remove the host name of for the primary node in a cluster resource group configuration.

Use the **show** form of this command to view the host name of for the primary node in a cluster resource group configuration.

cluster group <group> secondary <hostname>

Specifies the host name configured for the secondary node in the cluster.

Syntax

set cluster group *group* **secondary** *hostname*

delete cluster group *group* **secondary**

show cluster group *group* **secondary**

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
      secondary text
   }
}
```

Parameters

group	The name of the cluster group.
hostname	Mandatory. The host name configured for the secondary node in the cluster. Enter the host name exactly as configured for the device. You can view this by issuing the **show host name** command on the secondary (backup) node. Currently, only one secondary node is supported.

Default

None.

Usage Guidelines

Use this command to specify the host name of for the secondary node in a cluster resource group configuration.

Use the **set** form of this command to create the host name of for the secondary node in a cluster resource group configuration.

Use the **delete** form of this command to remove the host name of for the secondary node in a cluster resource group configuration.

Use the **show** form of this command to view the host name of for the secondary node in a cluster resource group configuration.

cluster group <group> service <service>

Specifies the services that will be started on the primary and secondary nodes.

Syntax

set cluster group *group* **service** *service*

delete cluster group *group* **service** *service*

show cluster group *group* **service**

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   group text {
      service text
   }
}
```

Parameters

group	The name of the cluster group.
service	Mandatory. Multi-node. The services that will be started on the primary node initially and will be restarted on the secondary node when failover occurs. The following formats are supported: • *ip-address/prefix* [*if-name* [*broadcast-address*]], where ip-address/prefix is the network address of the cluster, *if-name* is the interface to which the address is to be added, and *broadcast-address* is the broadcast address for the cluster. • *service*, where service is a script in /etc/init.d. • *script*::*args*, where script is a script in /etc/ha.d/resource.d and args are the arguments for the script. You can define more than one service node by creating multiple **service** configuration nodes. At least one service must be specified.

Default

None.

Usage Guidelines

Use this command to specify the services that will be started on the primary and secondary nodes in a cluster resource group configuration.

A service can be:

- An IP address/network prefix specification. IP addresses supplied as a service are used as cluster IP addresses.

  The cluster IP address is distinct from the IP address configured for the physical interface. Cluster IP addresses are applied to the cluster interfaces by the clustering mechanism. You do not explicitly apply the cluster IP address to the interface.
- A script as defined in the file **/etc/init.d**, in the form *script-name*.
- A script as defined in the file **/etc/ha.d/resource.d**, with arguments, in the form *script-name*::*args*.
- A cluster IP address/prefix length with two optional parameters: the interface to which this address will be added and the broadcast address.

Use the **set** form of this command to specify the services that will be started on the primary and secondary nodes in a cluster resource group configuration.

Use the **delete** form of this command to remove the services that will be started on the primary and secondary nodes in a cluster resource group configuration.

Use the **show** form of this command to view the services that will be started on the primary and secondary nodes in a cluster resource group configuration.

cluster interface <interface>

Defines a interface over which heartbeat messages will be sent.

Syntax

set cluster interface *interface*

delete cluster interface *interface*

show cluster interface

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   interface text
}
```

Parameters

interface	Mandatory. Multi-node. The name of the interface over which heartbeat messages will be sent to the peer cluster node. You can assign more than one interface to the cluster by creating multiple **interface** configuration nodes.

Default

None.

Usage Guidelines

Use this command to specify the interface over which heartbeat messages will be sent in a cluster configuration.

Use the **set** form of this command to specify the interface over which heartbeat messages will be sent in a cluster configuration.

Use the **delete** form of this command to remove the interface over which heartbeat messages will be sent in a cluster configuration.

Use the **show** form of this command to view the interface over which heartbeat messages will be sent in a cluster configuration.

cluster keepalive-interval <interval>

Defines the time interval between heartbeat messages.

Syntax

set cluster keepalive-interval *interval*

delete cluster keepalive-interval

show cluster keepalive-interval

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   keepalive-interval u32
}
```

Parameters

interval	The time interval between heartbeat messages, in milliseconds. The default is 5000 (5 seconds).

Default

The default is 5000.

Usage Guidelines

Use this command to specify the keepalive interval in a cluster configuration.

Use the **set** form of this command to create the keepalive interval in a cluster configuration.

Use the **delete** form of this command to remove the keepalive interval in a cluster configuration.

Use the **show** form of this command to view the keepalive interval in a cluster configuration.

cluster mcast-group <ipv4>

Defines the multicast group for sending and receiving heartbeat messages.

Syntax

set cluster mcast-group *ipv4*

delete cluster mcast-group

show cluster mcast-group

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   mcast-group ipv4
}
```

Parameters

ipv4	The IP address of the multicast group used for sending and receiving heartbeat messages.

Default

The default is **239.251.252.253**.

Usage Guidelines

Use this command to specify the multicast group for sending and receiving heartbeat messages. Typically it will only be changed if the default group conflicts with your network setup.

Use the **set** form of this command to create the multicast group for sending and receiving heartbeat messages.

Use the **delete** form of this command to remove the multicast group for sending and receiving heartbeat messages.

Use the **show** form of this command to view the multicast group for sending and receiving heartbeat messages

cluster pre-shared-secret <secret>

Defines the shared key for heartbeat authentication.

Syntax

set cluster pre-shared-secret *secret*

delete cluster pre-shared-secret

show cluster pre-shared-secret

Command Mode

Configuration mode.

Configuration Statement

```
cluster {
   pre-shared-secret text
}
```

Parameters

secret	Mandatory. A shared key for heartbeat authentication.

Default

None.

Usage Guidelines

Use this command to specify the shared key for heartbeat authentication.

Use the **set** form of this command to specify the shared key for heartbeat authentication.

Use the **delete** form of this command to remove the shared key for heartbeat authentication.

Use the **show** form of this command to view the shared key for heartbeat authentication.

show cluster status

Displays current clustering status.

Syntax

show cluster status

Command Mode

Operational mode.

Parameters

None.

Usage Guidelines

Use this command to view the operational status of a cluster.

Examples

Example 3-6 and Example 3-7 show output for **show cluster status** on the primary node and secondary nodes, respectively, in the case where the primary node is operational and active, and owns the cluster resources.

Example 3-6 “show cluster status”: Primary node active (primary output)

```
vyatta@R1> show cluster status
=== Status report on primary node R1 ===

  Primary R1 (this node): Active

  Secondary R2: Active (standby)

  Monitor 10.6.0.55: Reachable
  Monitor 10.1.0.1: Reachable

  Resources [10.6.0.100 10.1.0.170 ipsec]:
    Active on primary R1 (this node)
```

Example 3-7 “show cluster status”: Primary node output (secondary output)

```
vyatta@R2> show cluster status
=== Status report on secondary node R2 ===

  Primary R1: Active
  Secondary R2 (this node): Active (standby)

  Monitor 10.6.0.55: Reachable
  Monitor 10.1.0.1: Reachable

  Resources [10.6.0.100 10.1.0.170 ipsec]:
    Active on primary R1
```

Example 3-8 and Example 3-9 show output for **show cluster status** on the primary node and secondary nodes, respectively, in the case where interface eth1 R1 has failed and is unable to reach the upstream monitor mode (10.6.0.55). Therefore, the failover mechanism has migrated the cluster resources to the secondary node, R2.

Example 3-8 “show cluster status”: Failed link on primary (primary output)

```
vyatta@R1> show cluster status
=== Status report on primary node R1 ===

  Primary R1 (this node): Down (at least 1 monitor not reachable)

  Secondary R2: Active

  Monitor 10.6.0.55: Unreachable
  Monitor 10.1.0.1: Reachable

  Resources [10.6.0.100 10.1.0.170 ipsec]:
    Active on secondary R2
```

Example 3-9 “show cluster status”: Failed link on primary (secondary output)

```
vyatta@R2> show cluster status
=== Status report on secondary node R2 ===

  Primary R1: Down (at least 1 monitor node not reachable)

  Secondary R2 (this node): Active

  Monitor 10.6.0.55: Reachable
  Monitor 10.1.0.1: Reachable

  Resources [10.6.0.100 10.1.0.170 ipsec]:
    Active on secondary R2 (this node)
```

Example 3-10 shows output for **show cluster status** on the secondary node R2 in the case where the primary node R1 has failed altogether and the failover mechanism has migrated the resources to R2.

Example 3-10 “show cluster status”: Failed primary node (secondary output)

```
vyatta@R2> show cluster status
=== Status report on secondary node R2 ===

  Primary R1: Down

  Secondary R2(this node): Active

  Monitor 10.6.0.55: Reachable
  Monitor 10.1.0.1: Reachable

  Resources [10.6.0.100 10.1.0.170 ipsec]:
    Active on secondary R2 (this node)
```

Chapter 4: RAID 1

This chapter describes how to set up hard drives in a Redundant Array of Independent Disks (RAID) 1 deployment using the Vyatta system.

This chapter presents the following topics:

- RAID 1 Configuration
- RAID 1 Commands

RAID 1 Configuration

This section describes how to set up RAID 1 on the Vyatta system. This section presents the following topics:

- RAID 1 Overview
- RAID 1 Operational Examples

RAID 1 Overview

This section presents the following topics:

- RAID Implementations
- RAID-1 Set States
- Installation Implications
- BIOS Issues

RAID Implementations

A Redundant Array of Independent Disks (RAID) uses two or more hard disk drives to improve disk speed, store more data, and/or provide fault tolerance. There are several storage schemes possible in a RAID array, each offering a different combination of storage, reliability, and/or performance.

The Vyatta system supports a “RAID 1” deployment. RAID 1 allows two or more disks to mirror one another to provide system fault tolerance. In a RAID 1 solution, every sector of one disk is duplicated onto every sector of all disks in the array. Provided even one disk in the RAID 1 set is operational, the system continues to run, even through disk replacement (provided that the hardware supports in-service replacement of drives).

RAID 1 can be implemented using special hardware or it can be implemented in software. The Vyatta system supports software RAID 1 on two disks.

The Vyatta implementation of RAID 1 allows the following:

- Detection and reporting of disk failure
- The ability to maintain system operation with one failed disk
- The ability to boot the system with one failed disk
- The ability to replace a failed disk and initiate re-mirroring
- The ability to monitor the status of remirroring

On a Vyatta system, RAID 1 is configured during the installation process. Likewise, breaking a RAID 1 set into two component (non–RAID 1) disks also requires reinstallation of the Vyatta software. If two disks of dissimilar size are used together in a RAID 1 set, the system sizes the partitions based on the size of the smaller disk, and there will be unused space on the larger disk.

All RAID-1 configuration information is located on the hard disk, not in the Vyatta configuration file. For this reason, there are no configuration mode commands associated with this feature.

RAID-1 Set States

A RAID 1 set has several "states" associated with it which reflect the health of the array. Some of these states are completely independent (that is, their value has no bearing on other states), while others interact. States are reported in the "State" line in the output of the show raid command. Table 4-1 shows the relevant state variables.

Table 4-1 RAID 1 state variables

State Variable	Description
Active	There is outstanding write I/O. If the system crashes while in the Active state, it is considered an unclean shutdown and the system enters a Resyncing state after the system reboots. Active and Clean states are mutually exclusive, and both are independent of the other states.
Clean	All write I/O has been completed. Active and Clean states are mutually exclusive, and both are independent of the other states.
Degraded	The RAID 1 set is missing one or more members. Since the Vyatta system supports only two-disk RAID 1, this means that the RAID 1 set is operating with only one member.
Recovering	A new member has been added to a RAID 1 set, and the system is in the process of copying the data from another member to the new member. The new member will not be usable until the rebuild is completed. The Recovering state can only occur if the RAID 1 set is in the Degraded state.
Resyncing	The system is recovering from an unclean shutdown by copying all of the data from one member to the others. The objective of recovery is simply to make the two members identical, not to recover I/O that was lost at the time of the unclean shutdown. Since, after an unclean shutdown, the system has no way of knowing which of the members is more accurate, it therefore arbitrarily chooses one to be the source of the resync. Since both members hold valid data, this state is not considered "unhealthy"; the data on both disks is valid. The Resyncing state never occurs at the same time as Degraded or Recovering.
Synchronized	The RAID 1 set is not Degraded, Recovering, or Resyncing.

The RAID 1 set is considered “Synchronized” if it is not Degraded, Recovering, or Resyncing—that is, both disks are present and healthy, and the only state value displayed is either Clean or Active. This is shown in Example 4-1.

Example 4-1 RAID 1 Synchronized state

```
State:   clean
Number   Major Minor RaidDevice  State
0        8     2     0           active sync  /dev/sda2
1        8     18    1           active sync  /dev/sdb2
```

In Example 4-2, one disk has been removed, and the RAID set is running on only one member. The disk display section of the command output clearly shows that only one member of the RAID 1 set is present.

Example 4-2 RAID 1 Degraded state

```
State:   clean, degraded
Number   Major Minor RaidDevice  State
0        0     0     0           removed      /dev/sda2
1        8     18    1           active sync  /dev/sdb2
```

In Example 4-3, a second disk has been added and is in the process of rebuilding. The disk display shows which member is being rebuilt. Note that the member is considered “spare” until the rebuild is completed.

Example 4-3 RAID 1 Recovering state

```
State:   clean, degraded, recovering
Rebuild status: 3% complete

Number   Major Minor RaidDevice  State
2        8     18    0           spare rebuilding/dev/sda2
1        8     18    1           active sync  /dev/sdb2
```

In Example 4-4, the RAID 1 set is recovering from an unclean shutdown. As in the Recovering state, the status of the rebuild is displayed; unlike the Recovering state, both drives are considered healthy.

Example 4-4 RAID 1 Resyncing state

```
State:   active, resyncing
Rebuild status: 3% complete

Number   Major Minor RaidDevice  State
2        8     2     0           active sync  /dev/sda2
1        8     18    1           active sync  /dev/sdb2
```

Booting

The Vyatta system uses the **grub-2** boot package. The **install-system** utility installs the a small first-stage boot program from the **grub** package into the Master Boot Record (MBR), which occupies the first sector of both disk drives. It also installs a small second-stage **grub** boot program onto both disks at a location between the MBR and the first partition. The Vyatta software will reinstall this boot code when a new member is added to a RAID 1 set.

Neither of these sections are covered by the RAID 1 set, but by installing the identical boot code onto both drives, the system can boot from either drive.

- The purpose of the first-stage boot program is to load the second-stage boot program.
- The purpose of the second-stage boot program is to load the kernel and initial RAMdisk files residing on the root file system, which is located on the RAID 1 set.

The first-stage boot program is unaware of the RAID subsystem; it can only operate correctly if it can locate the second-stage boot program on the same disk drive. The second-stage boot program, on the other hand, is aware of the RAID subsystem; it can operate correctly provide one of the two disk partitions comprising the RAID 1 set is available.

When a new member is added to a RAID 1 group, the new member must be “rebuilt”: the contents of the good member is copied to the new member. The **grub** boot sections can only be installed *after* the rebuild is complete. When you issue the **add raid <RAID-1-device> member <disk-partition>** command to add a new member, the system starts the rebuild.

After rebuilding is complete, the system automatically writes the two **grub** sections on the new disk drive. This means you must wait for the rebuild to complete before rebooting the system; otherwise the new disk will not be bootable.

The system will write the boot sections only when the root file system is located on the RAID 1 group; It will not do so when the system is running on LiveCD.

Installation Implications

The Vyatta systems installation utility provides several options for installing to a RAID 1 set. You can:

- Use the **install-system** to create the RAID 1 set
- Use the underlying Linux commands to create a RAID 1 set before running the **install-system** command
- Use a previously-created RAID 1 set.

However the RAID 1 set is created, you must be aware of the states of the RAID 1 set, and observe the following rules:

It is SAFE to install when:

- The RAID 1 set is in Synchronized state. This is the normal case.
- A RAID 1 set is in Resyncing state. Sometimes, the RAID 1 set will go into Resyncing state when **install-system** creates it. It is also safe to reboot after running **install-system** if the system is in Resyncing state as the system will restart the resyncing after rebooting.
- The RAID 1 set state Degraded BUT NOT Recovering. However, in this case, be aware that the RAID 1 set is missing a member.

It is NOT SAFE to install when:

- The RAID 1 set state en a RAID 1 set is Degraded AND Recovering. This is because the system is in the process of adding a new member to the RAID 1 set, and the grub boot program will not be set up properly on the new member. Instead, the user should wait for the rebuild to complete before starting **install-system**.

It is NOT SAFE to add a new member to the RAID 1 set:

- AFTER running the **install-system** utility BUT BEFORE rebooting. This is because the **grub** boot program will not be set up properly on the new drive. Instead, you should reboot the system, let the system come up on the RAID 1 set, and only then add the new member. Once the system is running on the RAID 1 set, it will ensure that **grub** is properly set up whenever a new drive is added.

BIOS Issues

The first stage of booting takes place when the BIOS reads the master boot record from one of the disks and executes the small boot program it contains. This process is completely outside of the control of the software RAID feature, and different platforms behave differently.

The software RAID feature will set up both of the disks that are members of the RAID 1 set to be bootable. Most BIOS implementations provide control over boot order, allowing the user to select one or the other disk to be tried first. Some, but not all, BIOS implementations automatically fail over to the second disk if the first disk in the boot order is missing or failing in some way.

When a replacement disk drive is added, you may need to navigate the BIOS configuration menu in order to boot the system from the remaining good disk instead of the new disk drive. This procedure is necessarily platform-dependent.

RAID 1 Operational Examples

This section presents the following topics:

- Setting Up a Non–RAID 1 System
- Non–RAID 1 to RAID 1
- RAID 1 to Non–RAID 1
- RAID 1 to RAID 1
- RAID 1 to new RAID 1
- Detecting and Replacing a Failed RAID 1 Disk

Setting Up a Non–RAID 1 System

When the Vyatta system is installed, it automatically detects the presence of two disks not currently part of a RAID array. In these cases, the Vyatta installation utility automatically offers you the option of configuring RAID 1 mirroring for the drives, with the following prompt.

```
Would you like to configure RAID 1 mirroring on them?
```

- If you do not want to configure RAID 1 mirroring, enter “No” at the prompt and continue with installation in the normal way.

Non–RAID 1 to RAID 1

If you reinstall a non-RAID Vyatta system on a system with two identical disks that are not currently part of a RAID 1 set, the Vyatta installation utility automatically offers you the option of configuring RAID 1 mirroring for the drives, with the following prompt.

```
Would you like to configure RAID 1 mirroring on them?
```

1. To create a new RAID 1 array, enter “Yes” at the prompt. If the system detects a filesystem on the partitions being used for RAID 1 it will prompt you to indicate whether you want to continue creating the RAID 1 array.

   ```
   Continue creating array?
   ```

2. To overwrite the old filesystem, enter “Yes”.
3. The system informs you that all data on both drives will be erased. You are prompted to confirm that you want to continue

   ```
   Are you sure you want to do this?
   ```

4. Enter “Yes” at the prompt. The system prompts you to indicate whether you want to save the old configuration data. This represents the current Vyatta configuration.

```
Would you like me to save the data on it before I delete it?
```

5 Enter “Yes” at the prompt to retain the current Vyatta configuration once installation is complete. Enter “No” to delete the current Vyatta configuration.

6 Continue with installation in the normal way.

RAID 1 to Non–RAID 1

If you reinstall Vyatta software on a system with a RAID 1 set already configured, the installation utility will detect the array and will display the following prompt:

```
Would you like to use this one?
```

1 To break apart the current RAID 1 set, enter “No” at the prompt. The installation utility detects that there are two identical disks and offers you the option of configuring RAID 1 mirroring on them, displaying the following prompt:

```
Would you like to configure RAID 1 mirroring on them?
```

2 To decline to set up a new RAID 1 configuration on the disks, enter “No” at the prompt. The system prompts you to indicate which partition you would like the system installed on.

```
Which partition should I install the root on? [sda1]:
```

3 Enter the partition where you would like the system installed. The system then prompts you to indicate whether you want to save the old configuration data. This represents the current Vyatta configuration.

```
Would you like me to save the data on it before I delete it?
```

4 Enter “Yes” at the prompt to retain the current Vyatta configuration once installation is complete. Enter “No” to delete the current Vyatta configuration.

5 Continue with installation in the normal way.

RAID 1 to RAID 1

If you reinstall the Vyatta software on a system with a RAID 1 set already configured, the installation utility will detect the array and will display the following prompt:

```
Would you like to use this one?
```

1 To continue to use the existing RAID 1 array, enter “Yes” at the prompt. The system prompts you to indicate whether you want to save the old configuration data. This represents the current Vyatta configuration.

```
Would you like me to save the data on it before I delete it?
```

2 Enter “Yes” at the prompt to retain the current Vyatta configuration once installation is complete. Enter “No” to delete all current Vyatta configuration.

3 Continue with installation in the normal way.

RAID 1 to new RAID 1

You can also recreate the RAID 1 array on disk drives already configured for RAID-1.The installation utility will detect the array and will display the following prompt:

```
Would you like to use this one?
```

1 To stop using the existing RAID 1 array, enter “No” at the prompt. The system detects the two disks and prompts you to indicate whether you want to configure RAID 1 mirroring in them.

```
Would you like to configure RAID 1 mirroring on them?
```

2 To create a new RAID 1 array, enter “Yes” at the prompt. If the system detects a file system on the partitions being used for RAID 1 it will prompt you to indicate whether you want to continue creating the RAID 1 array.

```
Continue creating array?
```

3 To overwrite the old filesystem, enter “Yes”.

4 Continue with installation in the normal way.

Detecting and Replacing a Failed RAID 1 Disk

The Vyatta system automatically detects a disk failure within a RAID 1 set and reports it to the system console. You can verify the failure by issuing the **show raid** command.

To replace a bad disk within a RAID 1 set, perform the following steps:

1 Remove the failed disk from the RAID 1 set by issuing the following command:

```
remove raid RAID-1-device member disk-partition
```

where *RAID-1-device* is the name of the RAID 1 device (for example, **md0**) and *disk-partition* is the name of the failed disk partition (for example, s**db2**).

2 Physically remove the failed disk from the system. If the drives are not hot-swappable, then you must shut down the system before removing the disk.

3 Replace the failed drive with a drive of the same size or larger.

4 Format the new disk for RAID 1 by issuing the following command:

```
format disk-device1 like disk-device2
```

where *disk-device1* is the replacement disk (for example, **sdb**) and *disk-device2* is the existing healthy disk (for example, **sda**).

5 Add the replacement disk to the RAID 1 set by issuing the following command:

```
add RAID-1-device member disk-partition
```

where *RAID-1-device* is the name of the RAID 1 device (for example, **md0**) and *disk-partition* is the name of the replacement disk partition (for example, **sdb2**).

RAID 1 Commands

This section presents the following commands.

Configuration Commands	
None	

Operational Commands	
add raid <RAID-1-device> member <disk-partition>	Adds a disk partition to the specified RAID 1 set.
format <disk-device1> like <disk-device2>	Formats the first disk device to be just like the second.
remove raid <RAID-1-device> member <disk-partition>	Removes a member of the specified RAID 1 device.
show disk <disk-device> format	Displays the formatting of the specified disk.
show raid <RAID-1-device>	Displays the status of the specified RAID 1 device.

add raid <RAID-1-device> member <disk-partition>

Adds a disk partition to the specified RAID 1 set.

Syntax

add raid *RAID-1-device* **member** *disk-partition*

Command Mode

Operational mode.

Parameters

RAID-1-device	The name of the RAID 1 device. This name will have a form similar to **md0**; it represents the device name for the RAID 1 set of the same name residing in **/dev/**.
disk-partition	The disk partition to be made a RAID 1 member. The device name will have a form similar to **sda1**; it represents the block device of the same name residing in **/dev/**.

Default

None.

Usage Guidelines

Use this command to add a member disk partition to the RAID 1 set. Adding a disk partition to a RAID 1 set initiates mirror synchronization, where all data on the existing member partition is copied to the new partition.

Before adding a brand new drive to a RAID 1 set, the drive must be formatted using the **format <disk-device1> like <disk-device2>** command (see page 192).

format <disk-device1> like <disk-device2>

Formats the first disk device to be just like the second.

Syntax

format *disk-device1* **like** *disk-device2*

Command Mode

Operational mode.

Parameters

disk-device1	The disk to format. The device name will have a form similar to **sda**; it represents the block device of the same name residing in **/dev/**.
disk-device2	The disk whose partitioning you wish to emulate. The device name will have a form similar to **sdb**; it represents the block device of the same name residing in **/dev/**.

Default

None.

Usage Guidelines

Use this command to format a disk to be partitioned exactly like a second disk.

The disk to be formatted must be inactive; that is, it must not have any partitions mounted and it must not already be part of an active RAID 1 set. In formatting, no data is copied to the formatted device, but any existing data on the formatted device is lost.

This command is typically used to prepare a disk to be added to a preexisting RAID 1 set (of which *disk-device2* is already a member). To add the disk to the RAID 1 set. use the **add raid <RAID-1-device> member <disk-partition>** command (see page 191).

remove raid <RAID-1-device> member <disk-partition>

Removes a member of the specified RAID 1 device.

Syntax

remove raid *RAID-1-device* **member** *disk-partition*

Command Mode

Operational mode.

Parameters

RAID 1_device	The name of the RAID 1 device. This name will have a form similar to **md0**; it represents the device name for the RAID 1 set of the same name residing in **/dev/**.
disk_partition	The RAID 1 member disk partition. The device name will have a form similar to **sda1**; it represents the block device of the same name residing in **/dev/**.

Default

None.

Usage Guidelines

Use this command to remove a member disk partition from a RAID 1 set.

The command will not allow the last member disk to be removed from the RAID 1 set. To remove the last disk from the set, you must reinstall the Vyatta software and decline the offer to continue using the RAID 1 set. For this procedure, see “RAID 1 to Non–RAID 1” on page 187.

show disk <disk-device> format

Displays the formatting of the specified disk.

Syntax

show disk *disk-device* **format**

Command Mode

Operational mode.

Parameters

disk-device	The disk device name. The device name will have a form similar to **sda**; it represents the block device of the same name residing in **/dev/**.

Default

None.

Usage Guidelines

Use this command to display the formatting of a hard disk.

The information shown includes the partitions on the disk, their size, the start and end sectors, and the system ID.

Examples

Example 4-5 shows output for **show disk** *sda* **format**.

Example 4-5 “show disk sda format”: Displaying information about a member of a RAID 1 set.

```
vyatta@vyatta:~$ show disk sda format

Disk /dev/sda: 1073 MB, 1073741824 bytes
85 heads, 9 sectors/track, 2741 cylinders
Units = cylinders of 765 * 512 = 391680 bytes
Disk identifier: 0x000b7179

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               6        2737     1044922+  fd  Linux raid autodetect
vyatta@vyatta:~$
```

show raid <RAID-1-device>

Displays the status of the specified RAID 1 device.

Syntax

show raid *RAID 1_device*

Command Mode

Operational mode.

Parameters

RAID-1-device	The name of the RAID 1 device. This name will have a form similar to **md0**; it represents the device name for the RAID 1 set of the same name residing in **/dev/**.

Default

None.

Usage Guidelines

Use this command to display the status of a RAID 1 device.

A RAID 1 device is created during system installation. It consists of two identical partitions on two physical disks which mirror one another to provide fault tolerance. These are the members of the RAID 1 set.

The information shown includes the devices that are members of the RAID 1 set, whether any of the members are offline, whether the RAID 1 set is currently undergoing mirror resynchronization, and, if so, the percentage of synchronization that is complete.

Examples

Example 4-6 shows output for **show raid** *md0* as sdb1 is being added to the RAID 1 set and is in the process of being resynchronized.

Example 4-6 “show raid md0”: Displaying information about a RAID 1 set with two members - one being resynchronized.

```
vyatta@vyatta:~$ show raid md0
/dev/md0:
        Version : 00.90
  Creation Time : Wed Oct 29 09:19:09 2008
     Raid Level : raid1
     Array Size : 1044800 (1020.48 MiB 1069.88 MB)
  Used Dev Size : 1044800 (1020.48 MiB 1069.88 MB)
```

```
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Wed Oct 29 19:34:23 2008
          State : active, degraded, recovering
 Active Devices : 1
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 1

 Rebuild Status : 17% complete

           UUID : 981abd77:9f8c8dd8:fdbf4de4:3436c70f
         Events : 0.103

    Number   Major   Minor   RaidDevice State
       0       8        1        0      active sync   /dev/sda1
      2      8      17       1     spare rebuilding   /dev/sdb1
vyatta@vyatta:~$
```

Example 4-7shows output for **show raid** *md0*.

Example 4-7 “show raid md0”: Displaying information about a RAID 1 set with two synchronized members.

```
vyatta@vyatta:~$ show raid md0
/dev/md0:
        Version : 00.90
  Creation Time : Wed Oct 29 09:19:09 2008
     Raid Level : raid1
     Array Size : 1044800 (1020.48 MiB 1069.88 MB)
  Used Dev Size : 1044800 (1020.48 MiB 1069.88 MB)
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Wed Oct 29 18:05:26 2008
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

           UUID : 981abd77:9f8c8dd8:fdbf4de4:3436c70f
         Events : 0.6
    Number   Major   Minor   RaidDevice State
       0       8        1        0      active sync   /dev/sda1
       1       8       17        1      active sync   /dev/sdb1
vyatta@vyatta:~$
```

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System

DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol

MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RA	router advertisement
RIB	Routing Information Base
RIP	Routing Information Protocol

RIPng	RIP next generation
RS	router solicitation
Rx	receive
SLAAC	Stateless address auto-configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer 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