VYATTA, INC. | Vyatta System

Basic Routing

REFERENCE GUIDE

Forwarding and Routing
Static Routes

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

COPYRIGHT

PROPRIETARY NOTICES

RELEASE DATE: March 2012

DOCUMENT REVISION. R6.4 v01

RELEASED WITH: R6.4.0

PART NO. A0-0215-10-0014

Contents

Quick List of Commands

Use this list to help you quickly locate commands.

List of Examples

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

Preface

This document presents information about forwarding and routing on the Vyatta system.

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

Intended Audience

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

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

Organization of This Guide

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

- Quick List of Commands

  Use this list to help you quickly locate commands.

- List of Examples

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

This guide has the following chapters:

Chapter	Description	Page
Chapter 1: Forwarding and Routing	This chapter describes commands for forwarding and basic routing.	1
Chapter 2: Static Routes	This chapter explains how to set static routes using the Vyatta system.	36
Glossary of Acronyms		57

Document Conventions

This guide uses the following advisory paragraphs, as follows.

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

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

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

This document uses the following typographic conventions.

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[key1 | key2]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[*arg*...] *arg*[,*arg*...]	A value that can optionally represent a list of elements (a space-separated list and a comma-separated list, respectively).

Vyatta Publications

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

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

Chapter 1: Forwarding and Routing

This chapter describes commands for forwarding and basic routing.

This chapter presents the following topics:

Forwarding and Routing Commands

This chapter contains the following commands.

Configuration Commands	
None	
Operational Commands	
clear ip prefix-list	Clears prefix list statistics or status.
clear ipv6 prefix-list	Clears prefix list statistics or status.
`ping <host>`	Sends ICMP ECHO_REQUEST packets to network hosts.
reset ip route cache	Flushes the kernel route cache.
reset ipv6 route cache	Flushes the kernel IPv6 route cache.
show ip forwarding	Displays IP forwarding status.
show ip route	Displays routes stored in the RIB and FIB.
show ip route <ipv4net> longer-prefixes	Displays prefixes longer than a specified prefix.
show ip route cache	Displays the kernel route cache.
show ip route connected	Displays directly connected routes.
show ip route forward	Displays routes stored in the FIB.
show ip route static	Displays static routes.
show ip route kernel	Displays kernel routes.
show ip route summary	Displays routes summary.
show ip route supernets-only	Displays supernet routes.
show ipv6 route	Displays IPv6 routes stored in the RIB and FIB.
show ipv6 route <ipv6net> longer-prefixes	Displays IPv6 prefixes longer than a specified prefix.
show ipv6 route bgp	Displays IPv6 BGP routes.
show ipv6 route cache	Displays the kernel IPv6 route cache.
show ipv6 route connected	Displays IPv6 connected routes.
show ipv6 route forward	Displays IPv6 routes stored in the FIB.
show ipv6 route kernel	Displays IPv6 kernel routes.

show ipv6 route ripng	Displays IPv6 RIPng routes.
show ipv6 route static	Displays IPv6 static routes.
show ipv6 route summary	Displays IPv6 routes summary.
show monitoring protocols rib	Displays Routing Information Base (RIB) debugging flags.
show table	Displays the system's routing table.
`traceroute <host>`	Displays the route packets take to a network host.

clear ip prefix-list

Clears prefix list statistics or status.

Syntax

clear ip prefix-list [*list-name* [*ipv4net*]]

Command Mode

Operational mode.

Parameters

list-name	Optional. Clears statistics for the specified prefix list.
ipv4net	Optional. Clears statistics for the specified network.

Default

Statistics for all prefix-lists are cleared.

Usage Guidelines

Use this command to clear prefix list statistics or status.

clear ipv6 prefix-list

Clears prefix list statistics or status.

Syntax

clear ipv6 prefix-list [*list-name* [*ipv6net*]]

Command Mode

Operational mode.

Parameters

list-name	Optional. Clears statistics for the specified prefix list.
ipv6net	Optional. Clears statistics for the specified network.

Default

Statistics for all prefix-lists are cleared.

Usage Guidelines

Use this command to clear prefix list statistics or status.

ping <host>

Sends ICMP ECHO_REQUEST packets to network hosts.

Syntax

ping [**ipv4** | **ipv6**] *host*

Command Mode

Operational mode

Parameters

ipv4	Specifies that the *host's* IPv4 address is to be pinged. This option is to be used when the *host* specified is a hostname rather than an IP address.
ipv6	Specifies that the *host's* IPv6 address is to be pinged. This option is to be used when the *host* specified is a hostname rather than an IP address.
host	The host being pinged. Can be specified either as hostname (if DNS is being used on the network) or as an IPv4 or IPv6 address. If a hostname is specified and neither the **ipv4** or **ipv6** options are used, the IPv4 or the IPv6 address associated with the hostname will be pinged depending on which of these is resolved first.

Usage Guidelines

The **ping** command is used to test whether a network host is reachable or not.

The **ping** command uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (pings) have an IP and ICMP header, followed by a "`struct timeval`" and then an arbitrary number of pad bytes used to fill out the packet.

When using **ping** for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be "pinged." Round-trip times and packet loss statistics are computed.

If duplicate packets are received, they are not included in the packet loss calculation, although the round-trip time of these packets is used in calculating the minimum/average/maximum round-trip time numbers.

When the **ping** command is interrupted, using **<Ctrl>+c**, a brief statistical summary is displayed.

reset ip route cache

Flushes the kernel route cache.

Syntax

reset ip route cache [*ipv4net*]

Command Mode

Operational mode.

Parameters

ipv4net	Optional. Flushes the specified route from the kernel route cache.

Default

Flushes the entire route cache.

Usage Guidelines

Use this command to flush the kernel route cache or a flush a specific route from the cache.

reset ipv6 route cache

Flushes the kernel IPv6 route cache.

Syntax

reset ipv6 route cache [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Flushes the specified route from the kernel IPv6 route cache.

Default

Flushes the entire IPv6 route cache.

Usage Guidelines

Use this command to flush the kernel IPv6 route cache or a flush a specific route from the cache.

show ip forwarding

Displays IP forwarding status.

Syntax

show ip forwarding

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display the current IP forwarding status.

Examples

Example 1-1 shows how to display the status of IP forwarding.

Example 1-1 Displaying IP forwarding status

```
vyatta@vyatta:~$ show ip forwarding
IP forwarding is on
vyatta@vyatta:~$
```

show ip route

Displays routes stored in the RIB and FIB.

Syntax

show ip route [*ipv4* | *ipv4net*]

Command Mode

Operational mode.

Parameters

ipv4	Optional. Displays routing information for the specified address.
ipv4net	Optional. Displays routing information for the specified prefix.

Default

Lists all routes in the RIB and FIB.

Usage Guidelines

Use this command to display active prefixes stored in the Routing Information Base (RIB), as well as those stored in the Forwarding Information Base (FIB).

The routes shown in the FIB can also be seen using show ip route forward command.

Examples

Example 1-2 shows how to display routes in the RIB and FIB

Example 1-2 Displaying routes in the RIB and FIB

```
vyatta@vyatta:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.1.0.1, eth0
O   10.1.0.0/24 [110/10] is directly connected, eth0, 05:35:15
C>* 10.1.0.0/24 is directly connected, eth0
```

```
O>* 10.192.32.0/24 [110/20] via 10.1.0.45, eth0, 05:35:15
O>* 10.192.128.0/24 [110/11] via 10.1.0.66, eth0, 05:35:15
O>* 10.192.128.1/32 [110/11] via 10.1.0.66, eth0, 05:35:15
O>* 10.192.129.0/24 [110/11] via 10.1.0.66, eth0, 05:35:15
O>* 10.192.130.0/24 [110/11] via 10.1.0.66, eth0, 05:35:15
O>* 10.192.131.0/24 [110/11] via 10.1.0.66, eth0, 05:35:15
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.0.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.1.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.2.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.3.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.4.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.5.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.6.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.7.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.8.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
O>* 172.16.9.0/24 [110/11] via 10.1.0.4, eth0, 05:35:15
C>* 172.16.234.0/25 is directly connected, eth1
S>* 192.94.202.0/24 [1/0] via 172.16.234.27, eth1
vyatta@vyatta:~$
```

Example 1-3 shows information how to display information for the route to address 10.192.128.1.

Example 1-3 Displaying routing information about a specific address

```
vyatta@vyatta:~$ show ip route 10.192.128.1
Routing entry for 10.192.128.1/32
  Known via "ospf", distance 110, metric 11, best
  Last update 09:47:07 ago
  * 10.1.0.66, via eth0
vyatta@vyatta:~$
```

show ip route <ipv4net> longer-prefixes

Displays prefixes longer than a specified prefix.

Syntax

show ip route *ipv4net* **longer-prefixes**

Command Mode

Operational mode.

Parameters

ipv4net	Mandatory. Displays all prefixes longer than the specified prefix.

Default

None.

Usage Guidelines

Use this command to display all prefixes in the Routing Information Base (RIB) that are longer than a given IP address or prefix.

Examples

Example 1-4 shows how to list prefixes longer than the prefix 10.192.128.0/24.

Example 1-4 Displaying routes with longer prefixes

```
vyatta@vyatta:~$ show ip route 10.192.128.0/24 longer-prefixes
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O>* 10.192.128.0/24 [110/11] via 10.1.0.66, eth0, 09:36:20
O>* 10.192.128.1/32 [110/11] via 10.1.0.66, eth0, 09:36:20
vyatta@vyatta:~$
```

show ip route cache

Displays the kernel route cache.

Syntax

show ip route cache [*ipv4net*]

Command Mode

Operational mode.

Parameters

ipv4net	Optional. Displays kernel route cache information for the specified route.

Default

Lists routes in the kernel route cache.

Usage Guidelines

Use this command to display information about routes stored in the kernel route cache. The route cache contains all paths currently in use by the cache. Multiple equal-cost paths are necessary before equal-cost-multi-path (ECMP) routing can be performed.

Examples

Example 1-5 shows how to list routes in the kernel route cache.

Example 1-5 Listing routes in the kernel route cache

```
vyatta@vyatta:~$ show ip route cache
local 10.1.0.62 from 10.1.0.1 dev lo  src 10.1.0.62
    cache <local,src-direct>  users 1 age 42sec iif eth0
multicast 224.0.0.5 from 10.1.0.45 dev lo  src 10.1.0.62
    cache <local,mc>  users 1 used 8 age 5sec iif eth0
local 10.1.0.62 from 69.59.150.131 dev lo  src 10.1.0.62
    cache <local>  users 1 used 3 age 47sec iif eth0
10.1.0.1 from 10.1.0.62 dev eth0
```

```
    cache  users 1 age 42sec mtu 1500 advmss 1460 hoplimit 64
10.0.0.30 from 10.1.0.62 tos lowdelay via 10.1.0.1 dev eth0
    cache  users 2 age 0sec mtu 1500 advmss 1460 hoplimit 64
multicast 224.0.0.5 from 10.1.0.56 dev lo  src 10.1.0.62
    cache <local,mc>  users 1 used 8 age 8sec iif eth0
multicast 224.0.0.5 from 10.1.0.66 dev lo  src 10.1.0.62
    cache <local,mc>  users 1 used 8 age 0sec iif eth0
multicast 224.0.0.6 dev eth0  src 10.1.0.62
    cache <mc>  users 1 age 21sec mtu 1500 advmss 1460 hoplimit 64
multicast 224.0.0.5 from 10.1.0.4 dev lo  src 10.1.0.62
    cache <local,mc>  users 1 used 9 age 1sec iif eth0
69.59.150.131 via 10.1.0.1 dev eth0  src 10.1.0.62
    cache  users 1 age 47sec mtu 1500 advmss 1460 hoplimit 64
multicast 224.0.0.5 dev eth0  src 10.1.0.62
   cache <local,mc>  users 1 used 8 age 5sec mtu 1500 advmss 1460 hoplimit
64
69.59.150.131 from 10.1.0.62 via 10.1.0.1 dev eth0
    cache  users 1 used 1 age 47sec mtu 1500 advmss 1460 hoplimit 64
local 10.1.0.62 from 10.0.0.30 tos lowdelay dev lo  src 10.1.0.62
    cache <local>  users 1 used 1 age 0sec iif eth0
vyatta@vyatta:~$
```

Example 1-6 shows how to display information about route 10.1.0.62 in the kernel route cache.

Example 1-6 Displaying information about a route in the kernel route cache

```
vyatta@vyatta:~$ show ip route cache 10.1.0.62
local 10.1.0.62 from 10.1.0.1 dev lo  src 10.1.0.62
    cache <local,src-direct>  users 1 used 3 age 9sec iif eth0
local 10.1.0.62 from 69.59.150.131 dev lo  src 10.1.0.62
    cache <local>  users 1 used 7 age 102sec iif eth0
local 10.1.0.62 from 10.0.0.30 tos lowdelay dev lo  src 10.1.0.62
    cache <local>  users 1 used 33 iif eth0
vyatta@vyatta:~$
```

show ip route connected

Displays directly connected routes.

Syntax

show ip route connected

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display routes directly connected to the local system.

Examples

Example 1-7 shows how to list directly connected routes.

Example 1-7 Displaying connected routes

```
vyatta@vyatta:~$ show ip route connected
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.1.0.0/24 is directly connected, eth0
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.16.234.0/25 is directly connected, eth1
vyatta@vyatta:~$
```

show ip route forward

Displays routes stored in the FIB.

Syntax

show ip route forward [*ipv4net*]

Command Mode

Operational mode.

Parameters

ipv4net	Optional. Displays information from the kernel forwarding table for the specified route.

Default

Lists routes in the FIB.

Usage Guidelines

Use this command to display the FIB.

The FIB contains multiple equal-cost paths if existed. Multiple equal-cost paths are necessary before equal-cost multi-path (ECMP) routing or WAN load balancing can be performed.

Examples

Example 1-8 shows how to display routes recorded in the FIB.

Example 1-8 Displaying routes in the FIB

```
vyatta@vyatta:~$ show ip route forward
default via 10.1.0.1 dev eth0  proto zebra
10.1.0.0/24 dev eth0  proto kernel  scope link  src 10.1.0.62
10.192.32.0/24 via 10.1.0.45 dev eth0  proto zebra  metric 20
10.192.128.0/24 via 10.1.0.66 dev eth0  proto zebra  metric 11
10.192.128.1 via 10.1.0.66 dev eth0  proto zebra  metric 11
10.192.129.0/24 via 10.1.0.66 dev eth0  proto zebra  metric 11
10.192.130.0/24 via 10.1.0.66 dev eth0  proto zebra  metric 11
```

```
10.192.131.0/24 via 10.1.0.66 dev eth0  proto zebra  metric 11
172.16.0.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.1.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.2.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.3.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.4.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.5.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.6.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.7.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.8.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.9.0/24 via 10.1.0.4 dev eth0  proto zebra  metric 11
172.16.234.0/25 dev eth1  proto kernel  scope link  src 172.16.234.23
192.94.202.0/24 via 172.16.234.27 dev eth1  proto zebra
vyatta@vyatta:~$
```

Example 1-9 shows how to display information from the FIB about route 10.1.0.0/24.

Example 1-9 Displaying information about a route in the FIB

```
vyatta@vyatta:~$ show ip route forward 10.1.0.0/24
10.1.0.0/24 dev eth0  proto kernel  scope link  src 10.1.0.62
vyatta@vyatta:~$
```

show ip route kernel

Displays kernel routes.

Syntax

show ip route kernel

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display kernel routes. Kernel routes are routes that have been added through means other than by using the Vyatta CLI; for example by using the operating system route command, as in the following:

```
route add -net 10.172.24.0 netmask 255.255.255.0 gw 10.1.0.1
```

Examples

Example 1-10 shows how to display kernel routes.

Example 1-10 Displaying kernel routes

```
vyatta@vyatta:~$ show ip route kernel
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

K>* 10.172.24.0/24 via 10.1.0.1, eth0
vyatta@vyatta:~$
```

show ip route static

Displays static routes.

Syntax

show ip route static

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display static routes in the Routing Information Base (RIB).

Examples

Example 1-11 shows how to list static routes.

Example 1-11 Displaying static routes

```
vyatta@vyatta:~$ show ip route static
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.1.0.1, eth0
S>* 192.94.202.0/24 [1/0] via 172.16.234.27, eth1
vyatta@vyatta:~$
```

show ip route summary

Displays routes summary.

Syntax

show ip route summary

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display a summary of the various routes by route source.

Examples

Example 1-12 shows how to display a summary of routes.

Example 1-12 Displaying a summary of routes

```
vyatta@vyatta:~$ show ip route summary
Route Source         Routes               FIB
connected            4                    4
static               2                    2
ospf                 1                    0
ebgp                 0                    0
ibgp                 289016               289011
------
Totals               289023               289017
vyatta@vyatta:~$
```

show ip route supernets-only

Displays supernet routes.

Syntax

show ip route supernets-only

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display supernet routes.

Supernet routes are routes that have a subnet mask thatis less specific than the natural classful mask.

Examples

Example 1-13 shows how to list supernet routes.

Example 1-13 Displaying supernet routes

```
vyatta@vyatta:~$ show ip route supernets-only
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 0.0.0.0/0 [1/0] via 10.1.0.1, eth0
vyatta@vyatta:~$
```

show ipv6 route

Displays IPv6 routes stored in the RIB and FIB.

Syntax

show ipv6 route *[ipv6 | ipv6net]*

Command Mode

Operational mode.

Parameters

ipv6	Optional. Displays routing information for the specified IPv6 address.
ipv6net	Optional. Displays routing information for the specified IPv6 prefix.

Default

Lists all IPv6 routes in the RIB and FIB.

Usage Guidelines

Use this command to display active IPv6 prefixes stored in the Routing Information Base (RIB), as well as those stored in the Forwarding Information Base (FIB).

The routes shown in the FIB can also be seen using show ip route forward command.

show ipv6 route <ipv6net> longer-prefixes

Displays IPv6 prefixes longer than a specified prefix.

Syntax

show ipv6 route *ipv6net* **longer-prefixes**

Command Mode

Operational mode.

Parameters

ipv6net	Mandatory. Displays all prefixes longer than the specified IPv6 prefix.

Default

None.

Usage Guidelines

Use this command to display all prefixes in the Routing Information Base (RIB) that are longer than a given IPv6 address or prefix.

show ipv6 route bgp

Displays IPv6 BGP routes.

Syntax

show ipv6 route bgp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 BGP routes.

show ipv6 route cache

Displays the kernel IPv6 route cache.

Syntax

show ipv6 route cache [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Displays kernel IPv6 route cache information for the specified route.

Default

Lists routes in the kernel IPv6 route cache.

Usage Guidelines

Use this command to display information about routes stored in the kernel IPv6 route cache. The route cache contains all paths currently in use by the cache. Multiple equal-cost paths are necessary before equal-cost-multi-path (ECMP) routing can be performed.

show ipv6 route connected

Displays IPv6 connected routes.

Syntax

show ipv6 route connected

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 routes directly connected to the local system.

show ipv6 route forward

Displays IPv6 routes stored in the FIB.

Syntax

show ipv6 route forward [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Displays information from the kernel forwarding table for the specified IPv6 route.

Default

Lists IPv6 routes in the FIB.

Usage Guidelines

Use this command to display the FIB.

The FIB contains multiple equal-cost paths if existed. Multiple equal-cost paths are necessary before equal-cost multi-path (ECMP) routing or WAN load balancing can be performed.

show ipv6 route kernel

Displays IPv6 kernel routes.

Syntax

show ipv6 route kernel

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 kernel routes. Kernel routes are routes that have been added through means other than by using the Vyatta CLI.

show ipv6 route ripng

Displays IPv6 RIPng routes.

Syntax

show ipv6 route ripng

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 RIPng routes.

show ipv6 route static

Displays IPv6 static routes.

Syntax

show ipv6 route static

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 static routes in the Routing Information Base (RIB).

show ipv6 route summary

Displays IPv6 routes summary.

Syntax

show ipv6 route summary

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display a summary of the various IPv6 routes by route source.

Examples

Example 1-14 shows a summary of IPv6 routes.

Example 1-14 "show ipv6 route summary": Displaying a summary of IPv6 routes

```
vyatta@vyatta:~$ show ipv6 route summary
Route Source         Routes               FIB
connected            4                    4
static               2                    2
------
Totals               6                    6
vyatta@vyatta:~$
```

show monitoring protocols rib

Displays Routing Information Base (RIB) debugging flags.

Syntax

show monitoring protocols rib

Command Mode

Operational mode.

Parameters

None

Default

None.

Usage Guidelines

Use this command to see how debugging is set for the Routing Information Base.

show table

Displays the system's routing table.

Syntax

show table

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display the system's routing table.

Examples

Example 1-15 shows how to display the routing table.

Example 1-15 Displaying the routing table

```
vyatta@vyatta:~$ show table
table 0
vyatta@vyatta:~$
```

traceroute <host>

Displays the route packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host*

Command Mode

Operational mode

Parameters

ipv4	Display the route packets take to the *host's* IPv4 address. This option is to be used when the *host* specified is a hostname rather than an IP address.
ipv6	Display the route packets take to the *host's* IPv6 address. This option is to be used when the *host* specified is a hostname rather than an IP address.
host	The host that is the destination for the trace. Can be specified either as a name (if DNS is being used on the network) or as an IPv4 or IPv6 address.

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. Traceroute utilizes the IP protocol time to live ("ttl") field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host to track the route a set of packets follows. It attempts to trace the route an IP packet would follow to some Internet host by launching UDP probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Chapter 2: Static Routes

This chapter explains how to set static routes using the Vyatta system.

This chapter presents the following topics:

Static Route Configuration

This section presents the following topics:

Static Routes Overview

A static route is a manually configured route, which, in general, cannot be updated dynamically from information the Vyatta system learns about the network topology. However, if a link fails, the router will remove routes, including static routes, from the Routing Information Base (RIB) that used this interface to reach the next hop.

In general, static routes should only be used for very simple network topologies, or to override the behavior of a dynamic routing protocol for a small number of routes.

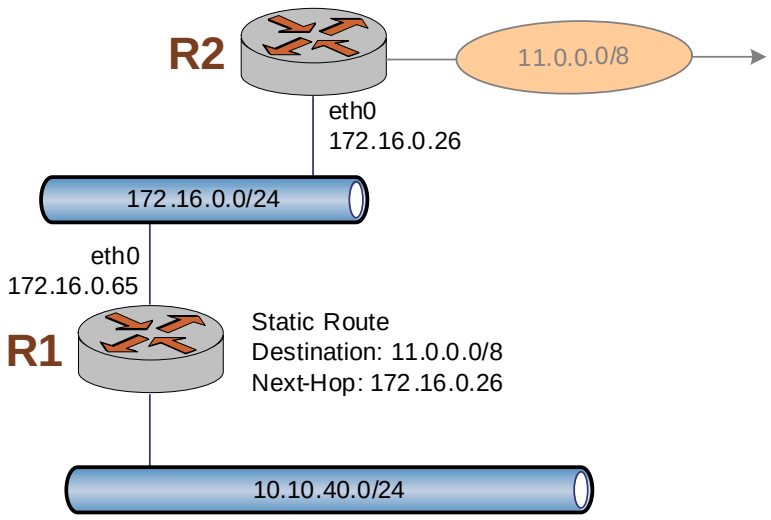
The collection of all routes the router has learned from its configuration or from its dynamic routing protocols is stored in its Routing Information Base (RIB).

Unicast routes are directly used to determine the forwarding table used for unicast packet forwarding.

Configuring Static Routes

In this example, a sample configurations are presented for basic static routes. When you are finished, the system will be configured as shown in Figure 2-1. In this example, a static route is created that says, in effect, “any packets destined for the 11.0.0.0/8 network should be forwarded to 172.16.0.26”.

Figure 2-1 Static routes



This section includes the following examples:

- Example 2-1 Creating a static route

Example 2-1 creates a static route to network 11.0.0.0/8 directed towards 172.16.0.26.

To create a static route, perform the following steps in configuration mode:

Example 2-1 Creating a static route

Step	Command
Create a static route to R2.	`vyatta@R1# set protocols static route 11.0.0.0/8 next-hop 172.16.0.26`
Commit the configuration.	`vyatta@R1# commit`

Floating Static Routes

Usually, static routes have a relatively short administrative distance—typically 1, and usually shorter than the administrative distances for dynamic (learned) routes. A “floating” static route is a static route with an administrative distance greater than that for dynamic routes.

You can configure a static route to be a floating route by setting the administrative distance higher than the distance applied to the routes in your dynamic routing protocol. This renders the static route less desirable than a dynamic route. At the same time, if the dynamic route is lost, the static route is available to take over traffic, which can be forwarded through the static route as an alternate path.

Showing Static Routes in the Routing Table

To display route information, use the **show ip route** command. To show just static routes, use the **show ip route static** filter, as shown in Example 2-2.

Example 2-2 Showing static routes in the routing table

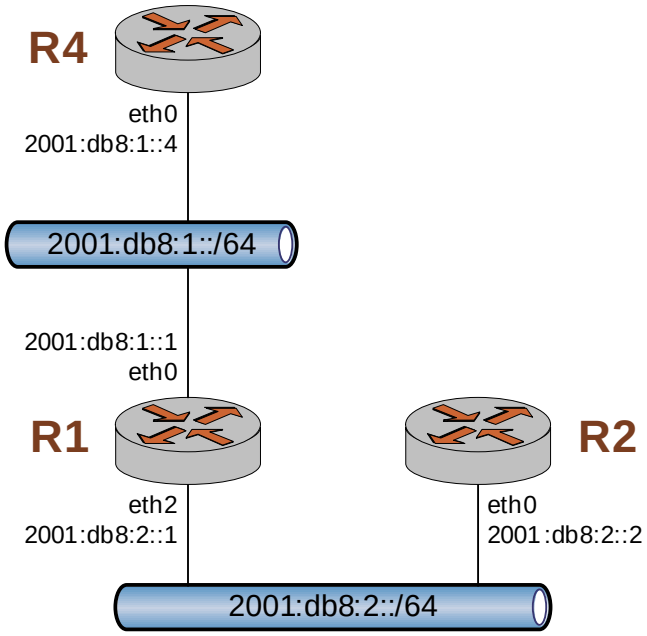
```
vyatta@R1:~$ show ip route static
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

S>* 11.0.0.0/8 [1/0] via 172.16.0.26, eth0
vyatta@R1:~$
```

Static IPv6 Route Configuration

Figure 2-2 shows an IPv6 network with three nodes. In this example we will show configuration of the nodes using static routes to enable R2 and R4 to communicate via R1.

Figure 2-2 Static IPv6 routing example



Verify That IPv6 Forwarding is Enabled

In order for R1 to be able to pass data between interfaces eth0 and eth2 (i.e., between R4 and R2) it must be configured to enable forwarding. To determine if forwarding is enabled, perform the following step in operational mode.

Example 2-3 Determine if forwarding is enabled on R1

Step	Command
Display the state of IPv6 forwarding on R1.	`vyatta@R1:~$ show ipv6 forwarding` `ipv6 forwarding is off`

If forwarding is not enabled, as is the case in Example 2-3, the system must be configured to enable forwarding. To enable forwarding, perform the following steps in configuration mode.

Example 2-4 Enable forwarding on R1

Step	Command
Enable forwarding on R1.	`vyatta@R1# delete system ipv6 disable-forwarding`

Example 2-4 Enable forwarding on R1

Commit the change.	`vyatta@R1# commit`

Change to operational mode	`vyatta@R1# exit`
	`exit`
	`vyatta@R1:~$`

Display the state of IPv6 forwarding on R1.	`vyatta@R1:~$ show ipv6 forwarding` `ipv6 forwarding is on`

Add the Default IPv6 Route

On R4, all traffic that is not routed elsewhere will be sent to R1. To configure the default route, perform the following steps in configuration mode.

Example 2-5 Add the default route on R4

Step	Command
Add the default route on R4.	`vyatta@R4# set protocols static route6 ::/0 next-hop 2001:db8:1::1`
Commit the change.	`vyatta@R4# commit`

Change to operational mode.	`vyatta@R4# exit`
	`exit`
	`vyatta@R4:~$`

Verify the default route in the routing table.	`vyatta@R4:~$ show ipv6 route`
	`Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,`
	`I - ISIS, B - BGP, * - FIB route.`
	
	`S>* ::/0 [1/0] via 2001:db8:1::1, eth0`
	`C>* ::1/128 is directly connected, lo`
	`C>* 2001:db8:1::/64 is directly connected, eth0`
	`C * fe80::/64 is directly connected, eth1`
	`C>* fe80::/64 is directly connected, eth0`
	`K>* ff00::/8 is directly connected, eth0`

Add a Static IPv6 Route

As an alternative to the default route we created on R4, we'll create a static route on R2. To configure a static route to the 2001:db8:1::/64 network, perform the following steps in configuration mode.

Example 2-6 Add a static route on R2

Step	Command
Add a static route on R2.	vyatta@R1# **set protocols static route6 2001:db8:1::/64 next-hop 2001:db8:2::1**
Commit the change.	vyatta@R1# commit
Change to operational mode.	vyatta@R1# **exit** exit vyatta@R2:~$
Verify the static route in the routing table.	vyatta@R2:~$ **show ipv6 route** Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3, I - ISIS, B - BGP, * - FIB route. C>* ::1/128 is directly connected, lo S>* 2001:db8:1::/64 [1/0] via 2001:db8:2::1, eth0 C>* 2001:db8:2::/64 is directly connected, eth0 C * fe80::/64 is directly connected, eth1 C>* fe80::/64 is directly connected, eth0 K>* ff00::/8 is directly connected, eth0

Confirm Connectivity

To confirm that R2 and R4 can communicate, use the **ping** command. To confirm connectivity between R2 and R4, perform the following step in operational mode.

Example 2-7 Confirm connectivity between R2 and R4

Step	Command
Ping R4 from R2.	vyatta@R2:~$ **ping 2001:db8:1::4** PING 2001:db8:1::4(2001:db8:1::4) 56 data bytes 64 bytes from 2001:db8:1::4: icmp_seq=1 ttl=63 time=5.65 ms 64 bytes from 2001:db8:1::4: icmp_seq=2 ttl=63 time=0.382 ms ^C --- 2001:db8:1::4 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1011ms rtt min/avg/max/mdev = 0.382/3.016/5.650/2.634 ms

As an alternative, use **traceroute** to verify that the goes from R2 to R1 to R4. To confirm connectivity between R2 and R4 through R1 using **traceroute**, perform the following step in operational mode.

Example 2-8 Confirm connectivity between R2 and R4 via R1

Step	Command
Trace the route from R2 to R4.	vyatta@R2:~$ **traceroute 2001:db8:1::4** traceroute to 2001:db8:1::4 (2001:db8:1::4), 30 hops max, 40 byte packets 1 (2001:db8:2::1) 4.448 ms 4.148 ms 4.092 ms 2 (2001:db8:1::4) 4.297 ms 4.306 ms 4.308 ms

Static Route Commands

This chapter contains the following commands.

Configuration Commands	
protocols static interface-route <subnet> next-hop-interface <interface>	Allows you to configure the next-hop interface for an interface-based static route.
protocols static interface-route6 <subnet> next-hop-interface <ethx>	Allows you to configure the next hop interface for an interface-based IPv6 static route.
protocols static route <subnet> blackhole	Allows you to configure a “black-hole” static route.
protocols static route <subnet> next-hop <address>	Allows you to configure the next hop for a static route.
protocols static route6 <subnet> blackhole	Allows you to configure a blackhole IPv6 static route.
protocols static route6 <subnet> next-hop <address>	Allows you to configure the next hop for an IPv6 static route.
Operational Commands	
show ip route static	Displays static routes. *See page 20.*
show ipv6 route static	Displays IPv6 static routes. *See page 20.*

protocols static interface-route <subnet> next-hop-interface <interface>

Allows you to configure the next-hop interface for an interface-based static route.

Syntax

set protocols static interface-route *subnet* **next-hop-interface** *interface* [**disable** | **distance** *distance*]

delete protocols static interface-route *subnet* **next-hop-interface** *interface* [**disable** | **distance**]

show protocols static interface-route *subnet* **next-hop-interface** *interface* [**disable** | **distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        interface-route subnet {
            next-hop-interface interface {
                disable
                distance distance
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an interface-based static route. The format is a destination subnet of the form *address/prefix*. You can define multiple interface-based routes by creating multiple **interface-route** configuration nodes.
interface	Mandatory. The next-hop interface.
disable	Disables the interface-based static route.

distance	Optional. Sets the next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The range is 1 to 255. The default is 1.

Default

None.

Usage Guidelines

Use this command to configure interface-based static routes on the router.

Use the **set** form of this command to specify the next-hop interface for the route.

Use the **delete** form of this command to remove the next-hop interface.

Use the **show** form of this command to view the next-hop interface for the route.

protocols static interface-route6 <subnet> next-hop-interface <ethx>

Allows you to configure the next hop interface for an interface-based IPv6 static route.

Syntax

set protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable | distance** *distance*]

delete protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable | distance**]

show protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable | distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        interface-route6 subnet {
            next-hop-interface ethx {
                disable
                distance distance
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an interface-based static route. The format is a destination subnet of the form IPv6-*address/prefix*. You can define multiple interface-based routes by creating multiple **interface-route6** configuration nodes.
ethx	Mandatory. The next hop Ethernet interface.

disable	Disables the interface-based IPv6 static route.
distance	Optional. Defines the next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Default

None.

Usage Guidelines

Use this command to configure interface-based IPv6 static routes on the system.

Use the **set** form of this command to specify the next hop interface for the route.

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

Use the **show** form of this command to view the next hop interface for the route.

protocols static route <subnet> blackhole

Allows you to configure a “black-hole” static route.

Syntax

set protocols static route *subnet* **blackhole** [**distance** *distance*]

delete protocols static route *subnet* **blackhole** [**distance**]

show protocols static route *subnet* **blackhole** [**distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        route subnet {
            blackhole {
                distance distance
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines a static route. The format is a destination subnet of the form *address/prefix*. You can define multiple static routes by creating multiple **route** configuration nodes.
distance	Optional. Defines the black-hole distance for this route. Routes with a smaller distance are selected before those with a larger distance. The range is 1 to 255. The default is 1.

Default

None.

Usage Guidelines

Use this command to configure a “black-hole” static route on the router. A black-hole route is a route for which the system silently discard packets that are matched.

Use the **set** form of this command to set a black-hole route.

Use the **delete** form of this command to remove a black-hole route.

Use the **show** form of this command to view black-hole route configuration.

protocols static route <subnet> next-hop <address>

Allows you to configure the next hop for a static route.

Syntax

set protocols static route *subnet* **next-hop** *address* [**disable** | **distance** *distance*]

delete protocols static route *subnet* **next-hop** *address* [**disable** | **distance**]

show protocols static route *subnet* **next-hop** *address* [**disable** | **distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        route subnet {
            next-hop address {
                disable
                distance distance
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines a static route. The format is a destination subnet of the form *address/prefix*. You can define multiple static routes by creating multiple **route** configuration nodes.
address	Mandatory. The address of the next-hop router.
disable	Disables the static route.

distance	Optional. Defines the next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The range is 1 to 255. The default is 1.

Default

None.

Usage Guidelines

Use this command to configure static routes on the router.

Use the **set** form of this command to specify the next hop for the route.

Use the **delete** form of this command to remove the static route next hop.

Use the **show** form of this command to view static route next-hop configuration.

protocols static route6 <subnet> blackhole

Allows you to configure a blackhole IPv6 static route.

Syntax

set protocols static route6 *subnet* **blackhole** [**distance** *distance*]

delete protocols static route6 *subnet* **blackhole** [**distance**]

show protocols static route6 *subnet* **blackhole** [**distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        route6 subnet {
            blackhole {
                distance distance
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an IPv6 static route. The format is a destination subnet of the form IPv6-*address*/*prefix*. You can define multiple static routes by creating multiple **route** configuration nodes.
distance	Optional. Defines the blackhole distance for this route. Routes with a smaller distance will be selected before those with a larger distance.

Default

None.

Usage Guidelines

Use this command to configure a blackhole IPv6 static route. A blackhole route silently discards packets that are matched.

Use the **set** form of this command to specify a blackhole IPv6 static route.

Use the **delete** form of this command to remove a blackhole IPv6 static route.

Use the **show** form of this command to view blackhole IPv6 static route configuration.

protocols static route6 <subnet> next-hop <address>

Allows you to configure the next hop for an IPv6 static route.

Syntax

set protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** *distance* | **interface** *interface*]

delete protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** | **interface**]

show protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** | **interface**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
    static {
        route6 subnet {
            next-hop address {
                disable
                distance distance
                interface interface
            }
        }
    }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an IPv6 static route. The format is a destination subnet of the form IPv6-*address*/*prefix*. You can define multiple static routes by creating multiple **route6** configuration nodes.
address	Mandatory. The IPv6 address of the next hop router.
disable	Disable the IPv6 static route.

distance	Optional. Defines the next-hop distance for this route. Routes with a smaller distance will be selected before those with a larger distance.
interface	Optional. The outgoing interface used to reach the next-hop address. This is necessary when the next-hop address is a link-local address (that is, it has a fe80::/64 prefix).

Default

None.

Usage Guidelines

Use this command to configure IPv6 static routes on the system.

Use the **set** form of this command to specify the next hop for the route.

Use the **delete** form of this command to remove the static route next hop.

Use the **show** form of this command to view static route next hop configuration.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6

DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
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
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
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
P2P	peer-to-peer
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
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service

Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access