BROCADE

CONFIGURATION GUIDE

Brocade 5600 vRouter Basic Routing Configuration Guide

Supporting Brocade 5600 vRouter 4.2 R1

53-1004246-01
16 May 2016

Contents

Preface 7
- Document conventions 7
  - Text formatting conventions 7
  - Command syntax conventions 7
  - Notes, cautions, and warnings 8
- Brocade resources 8
- Contacting Brocade Technical Support 8
  - Brocade customers 8
  - Brocade OEM customers 9
- Document feedback 9

About This Guide 11

Forwarding and Routing Commands 13
- clear ip prefix-list 15
- clear ipv6 prefix-list 16
- monitor command <traceroute-command> 17
- ping <host> 18
- ping <host> adaptive <option> 19
- ping <host> allow-broadcast <option> 22
- ping <host> audible <option> 25
- ping <host> bypass-route <option> 28
- ping <host> count <option> 31
- ping <host> deadline <seconds> <option> 34
- ping <host> ether-size <bytes> <option> 36
- ping <host> flood <option> 38
- ping <host> interface <host> <option> 40
- ping <host> interval <seconds> <option> 43
- ping <host> mark <fwmark> <option> 46
- ping <host> mtu-discovery < do | dont | want > <option> 48
- ping <host> no-loopback <option> 51
- ping <host> numeric <option> 53
- ping <host> pattern <hexadecimal-digit> <option> 56
- ping <host> quiet <option> 59
- ping <host> record-route <option> 61
- ping <host> size <bytes> <option> 63
- ping <host> tos <number> <option> 66
- ping <host> ttl <seconds> <option> 68
- ping <host> verbose <option> 71
- protocols nsm log 73
- protocols nsm log ha 75
- reset ip route kernel 76
- reset ipv6 route kernel 77
- resources group address-group <group-name> 78
- resources group icmp-group <group-name> 79
- resources group icmpv6-group <group-name> 81
- resources group port-group <group-name> 83

show ip forwarding....85
show ip route....86
show ip route <ipv4net> longer-prefixes....88
show ip route connected....89
show ip route forward....90
show ip route kernel....92
show ip route static....93
show ip route summary....94
show ip route supernets-only....95
show ip route table <table>....96
show ip route variance....97
show ip route variance console....98
show ipv6 route....100
show ipv6 route <ipv6net> longer-prefixes....101
show ipv6 route bgp....102
show ipv6 route connected....103
show ipv6 route forward....104
show ipv6 route kernel....105
show ipv6 route ripng....106
show ipv6 route static....107
show ipv6 route variance....108
show ipv6 route variance console....109
show monitoring protocols rib....110
traceroute <host> as-path....111
traceroute <host> bypass-routing....113
traceroute <host> debug-socket....115
traceroute <host> first-ttl <value>....118
traceroute <host> gateway <address>....120
traceroute <host> icmp-echo....122
traceroute <host> icmp-extensions....124
traceroute <host> interface <value>....126
traceroute <host> max-ttl <value>....128
traceroute <host> interval <value>....130
traceroute <host> max-ttl <value>....132
traceroute <host> no-fragment....134
traceroute <host> num-queries <num>....136
traceroute <host> port <number>....138
traceroute <host> seq-queries <number>....140
traceroute <host> source-addr <host>....142
traceroute <host> tcp-syn....144
traceroute <host> tos <value>....146
traceroute <host> version....148
traceroute <host> wait-time <value>....150
traceroute <protocol> <host>....152
traceroute <host>....153

ECMP....155
ECMP overview....155

ECMP Commands....157
protocols ecmp disable....157

protocols ecmp maximum-paths....158
protocols ecmp mode <mode>....159
show dataplane route....161
show dataplane route6....162

Static Routes....163
Static route configuration....163
Static routes overview....163
Configuring static routes....163
Creating floating static routes....164
Showing static routes in the routing table....165
Static IPv6 route configuration....165
Verify that IPv6 forwarding is enabled....166
Add the default IPv6 route....166
Add a static IPv6 route....167
Confirm connectivity....167

Static Route Commands....169
protocols static interface-route <subnet> next-hop-interface <interface>....170
protocols static interface-route6 <subnet> next-hop-interface <interface>....171
protocols static route <subnet> blackhole <distance>....172
protocols static route <subnet> next-hop <address>....173
protocols static route6 <subnet> blackhole....174
protocols static route6 <subnet> next-hop <address>....175
protocols static table <table> interface-route <subnet> next-hop-interface <interface>....177
protocols static table <table> route <subnet> blackhole <distance>....179
protocols static table <table> route <subnet> next-hop <address>....180
protocols static table <table> route6 <subnet> next-hop <address>....182
protocols static table <table> route6 <subnet> blackhole [distance]....184

Source Routes....187
Source routing example....187

List of Acronyms....191

Preface

- Document conventions 7
- Brocade resources 8
- Contacting Brocade Technical Support 8
- Document feedback 9

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables
	Identifies document titles
`Courier font`	Identifies CLI output
	Identifies command syntax examples

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional.
	Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
	In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.

Convention	Description
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads.

White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • My Cases through MyBrocade • Software downloads and licensing tools • Knowledge Base	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • For areas unable to access toll free number: +1-408-333-6061 • Toll-free numbers are available in many countries.	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Guide

This guide describes information about forwarding and routing on the Brocade 5600 vRouter (referred to as a virtual router, vRouter, or router in the guide).

Forwarding and Routing Commands

- clear ip prefix-list 15
- clear ipv6 prefix-list 16
- monitor command <traceroute-command> 17
- ping <host> 18
- ping <host> adaptive <option> 19
- ping <host> allow-broadcast <option> 22
- ping <host> audible <option> 25
- ping <host> bypass-route <option> 28
- ping <host> count <option> 31
- ping <host> deadline <seconds> <option> 34
- ping <host> ether-size <bytes> <option> 36
- ping <host> flood <option> 38
- ping <host> interface <host> <option> 40
- ping <host> interval <seconds> <option> 43
- ping <host> mark <fwmark> <option> 46
- ping <host> mtu-discovery < do | dont | want > <option> 48
- ping <host> no-loopback <option> 51
- ping <host> numeric <option> 53
- ping <host> pattern <hexadecimal-digit> <option> 56
- ping <host> quiet <option> 59
- ping <host> record-route <option> 61
- ping <host> size <bytes> <option> 63
- ping <host> tos <number> <option> 66
- ping <host> ttl <seconds> <option> 68
- ping <host> verbose <option> 71
- protocols nsm log 73
- protocols nsm log ha 75
- reset ip route kernel 76
- reset ipv6 route kernel 77
- resources group address-group <group-name> 78
- resources group icmp-group <group-name> 79
- resources group icmpv6-group <group-name> 81
- resources group port-group <group-name> 83
- show ip forwarding 85
- show ip route 86
- show ip route <ipv4net> longer-prefixes 88
- show ip route connected 89
- show ip route forward 90
- show ip route kernel 92
- show ip route static 93
- show ip route summary 94
- show ip route supernets-only 95
- show ip route table <table> 96
- show ip route variance 97
- show ip route variance console 98
- show ipv6 route 100
- show ipv6 route <ipv6net> longer-prefixes 101
- show ipv6 route bgp 102
- show ipv6 route connected 103

- show ipv6 route forward....104
- show ipv6 route kernel....105
- show ipv6 route ripng....106
- show ipv6 route static....107
- show ipv6 route variance....108
- show ipv6 route variance console....109
- show monitoring protocols rib....110
- traceroute <host> as-path....111
- traceroute <host> bypass-routing....113
- traceroute <host> debug-socket....115
- traceroute <host> first-ttl <value>....118
- traceroute <host> gateway <address>....120
- traceroute <host> icmp-echo....122
- traceroute <host> icmp-extensions....124
- traceroute <host> interface <value>....126
- traceroute <host> max-ttl <value>....128
- traceroute <host> interval <value>....130
- traceroute <host> max-ttl <value>....132
- traceroute <host> no-fragment....134
- traceroute <host> num-queries <num>....136
- traceroute <host> port <number>....138
- traceroute <host> seq-queries <number>....140
- traceroute <host> source-addr <host>....142
- traceroute <host> tcp-syn....144
- traceroute <host> tos <value>....146
- traceroute <host> version....148
- traceroute <host> wait-time <value>....150
- traceroute <protocol> <host>....152
- traceroute <host>....153

clear ip prefix-list

Clears statistics for or the status of a prefix list.

Syntax

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

Command Default

Statistics for or the status of all prefix lists is cleared.

Parameters

list-name

Optional. A prefix list.

ipv4net

Optional. A network.

Modes

Operational mode

Usage Guidelines

Use this command to clear statistics for or the status of a prefix list.

clear ipv6 prefix-list

Clears statistics for or the status of an IPv6 prefix list.

Syntax

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

Command Default

Statistics for or the status of all IPv6 prefix lists is cleared.

Parameters

list-name

Optional. An IPv6 prefix list.

ipv6net

Optional. An IPv6 network.

Modes

Operational mode

Usage Guidelines

Use this command to clear statistics for or the status of an IPv6 prefix list.

monitor command <traceroute-command>

Monitors a traceroute command.

Syntax

monitor command *traceroute-command*

run monitor command *traceroute-command*

Parameters

traceroute-command

The **traceroute** command to be monitored. The **traceroute** command must be enclosed in quotation marks.

Modes

Operational mode

Configuration mode

Usage Guidelines

Use this command to display the output of a **traceroute** command. The display information is refreshed every two seconds.

Use the **run** form of this command in configuration mode.

ping <host>

Sends Internet Control Message Protocol (ICMP) ECHO_REQUEST packets to a network host.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* }

Parameters

ipv4_address

Pings the IPv4 address of the host.

ipv6_address

Pings the IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

ipv4 ipv6

When using the **ping** command for fault isolation, enter the command on the local host to verify that the local network interface is up and running. Then, ping hosts and gateways farther away. 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, and maximum round-trip times.

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

Examples

This example shows how to test whether the network host, www.google.com is reachable.

```
vyatta@VR-2:/etc$ ping www.google.com
PING www.google.com (216.58.196.100) 56(84) bytes of data.
64 bytes from maa03s19-in-f4.1e100.net (216.58.196.100): icmp_req=1 ttl=54 time=42.3 ms
^C
--- www.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 42.364/42.364/42.364/0.000 ms
```

ping <host> adaptive <option>

Sets the interpacket interval adaptively such that the interpacket interval adjusts to round-trip time.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **adaptive** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

adaptive

Sets the interpacket interval adaptively such that the interpacket interval adjusts to round-trip time. The **adaptive** setting ensures that not more than one (or more, if preload is set) unanswered probes are present in the network.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

ipv4 ipv6

When using the **ping** command for fault isolation, enter the command on the local host to verify that the local network interface is up and running. Then, ping hosts and gateways farther away. 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, and maximum round-trip times.

NOTE
The minimum interpacket interval is 200 ms for all users, except for a super-user.

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

Examples

This example shows how to test whether the www.google.com network is reachable with an interpacket interval. The example also displays the time stamp in the ping output.

```
vyatta@vyatta:~$ ping www.google.com adaptive timestamp audible
PING www.google.com (216.58.216.164) 56(84) bytes of data.
[1428886179.416698] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=2 ttl=54
time=20.0 ms
[1428886179.457896] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=4 ttl=54
time=20.0 ms
[1428886179.499170] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=6 ttl=54
time=20.2 ms
[1428886179.539836] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=8 ttl=54
time=19.9 ms
[1428886179.580788] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=10 ttl=54
time=19.9 ms
[1428886179.621507] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=12 ttl=54
time=20.0 ms
[1428886179.662363] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=14 ttl=54
time=19.8 ms
[1428886179.703528] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=16 ttl=54
time=20.1 ms
[1428886179.744554] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=18 ttl=54
time=20.0 ms
[1428886179.785702] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=20 ttl=54
time=20.1 ms
[1428886179.826861] 64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=22 ttl=54
time=20.2 ms
[1428886181.384621] From 10.18.170.201 icmp_seq=1 Destination Host Unreachable
[1428886181.385770] From 10.18.170.201 icmp_seq=3 Destination Host Unreachable
[1428886181.386512] From 10.18.170.201 icmp_seq=5 Destination Host Unreachable
[1428886181.387046] From 10.18.170.201 icmp_seq=7 Destination Host Unreachable
[1428886181.387599] From 10.18.170.201 icmp_seq=9 Destination Host Unreachable
[1428886181.388177] From 10.18.170.201 icmp_seq=11 Destination Host Unreachable
[1428886181.388707] From 10.18.170.201 icmp_seq=13 Destination Host Unreachable
[1428886181.389269] From 10.18.170.201 icmp_seq=15 Destination Host Unreachable
[1428886181.389865] From 10.18.170.201 icmp_seq=17 Destination Host Unreachable
[1428886181.390681] From 10.18.170.201 icmp_seq=19 Destination Host Unreachable
[1428886181.391494] From 10.18.170.201 icmp_seq=21 Destination Host Unreachable
[216.164): icmp_req=165 ttl=54 time=20.4 ms
```

ping <host> allow-broadcast <option>

Allows the pinging of a broadcast address.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **allow-broadcast** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

allow-broadcast

Allows pinging a broadcast address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies the interface that the device must use as source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode.

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

ipv4 ipv6

When using the **ping** command for fault isolation, enter the command on the local host to verify that the local network interface is up and running. Then, ping hosts and gateways farther away. 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, and maximum round-trip times.

When the **ping** command is interrupted by typing `<Ctrl>+c`s, a brief statistical summary is displayed.

Examples

This example shows how to test whether www.google.com network is reachable. It also shows how to allow the pinging of the broadcast address and define a life-time of 30 hops for each host.

```
vyatta@vyatta:~$ ping www.google.com allow-broadcast ttl
Possible completions:
  <hops>    Number of hops

vyatta@vyatta:~$ ping www.google.com allow-broadcast ttl 30
PING www.google.com (216.58.216.164) 56(84) bytes of data.
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=2 ttl=54 time=20.1 ms
From 10.18.170.201 icmp_seq=1 Destination Host Unreachable
From 10.18.170.201 icmp_seq=3 Destination Host Unreachable
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=4 ttl=54 time=20.2 ms
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=6 ttl=54 time=20.3 ms
From 10.18.170.201 icmp_seq=5 Destination Host Unreachable
From 10.18.170.201 icmp_seq=7 Destination Host Unreachable
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=8 ttl=54 time=19.9 ms
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=10 ttl=54 time=20.4 ms
From 10.18.170.201 icmp_seq=9 Destination Host Unreachable
From 10.18.170.201 icmp_seq=11 Destination Host Unreachable
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=12 ttl=54 time=20.0 ms
```

ping <host> audible <option>

Makes a beep sound when the router pings for the host details.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **audible** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

audible

Makes a beep sound while the device pings for the host details.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies the interface that the device must use as source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

ipv4 ipv6

When using the **ping** command for fault isolation, enter the command on the local host to verify that the local network interface is up and running. Then, ping hosts and gateways farther away. 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, and maximum round-trip times.

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

Examples

This example shows how to configure the router to send an ICMP_ECHO request five times, making a beep sound on every ping.

```
vyatta@vyatta:~$ ping www.google.com audible count 5
PING www.google.com (216.58.216.164) 56(84) bytes of data.
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=2 ttl=54 time=20.0 ms
From 10.18.170.201 icmp_seq=1 Destination Host Unreachable
From 10.18.170.201 icmp_seq=3 Destination Host Unreachable
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=4 ttl=54 time=20.1 ms

--- www.google.com ping statistics ---
5 packets transmitted, 2 received, +2 errors, 60% packet loss, time 4010ms
rtt min/avg/max/mdev = 20.048/20.075/20.103/0.144 ms, pipe 3
vyatta@vyatta:~$
```

ping <host> bypass-route <option>

Bypasses the normal routing tables and sends a ping request directly to a host on an attached interface.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **bypass-route** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

bypass-route

Bypasses the normal routing tables and sends directly to a host on an attached interface. If the host is not directly attached to a network, an error is returned.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

If the host is not directly attached to a network, an error is returned.

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

Examples

This example shows how to bypass the normal routing tables and send a ping request directly to a host on an attached interface.

```
vyatta@vyatta:~$ ping www.google.com bypass-route
PING www.google.com (216.58.216.164) 56(84) bytes of data.
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
```

ping <host> count <option>

Specifies a number of ping requests that the router must send.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **count** *numberoption*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

number

The number of ping requests to send.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Supress loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode.

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to test whether www.google.com network is reachable by sending five ICMP_ECHO requests and that the router waits for five reply packets for ten seconds.

```
vyatta@vyatta:~$ ping www.google.com count 5 deadline 10
PING www.google.com (216.58.216.164) 56(84) bytes of data.
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=2 ttl=54 time=20.0 ms
From 10.18.170.201 icmp_seq=1 Destination Host Unreachable
From 10.18.170.201 icmp_seq=3 Destination Host Unreachable
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=4 ttl=54 time=20.1 ms

--- www.google.com ping statistics ---
5 packets transmitted, 2 received, +2 errors, 60% packet loss, time 4010ms
rtt min/avg/max/mdev = 20.048/20.075/20.103/0.144 ms, pipe 3

vyatta@vyatta:~$ ping www.google.com count 5
```

ping <host> deadline <seconds> <option>

Specifies the number of seconds before which the ping expires.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **deadline** *seconds option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

seconds

The number of seconds before which ping exits.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to test whether www.google.com network is reachable within three seconds.

```
vyatta@vyatta:~$ ping www.google.com deadline 3
PING www.google.com (216.58.216.164) 56(84) bytes of data.
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=1 ttl=54 time=19.8 ms
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=3 ttl=54 time=20.1 ms

--- www.google.com ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2004ms
rtt min/avg/max/mdev = 19.873/19.995/20.118/0.187 ms
```

ping <host> ether-size <bytes> <option>

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **ether-size** *bytes option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

bytes

The number of bytes.

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound during every ping, while the router pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies the interface that the device must use as source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command when used with the ether-size option specifies the size of the resultant Layer 3 packet, such as ICMP data plus ICMP headers, IP headers, and so on.

The **ping** *host* **ether-size** command ensures that the resultant size supports Layer 3 IP packet size. The Ethernet MTU is 1,500 bytes. Therefore, the **ping** *host* **ether-size** command subtracts 28 bytes from the host size of the Layer 3 packet to ensure that the resultant value matches the overall size of the data packet. The **ping** *host* **ether-size** command avoids the fragmentation overhead due to the MTU.

NOTE

You can use either the **ping** *host* **size** or **ping** *host* **ether-size** command to ping a host network. You cannot use both commands simultaneously.

ping <host> flood <option>

Sends 100 ping requests each second.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **flood** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes noise while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies the interface that the device must use as source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to test whether www.google.com network is reachable by sending a maximum of 100 ping requests each second.

```
vyatta@vyatta:~$ ping www.google.com flood
PING www.google.com (216.58.216.164) 56(84) bytes of data.
..........E..........E..........E..........E..........E..........E..........E.......
....E..........E..........E..........E..........E..........E.......^C
--- www.google.com ping statistics ---
345 packets transmitted, 169 received, +168 errors, 51% packet loss, time 42361ms
rtt min/avg/max/mdev = 19.917/20.779/33.826/1.092 ms, pipe 24, ipg/ewma 123.144/20.398 ms
```

ping <host> interface <host> <option>

Specifies an interface that the device must use as the source address.

Syntax

ping { *ipv4_addres* | *ipv6_address* | *hostname* } **interface** { *ipv4_addres* | *ipv6_address* | *hostname* } *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

Use the **ping** *host* **interface** command when pinging IPv6 link-local address.

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

Examples

This example shows how to test whether an IPv4 interface is reachable.

```
vyatta@VR-1:~$ ping 192.1.2.2 interface dp0s6
PING 192.1.2.2 (192.1.2.2) from 192.1.2.1 dp0s6: 56(84) bytes of data.
64 bytes from 192.1.2.2: icmp_req=1 ttl=64 time=1.66 ms
^C
--- 192.1.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.664/1.664/1.664/0.000 ms
vyatta@VR-1:~$ ping 192.1.2.2 interface   192.1.2.1
PING 192.1.2.2 (192.1.2.2) from 192.1.2.1 : 56(84) bytes of data.
64 bytes from 192.1.2.2: icmp_req=1 ttl=64 time=1.02 ms
^C
```

This example shows how to test whether an IPv6 interface is reachable.

```
vyatta@VR-1:~$ ping  2012:dead::2 interface  2012:dead::1
PING 2012:dead::2(2012:dead::2) from 2012:dead::1 : 56 data bytes
64 bytes from 2012:dead::2: icmp_seq=1 ttl=64 time=3.04 ms
64 bytes from 2012:dead::2: icmp_seq=2 ttl=64 time=1.01 ms
^C
--- 2012:dead::2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.012/2.027/3.043/1.016 ms
```

ping <host> interval <seconds> <option>

Specifies the time in seconds for which the device must wait between ping requests.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **interval** *secondsoption*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

seconds

The number of seconds for which the device must wait between ping requests.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to test whether www.google.com network is reachable by providing 3 seconds of time between 5 ping attempts.

```
vyatta@vyatta:~$ ping www.google.com interval 3 count 15
PING www.google.com (216.58.216.164) 56(84) bytes of data.
64 bytes from sea15s02-in-f4.1e100.net (216.58.216.164): icmp_req=1 ttl=54 time=20.2 ms
From 10.18.170.201 icmp_seq=2 Destination Host Unreachable
From 10.18.170.201 icmp_seq=3 Destination Host Unreachable
From 10.18.170.201 icmp_seq=4 Destination Host Unreachable
From 10.18.170.201 icmp_seq=5 Destination Host Unreachable
From 10.18.170.201 icmp_seq=6 Destination Host Unreachable
From 10.18.170.201 icmp_seq=7 Destination Host Unreachable
From 10.18.170.201 icmp_seq=8 Destination Host Unreachable
From 10.18.170.201 icmp_seq=9 Destination Host Unreachable
From 10.18.170.201 icmp_seq=10 Destination Host Unreachable
From 10.18.170.201 icmp_seq=11 Destination Host Unreachable
From 10.18.170.201 icmp_seq=12 Destination Host Unreachable
From 10.18.170.201 icmp_seq=13 Destination Host Unreachable
From 10.18.170.201 icmp_seq=14 Destination Host Unreachable
From 10.18.170.201 icmp_seq=15 Destination Host Unreachable

--- www.google.com ping statistics ---
15 packets transmitted, 1 received, +14 errors, 93% packet loss, time 41999ms
rtt min/avg/max/mdev = 20.264/20.264/20.264/0.000 ms
```

ping <host> mark <fwmark> <option>

Specifies that the device must consider the ping request for special processing.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **mark** *fwmark option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

fwmark

Marks the outgoing packet.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

The **ping** *host* **mark** *fwmark* command is used in usecases within the operating system. For example, to tag the outgoing packets while configuring policy routing, to select specific outbound processing.

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

ping <host> mtu-discovery < do | dont | want > <option>

Selects the discovery strategy of the path maximum transmission unit (PMTU).

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **mtu-discovery** { **do** | **dont** | **want** } *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

do

Prohibits fragmentation, even for the local packet. Sets a do-not fragment (DF) flag to the router.

want

Performs a PMTU discovery. During the discovery the device fragments the packet locally.

dont

Prohibits fragmentation, but does not set a DF flag.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval
Specifies the time in seconds for which the device must wait between ping requests.

mark
Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** *host* **mtu-discovery** command selects the discovery strategy of the PMTU, based on the parameters provided. The command checks the size of the packet. If the size is equal to or greater than the maximum data payload that is available in a packet, the device determines whether the packet has to be fragmented, based on the discovery strategy of the PMTU.

Examples

This example shows how to test 1472 bytes of data of the host 10.0.0.103 for network reachability while prohibiting network fragmentation during the ping. However a do-not-fragment flag is set for the router.

```
vyatta@VR-1:~$ ping 10.0.0.103 size 1472 mtu-discovery do
PING 10.0.0.103 (10.0.0.103) 1472(1500) bytes of data.
1480 bytes from 10.0.0.103: icmp_req=1 ttl=64 time=0.923 ms
^[[A1480 bytes from 10.0.0.103: icmp_req=2 ttl=64 time=1.35 ms
^C
--- 10.0.0.103 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.923/1.137/1.352/0.217 ms
```

This example shows how to test 1472 bytes of data of the host 10.0.0.103 for network reachability while prohibiting network fragmentation during the ping. However a do-not-fragment flag is not set for the router.

```
vyatta@VR-1:~$ ping 10.0.0.103 size 1472 mtu-discovery dont
PING 10.0.0.103 (10.0.0.103) 1472(1500) bytes of data.
1480 bytes from 10.0.0.103: icmp_req=1 ttl=64 time=1.25 ms
^C
--- 10.0.0.103 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.250/1.250/1.250/0.000 ms
```

This example shows how to perform a discovery strategy for PMTU during a ping request. During the discovery, the router fragments the packet locally.

```
vyatta@VR-1:~$ ping 10.0.0.103 size 1472 mtu-discovery want
PING 10.0.0.103 (10.0.0.103) 1472(1500) bytes of data.
1480 bytes from 10.0.0.103: icmp_req=1 ttl=64 time=1.00 ms
^C
--- 10.0.0.103 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.005/1.005/1.005/0.000 ms
```

ping <host> no-loopback <option>

Suppresses loopback of multicast pings.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname*} **no-loopback** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes noise while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

ping <host> numeric <option>

Specifies that the router must not resolve domain name system (DNS) names during a ping.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **numeric** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

numeric

The number of ping requests to send.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to ensure that a ping command provides only numeric output, which means that the router does not look up symbolic names for host addresses.

```
vyatta@VR-2:/etc$ ping www.google.com numeric.
PING www.google.com (216.58.196.100) 56(84) bytes of data.
64 bytes from 216.58.196.100: icmp_req=1 ttl=54 time=42.1 ms
64 bytes from 216.58.196.100: icmp_req=2 ttl=54 time=44.3 ms
^C
--- www.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 42.177/43.278/44.379/1.101 ms
```

ping <host> pattern <hexadecimal-digit> <option>

Specifies a hexadecimal digit pattern to fill the packet.

Syntax

ping { *ipv4_addres* | *ipv6_address* | *hostname* } **pattern** *hexadecimal-digit option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

hexadecimal-digit

Hexadecimal digit to fill the packet.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies the interface that the device must use as source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to send a packet that contains all 1's. This example helps to diagnose data-dependent problems in a network.

```
vyatta@vyatta:~$ ping www.google.com pattern BCD
PATTERN: 0xbc0d
PING www.google.com (173.194.33.176) 56(84) bytes of data.
64 bytes from sea09s18-in-f16.1e100.net (173.194.33.176): icmp_req=1 ttl=54 time=20.4 ms
64 bytes from sea09s18-in-f16.1e100.net (173.194.33.176): icmp_req=3 ttl=54 time=20.3 ms
From 10.18.170.201 icmp_seq=2 Destination Host Unreachable
From 10.18.170.201 icmp_seq=4 Destination Host Unreachable

--- www.google.com ping statistics ---
205 packets transmitted, 103 received, +102 errors, 49% packet loss, time 204233ms
rtt min/avg/max/mdev = 20.095/20.417/20.861/0.235 ms, pipe 4
vyatta@vyatta:~$ ^C
vyatta@vyatta:~$
```

ping <host> quiet <option>

Prints only the ping summary page.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **quiet** *option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to print only the ping summary page for www.google.com.

```
vyatta@vyatta:~$ ping www.google.com quiet
--- www.google.com ping statistics ---
15 packets transmitted, 1 received, +14 errors, 93% packet loss, time 41999ms
rtt min/avg/max/mdev = 20.264/20.264/20.264/0.000 ms
```

ping <host> record-route <option>

Records the route that a packet takes.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **record-route** *option*

Parameters

ipv4_address

Pings the IPv4 address of the host.

ipv6_address

Pings the IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound on every ping, while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

ping <host> size <bytes> <option>

Specifies the number of bytes to send for a ping request.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **size** *bytes option*

Parameters

ipv4_address

Pings the IPv4 address of the host.

ipv6_address

Pings the IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

bytes

The number of bytes to send for a ping request.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound during every ping, while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to specify the number of bytes to send while testing the IP address for network reachability.

```
vyatta@VR-1:~$ ping  2012:dead::1 size 1200
PING 2012:dead::1(2012:dead::1) 1200 data bytes
1208 bytes from 2012:dead::1: icmp_seq=1 ttl=64 time=0.046 ms
1208 bytes from 2012:dead::1: icmp_seq=2 ttl=64 time=0.128 ms
^C
--- 2012:dead::1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.046/0.087/0.128/0.041 ms
vyatta@VR-1:~$ ping  2012:dead::1
PING 2012:dead::1(2012:dead::1) 56 data bytes
64 bytes from 2012:dead::1: icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from 2012:dead::1: icmp_seq=2 ttl=64 time=0.034 ms
^C
```

ping <host> tos <number> <option>

Marks a packet with a specified time of service (TOS).

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **tos** *number option*

Parameters

ipv4_address

Pings the IPv4 address of the host.

ipv6_address

Pings the IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

number

The tos number.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound during every ping, while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to test a packet for network reachability by defining the time of service as one second.

```
vyatta@VR-1:~$ ping 127.0.0.1 tos 1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.409 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.027 ms
^C
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.027/0.218/0.409/0.191 ms
```

ping <host> ttl <seconds> <option>

Specifies the maximum packet life-time for a host.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **ttl** *seconds option*

Parameters

ipv4_address

The IPv4 address of the host.

ipv6_address

The IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

seconds

The time in milliseconds to specify the maximum packet life-time for a host.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound during every ping, while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer 3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery
Specifies the path MTU discovery strategy.

no-loopback
Supress loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

Examples

This example shows how to specify the maximum packet life-time for a host while testing the host for network reachability. The life-time defined in this example is 200 seconds.

```
vyatta@VR-1:~$ ping 192.1.2.2 ttl 200
PING 192.1.2.2 (192.1.2.2) 56(84) bytes of data.
64 bytes from 192.1.2.2: icmp_req=1 ttl=64 time=1.41 ms
^C
--- 192.1.2.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.412/1.412/1.412/0.000 ms
```

ping <host> verbose <option>

Displays a detailed output for the ping command.

Syntax

ping { *ipv4_address* | *ipv6_address* | *hostname* } **verbose** *option*

Parameters

ipv4_address

Pings the IPv4 address of the host.

ipv6_address

Pings the IPv6 address of the host.

hostname

A host being pinged. This keyword is used when the host is specified as a host name rather than as an IP address.

option

Each of the following entries are considered options. These options can be issued consecutively, that is, in the same command line.

adaptive

Adaptively sets interpacket interval.

allow-broadcast

Allows you to ping broadcast address.

audible

Makes a beep sound during every ping, while the device pings for the host details.

bypass-route

Bypasses normal routing tables during ping.

count

Specifies the number of ping requests to send.

deadline

Specifies the number of seconds before which ping expires.

ether-size

Matches the overall size of the data packet with the resultant size of the Layer-3 packet.

flood

Sends 100 ping requests each second.

interface

Specifies an interface that the device must use as the source address.

interval

Specifies the time in seconds for which the device must wait between ping requests.

mark

Specifies that the device must consider the ping request for special processing.

mtu-discovery

Specifies the path MTU discovery strategy.

no-loopback
Suppresses loop-back of multicast pings.

numeric
Does not resolve domain name system (DNS) names during ping.

pattern
Specifies the hexadecimal digit pattern to fill the packet.

quiet
Prints only the ping summary page.

record-route
Records the route that the packet takes.

size
Specifies the number of bytes to send for a ping request.

timestamp
Displays the timestamp during ping output.

tos
Marks packet with specified time of service (TOS).

ttl
Specifies the maximum packet life-time for a host.

verbose
Displays a detailed output for the ping command.

Modes

Operational mode

Usage Guidelines

The **ping** command tests whether a network host is reachable.

The **ping** command uses the ECHO_REQUEST datagram (ping) of the ICMP protocol to get an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams have an IP and an ICMP header, followed by a *struct timeval* data type and then an arbitrary number of pad bytes that are used to fill the packet.

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

protocols nsm log

Enables logging for NSM.

Syntax

set protocols nsm { **all** | **events** | **ha** | **kernel**| **packet** }

delete protocols nsm { **all** | **events** | **ha** | **kernel**| **packet** }

show protocols nsm { **all** | **events** | **ha** | **kernel**| **packet** }

Command Default

None

Parameters

all

Enables all NSM logs.

events

Enables only NSM event logs.

ha

Enables only NSM high availability (HA) logs.

kernel

Enables only NSM kernel logs.

packet

Enables only NSM packet logs.

Modes

Configuration mode

Configuration Statement

```
protocols {
     nsm {
      log {
          all
       events
           ha
       kernel
       packet
           }
        }
    }
```

Usage Guidelines

Use the **set** form of this command to enable NSM logs.

Use the **delete** form of this command to remove NSM logs.

Use the **show** form of this command to view NSM logs.

protocols nsm log ha

Enables logging for NSM HA.

Syntax

set protocols nsm ha { **all** }

delete protocols nsm ha { **all** }

show protocols nsm ha { **all** }

Command Default

None

Parameters

all

Enables all NSM HA logs

Modes

Configuration mode

Configuration Statement

```
protocols {
     nsm {
      log {
        ha {
           all
            }
         }
      }
 }
```

Usage Guidelines

Use the **set** form of this command to enable NSM high availability (HA) logs.

Use the **delete** form of this command to remove NSM HA logs.

Use the **show** form of this command to view NSM HA logs.

reset ip route kernel

Clears all the entries from the IP kernel route.

Syntax

reset ip route kernel

Modes

Operational mode

Usage Guidelines

Use this command to clear the entries from the IP kernel route.

reset ipv6 route kernel

Clears all the entries from the IPv6 kernel route.

Syntax

reset ipv6 route kernel

Modes

Operational mode

Usage Guidelines

Use this command to clear the entries from the IPv6 kernel route.

resources group address-group <group-name>

Defines a group of IP addresses that are referenced in firewall rules.

Syntax

set resources group address-group *group-name* { **address** *address* | **description** *desc* }

delete resources group address-group *group-name* { **address** *address* | **description** *desc* }

show resources group address-group *group-name* { **address** *address* | **description** *desc* }

Parameters

address-group

A group of IPv4 addresses or address ranges.

group-name

Mandatory. The name of a firewall address group.

address *address*

Mandatory. Adds the specified IPv4 address or range of IPv4 addresses to the specified firewall address group. IPv4 address ranges are specified by separating two contiguous IPv4 addresses with a hyphen, for example, 10.0.0.1-10.0.0.50.

description *desc*

Provides a brief description for the firewall address group.

Modes

Configuration mode

Configuration Statement

```
resources {
    group {
        address-group group-name {
            address address
            description desc
        }
          }
}
```

Usage Guidelines

Use this command to specify an address group. A firewall address group is a collection of host IP addresses and address ranges that, once defined, can be collectively referenced within a firewall command.

A firewall address group is considered matched if the packet address matches any address or address range within the group.

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

Use the **delete** form of this command to remove a firewall address group or its members.

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

resources group icmp-group <group-name>

Defines a group of ICMP types that may be referenced in firewall rules, policy-based routing rules or QoS rules.

Syntax

set resources group icmp-group *group-name* { **description** *description* | **name** *name* | **type** *number* [**code** *number*] }

delete resources group icmp-group *group-name* [**description** *description* | **name** *name* | **type** *number* [**code** *number*]]

show resources group icmp-group *group-name* [**description** *description* | **name** *name* | **type** *number* [**code** *number*]]

Parameters

group-name
Name of an IPv4 ICMP group.

description *description*
Describes an IPv4 ICMP group.

name *name*
Specifies the name of an ICMP type.

type *number*
Specifies the numeric identifier of an IPv4 ICMP type. The numeric identifier ranges from 0 through 255.

code *number*
Specifies the numeric identifier of an IPv4 ICMP code. The numeric identifier ranges from 0 through 255.

Modes

Configuration mode

Configuration Statement

```
resources {
    group {
        icmp-group group-name {
            description description
            name name
            type number {
                code number
            }
        }
    }
}
```

Usage Guidelines

Use this command to define an IPv4 Internet Control Message Protocol (ICMP) group. An ICMP group is a collection of ICMP types that, once defined, can be collectively referenced in firewall rules, policy-based routing rules and Quality of Service (QoS) rules.

An ICMP group is considered matched if any ICMP type in the group is matched.

Use the **set** form of this command to define an ICMP group.

Use the **delete** form of this command to remove an ICMP group or its members.

Use the **show** form of this command to display an ICMP group or its members.

resources group icmpv6-group <group-name>

Defines a group of ICMPv6 types that may be referenced in firewall rules, policy-based routing rules or QoS rules.

Syntax

set resources group icmpv6-group *group-name* { **description** *description* | **name** *name* | **type** *number* [**code** *number*] }

delete resources group icmpv6-group *group-name* [**description** *description* | **name** *name* | **type** *number* [**code** *number*]]

show resources group icmpv6-group *group-name* [**description** *description* | **name** *name* | **type** *number* [**code** *number*]]

Parameters

group-name
Name of an ICMPv6 group.

description *description*
Describes an ICMPv6 group.

name *name*
Specifies the name of an ICMPv6 type.

type *number*
Specifies the numeric identifier of an ICMPv6 type. The numeric identifier ranges from 0 through 255.

code *number*
Specifies the numeric identifier of an ICMPv6 code. The numeric identifier ranges from 0 through 255.

Modes

Configuration mode

Configuration Statement

```
resources {
    group {
        icmpv6-group group-name {
            description description
            name name
            type number {
                code number
            }
        }
    }
}
```

Usage Guidelines

Use this command to define an Internet Control Message Protocol Version 6 (ICMPv6) group. An ICMPv6 group is a collection of ICMPv6 types that, once defined, can be collectively referenced in firewall rules, policy-based routing rules and Quality of Service (QoS) rules.

An ICMPv6 group is considered matched if any ICMPv6 type in the group is matched.

Use the **set** form of this command to define an ICMPv6 group.

Use the **delete** form of this command to remove an ICMPv6 group or its members.

Use the **show** form of this command to display an ICMPv6 group or its members.

resources group port-group <group-name>

Defines a group of ports that are referenced in firewall rules.

Syntax

set resources group port-group *port-group-name* { **description** *description* { **port** [*name* | *1-65535* | *start* **-** *end*] }

delete resources group port-group *port-group-name* { **description** *description* | **port** [*name* | *1-65535* | *start* **-** *end*] }

show resources group port-group *port-group-name* { **description** *description* | **port** *name* | *1-65535* | *start* **-** *end* **test**] }

Parameters

port-group *port-group-name*

Matches the destination port packets against the specified port group. The packet is considered a match if it matches any port name or number specified in the group. Only one port group may be specified. The port group must already be defined. A packet is considered a match for an address, a network, or a port group if it matches any host IP address, network address, or port name or number, respectively, in the group. However, if more than one group is specified, the packet must be a match for both groups in order to be considered a match. For example, if an address group and a port group are both specified, the packet's destination must match at least one item in the address group and at least one item in the port group.

description *description*

Provides a brief description for the network group.

port [*name* | *1-65535* | *start* **-** *end*]

Specifies the port group parameters.

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 through 65535.

start-end

Matches the specified range of ports; for example, 1001-1005.

Modes

Configuration mode

Configuration Statement

```
resources {
    group {
        port-group group-name {
            port name
            description desc
        }
    }
}
```

Usage Guidelines

Use this command to define a network group. A network group is a collection of network addresses that, once defined, can be collectively referenced within a firewall command.

A network group is considered matched if the packet address matches any network address or address range within the group.

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

Use the **delete** form of this command to remove a network group or its members.

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

show ip forwarding

Displays IP forwarding status.

Syntax

show ip forwarding

Modes

Operational mode

Usage Guidelines

Use this command to display IP forwarding status.

Examples

The following example shows how to display the status of IP forwarding.

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

show ip route

Displays routes stored in the Routing Information Base (RIB) and Forwarding Information Base (FIB).

Syntax

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

Command Default

Lists all routes stored in the RIB and FIB.

Parameters

ipv4

Optional. An IP address.

ipv4net

Optional. A prefix.

Modes

Operational mode

Usage Guidelines

Use this command to display routes stored in the RIB and FIB.

You can also see the routes shown in the FIB by using show ip route forward on page 90.

Examples

The following example shows how to display routes stored 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, dp0p0p0
O   10.1.0.0/24 [110/10] is directly connected, dp0p0p0, 05:35:15
C>* 10.1.0.0/24 is directly connected, dp0p0p0
O>* 10.192.32.0/24 [110/20] via 10.1.0.45, dp0p0p0, 05:35:15
O>* 10.192.128.0/24 [110/11] via 10.1.0.66, dp0p0p0, 05:35:15
O>* 10.192.128.1/32 [110/11] via 10.1.0.66, dp0p0p0, 05:35:15
O>* 10.192.129.0/24 [110/11] via 10.1.0.66, dp0p0p0, 05:35:15
O>* 10.192.130.0/24 [110/11] via 10.1.0.66, dp0p0p0, 05:35:15
O>* 10.192.131.0/24 [110/11] via 10.1.0.66, dp0p0p0, 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, dp0p0p0, 05:35:15
O>* 172.16.1.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.2.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.3.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.4.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.5.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.6.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.7.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.8.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
O>* 172.16.9.0/24 [110/11] via 10.1.0.4, dp0p0p0, 05:35:15
C>* 172.16.234.0/25 is directly connected, dp0p0p1
S>* 192.94.202.0/24 [1/0] via 172.16.234.27, dp0p0p1
```

The following example shows how to display information for the route to the 10.192.128.1 IP 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 dp0p0p0
vyatta@vyatta:~$
```

show ip route <ipv4net> longer-prefixes

Displays prefixes in the Routing Information Base (RIB) that are longer than a specific IP address or prefix.

Syntax

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

Parameters

ipv4net

Mandatory. An IP address or prefix.

Modes

Operational mode

Usage Guidelines

Use this command to display all prefixes in the RIB that are longer than a specific IP address or prefix.

Examples

The following example shows how to display prefixes that are longer than the 10.192.128.0/24 prefix.

```
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, dp0p0p0, 09:36:20
O>* 10.192.128.1/32 [110/11] via 10.1.0.66, dp0p0p0, 09:36:20
vyatta@vyatta:~$
```

show ip route connected

Displays directly connected routes.

Syntax

show ip route connected

Modes

Operational mode

Usage Guidelines

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

Examples

The following example shows how to display directly 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, dp0p0p0
C>* 127.0.0.0/8 is directly connected, lo
C>* 172.16.234.0/25 is directly connected, dp0p0p1
vyatta@vyatta:~$
```

show ip route forward

Displays routes stored in the Forwarding Information Base (FIB).

Syntax

show ip route forward [*ipv4net*]

Command Default

Displays routes stored in the FIB.

Parameters

ipv4net

Optional. A route for which information from the kernel forwarding table is displayed.

Modes

Operational mode

Usage Guidelines

Use this command to display routes that are stored in the FIB.

The FIB contains multiple equal-cost paths, if they exist. Multiple equal-cost paths are needed before equal-cost multipath (ECMP) routing or WAN load balancing is performed.

Examples

The following example shows how to display routes stored in the FIB.

```
vyatta@vyatta:~$ show ip route forward
default via 10.1.0.1 dev dp0p0p0  proto zebra
10.1.0.0/24 dev dp0p0p0  proto kernel  scope link  src 10.1.0.62
10.192.32.0/24 via 10.1.0.45 dev dp0p0p0  proto zebra  metric 20
10.192.128.0/24 via 10.1.0.66 dev dp0p0p0  proto zebra  metric 11
10.192.128.1 via 10.1.0.66 dev dp0p0p0  proto zebra  metric 11
10.192.129.0/24 via 10.1.0.66 dev dp0p0p0  proto zebra  metric 11
10.192.130.0/24 via 10.1.0.66 dev dp0p0p0  proto zebra  metric 11
10.192.131.0/24 via 10.1.0.66 dev dp0p0p0  proto zebra  metric 11
172.16.0.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.1.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.2.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.3.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.4.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.5.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.6.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.7.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.8.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.9.0/24 via 10.1.0.4 dev dp0p0p0  proto zebra  metric 11
172.16.234.0/25 dev dp0p0p1  proto kernel  scope link  src 172.16.234.23
192.94.202.0/24 via 172.16.234.27 dev dp0p0p1  proto zebra
vyatta@vyatta:~$
```

The following example shows how to display information from the FIB about the 10.1.0.0/24 route.

```
vyatta@vyatta:~$ show ip route forward 10.1.0.0/24
10.1.0.0/24 dev dp0p0p0  proto kernel  scope link  src 10.1.0.62
vyatta@vyatta:~$
```

show ip route kernel

Displays kernel routes.

Syntax

show ip route kernel

Modes

Operational mode

Usage Guidelines

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

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

Examples

The following example shows how to display 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, dp0p0p0
vyatta@vyatta:~$
```

show ip route static

Displays static routes in the Routing Information Base (RIB).

Syntax

show ip route static

Modes

Operational mode

Usage Guidelines

Use this command to display static routes in the RIB.

Examples

The following example shows how to display static routes in the RIB.

```
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, dp0p0p0
S>* 192.94.202.0/24 [1/0] via 172.16.234.27, dp0p0p1
vyatta@vyatta:~$
```

show ip route summary

Displays a summary of routes.

Syntax

show ip route summary

Modes

Operational mode

Usage Guidelines

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

Examples

The following example shows how to display 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

Modes

Operational mode

Usage Guidelines

Use this command to display supernet routes.

Supernet routes are routes that have a subnet mask that is less specific than the usual classful mask.

Examples

The following example shows how to display 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, dp0p0p0
vyatta@vyatta:~$
```

show ip route table <table>

Displays routes stored in an alternate routing table.

Syntax

show ip route table *table-number*

Parameters

table *table-number*

An alternate routing table.

Modes

Operational mode

Usage Guidelines

Use this command to view routes stored in an alternate routing table. Alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

Examples

The following example shows how to display routes in the 5 alternate routing table.

```
vyatta@vyatta:~$ show ip route table 5
table 5:
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route
S>* 12.34.56.0/24 [1/0] via 192.168.1.254, dp0p0p0
vyatta@vyatta:~$
```

show ip route variance

Detects the routes that are missing from the RIB, kernel, and data plane table and stores the discrepancy in a file.

Syntax

show ip route variance

Modes

Operational mode

Usage Guidelines

Detects the routes that are missing from the RIB, kernel, and data plane table and stores the discrepancy in the following file: `/home/<username>/vyatta_rtvariance.output`. A missing route is identified by using codes such as R for RIB, K for kernel, and D for data plane. This command verifies only active routes and interfaces and can help during debugging.

NOTE

Brocade recommends that you use the command in a stable environment to ensure that you do not get wrong results. The vRouter may take more time to generate the output if the system has millions of routes.

Examples

The following example shows how to store the discrepancy in the following file: `/home/vyatta/vyatta_rtvariance.output`. The following output indicates that an IPv4 route is missing from the kernel.

```
vyatta@vyatta# show ip route variance
Output is dumped in the file: /home/vyatta/vyatta_rtvariance.output
```

show ip route variance console

Detects the routes that are missing from the RIB, kernel, and data plane table and displays the discrepancy at the console.

Syntax

show ip route variance console

Modes

Operational mode

Usage Guidelines

Compares the routes in the RIB, kernel, and data plane table, detects missing routes, and displays the discrepancy at the console . The discrepancy contains IPv4 routes and addresses that are missing from the RIB, kernel, and data plane table. A missing route is identified by using codes such as R for RIB, K for kernel, and D for data plane. This command verifies only active routes and interfaces and can help during debugging.

NOTE

Brocade recommends that you use the command in a stable environment to ensure that you do not get wrong results. The vRouter may take more time to generate the output if the system has millions of routes.

Examples

The following example shows how to display the discrepancy information at the console. The output displays the following information:

- The interface variance table indicates that the **88.88.88.4** address which is configured on the **lo4** interface is missing from the data plane.
- The route variance table indicates that the **200.9.9.0/32** address route with the **12.12.12.20** nexthop is missing from both the kernel and data plane.

```
vyatta@vyatta#show ip route variance console
Codes: R - RIB, K - Kernel, D - Dataplane
(Indicates the table Id in which the address/route is missing)

Interface Variance Table:
D     88.88.88.4 lo4 (present in Kernel)
K     201.202.203.0 dp0s8 (present in RIB)
K     6.6.6.0 dp0s7.1 (present in RIB)
R     12.12.12.0 dp0s7 (present in Kernel)

Route Variance Table:
Table Absence  Route
def    D        100.4.4.4/32 via 12.12.12.18
def     K      100.1.1.1/32 via 12.12.12.19
def     KD      200.9.9.0/24 via 12.12.12.20
def     R      100.1.1.1/32 via 12.12.12.17
def     R      200.1.1.1/32 via 12.12.12.20
def     RK     100.4.4.4/32 via 12.12.12.28
def     RK      200.20.20.20/32 via 12.12.12.30
```

Examples

The following example shows how to display the discrepancy at the console. The output indicates that all tables are synchronized and that no routes and interfaces are missing.

```
vyatta@vyatta#show ip route variance console

No Variance, Inteface addresses and Routes are in sync
```

show ipv6 route

Displays IPv6 routes stored in the Routing Information Base (RIB) and Forwarding Information Base (FIB).

Syntax

show ipv6 route [*ipv6* | *ipv6net*]

Command Default

Displays all IPv6 routes in the RIB and FIB.

Parameters

ipv6

Optional. An IPv6 address.

ipv6net

Optional. An IPv6 prefix.

Modes

Operational mode

Usage Guidelines

Use this command to display IPv6 routes stored in the RIB and FIB.

You can also see the routes shown in the FIB by using show ip route forward on page 90.

show ipv6 route <ipv6net> longer-prefixes

Displays IPv6 prefixes in the Routing Information Base (RIB) that are longer than a specific IPv6 address or prefix.

Syntax

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

Parameters

ipv6net

Mandatory. An IPv6 address or prefix.

Modes

Operational mode

Usage Guidelines

Use this command to display all prefixes in the RIB that are longer than a specific IPv6 address or prefix.

show ipv6 route bgp

Displays IPv6 Border Gateway Protocol (BGP) routes.

Syntax

show ipv6 route bgp

Modes

Operational mode

Usage Guidelines

Use this command to display IPv6 BGP routes.

show ipv6 route connected

Displays IPv6 connected routes.

Syntax

show ipv6 route connected

Modes

Operational mode

Usage Guidelines

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

show ipv6 route forward

Displays IPv6 routes stored in the Forwarding Information Base (FIB).

Syntax

show ipv6 route forward [*ipv6net*]

Command Default

Displays IPv6 routes stored in the FIB.

Parameters

ipv6net

Optional. An IPv6 route for which information from the kernel forwarding table is displayed.

Modes

Operational mode

Usage Guidelines

Use this command to display routes that are stored in the FIB.

The FIB contains multiple equal-cost paths, if they exist. Multiple equal-cost paths are needed before equal-cost multipath (ECMP) routing or WAN load balancing is performed.

show ipv6 route kernel

Displays IPv6 kernel routes.

Syntax

show ipv6 route kernel

Modes

Operational mode

Usage Guidelines

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

show ipv6 route ripng

Displays IPv6 Routing Information Protocol next generation (RIPng) routes.

Syntax

show ipv6 route ripng

Modes

Operational mode

Usage Guidelines

Use this command to display IPv6 RIPng routes.

show ipv6 route static

Displays IPv6 static routes.

Syntax

show ipv6 route static

Modes

Operational mode

Usage Guidelines

Use this command to display IPv6 static routes.

show ipv6 route variance

Detects the IPv6 routes that are missing from the RIB, kernel, and data plane table and stores the discrepancy in a file.

Syntax

show ipv6 route variance

Modes

Operational mode

Usage Guidelines

Compares the routes in the RIB, kernel, and data plane table, detects missing IPv6 routes, and stores the discrepancy in the following file: `/home/<username>/vyatta_rtvariance.output`. The discrepancy contains IPv6 routes and addresses that are missing from the RIB, kernel, or data plane table. A missing route is identified by using codes such as R for RIB, K for kernel, and D for data plane. This command verifies only active routes and interfaces and can help during debugging.

NOTE

Brocade recommends that you use the command in a stable environment to ensure that you do not get wrong results. The vRouter may take more time to generate the output if the system has millions of routes.

Examples

The following example shows how to store the discrepancy in the following file: `/home/<username>/vyatta_rtvariance.output`.

```
vyatta@vyatta# show ipv6 route variance
Output is dumped in the file: /home/vyatta/vyatta_rtvariance.output
```

show ipv6 route variance console

Detects IPv6 routes that are missing from the RIB, kernel, and data plane table and displays the discrepancy at the console.

Syntax

show ipv6 route variance console

Modes

Operational mode

Usage Guidelines

Detects the IPv6 routes that are missing from the RIB, kernel, and data plane tables and displays the discrepancy at the console. The discrepancy contains IPv6 routes and addresses that are missing from the RIB, kernel, and data plane table. A missing route is identified by using codes such as R for RIB, K for kernel, and D for data plane. This command verifies only active routes and interfaces and can help during debugging.

NOTE

Brocade recommends that you use the command in a stable environment to ensure that you do not get wrong results. The vRouter may take more time to generate the output if the system has millions of routes.

Examples

The following example shows how to display the discrepancy at the console. The following output indicates that **2626:1111:2222:3333:4444:5555:8888:1** address is missing from the data plane. The route variance table indicates that all tables are synchronized and that no IPv6 route is missing.

```
vyatta@vyatta#show ipv6 route variance console

Codes: R - RIB, K - Kernel, D - Dataplane (Indicates the table Id in which the address/route is missing)

Interface Variance Table:

     D    2626:1111:2222:3333:4444:5555:8888:1 dp0s3 (present in Kernel)

Route Variance Table:
Routes are in sync
```

show monitoring protocols rib

Displays Routing Information Base (RIB) debugging flags.

Syntax

show monitoring protocols rib

Modes

Operational mode

Usage Guidelines

Use this command to display RIB debugging flags.

traceroute <host> as-path

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **as-path** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

options

The following entries are options. Multiple options can be included on the same command line.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com including AS path information.

```
vyatta@vyatta#traceroute google.com as-path
traceroute to google.com (216.58.192.14), 30 hops max, 60 byte packets
 1  10.18.170.1 (10.18.170.1) [*]  0.681 ms  0.529 ms  0.580 ms
 2  10.31.23.6 (10.31.23.6) [*]  0.476 ms  0.433 ms  0.481 ms
 3  10.254.33.1 (10.254.33.1) [*]  0.864 ms  0.831 ms  0.826 ms
 4  144.49.130.145 (144.49.130.145) [AS29791/AS21948]  8.022 ms  1.183 ms  1.295 ms
 5  ae6-395.edge8.sanjose1.level3.net (209.244.104.65) [AS3356]  2.106 ms  2.046 ms  2.006 ms
 6  ae-1-60.edge1.sanjose3.level3.net (4.69.152.16) [AS3356]  2.887 ms * *
 7  72.14.223.91 (72.14.223.91) [AS15169]  2.878 ms  2.972 ms  2.938 ms
 8  209.85.249.3 (209.85.249.3) [AS15169]  3.430 ms  4.754 ms  3.460 ms
 9  74.125.37.41 (74.125.37.41) [AS15169]  4.568 ms  4.516 ms  4.455 ms
10  nuq04s29-in-f14.1e100.net (216.58.192.14) [AS15169]  3.159 ms  3.134 ms  3.097 ms
```

traceroute <host> bypass-routing

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **bypass-routing** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com bypassing the normal routing tables. Note that an error message appears because google.com is not on a directly attached network.

```
vyatta@vyatta#traceroute google.com bypass-routing
traceroute to google.com (216.58.192.14), 30 hops max, 60 byte packets
connect:network is unreachable
```

traceroute <host> debug-socket

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **debug-socket** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

debug-socket

Enables socket level debugging.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with socket level debugging enabled.

```
vyatta@vyatta#traceroute google.com debug-socket
traceroute to google.com (216.58.216.14), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.422 ms  0.399 ms  0.498 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  16.520 ms  6.484 ms  6.460 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  33.529 ms  36.814 ms  26.584 ms
4  71.145.0.192 (71.145.0.192)  26.546 ms  39.056 ms  38.523 ms
5  12.83.39.189 (12.83.39.189)  39.763 ms  30.819 ms  44.405 ms
6  12.122.136.181 (12.122.136.181)  44.407 ms  43.366 ms  43.334 ms
7  * * *
8  216.239.49.170 (216.239.49.170)  28.635 ms 216.239.49.168 (216.239.49.168)  28.293 ms
                                                    216.239.49.170 (216.239.49.170)  25.805 ms
9  209.85.246.253 (209.85.246.253)  32.914 ms  34.112 ms 209.85.246.20 (209.85.246.20)  30.330 ms
10  64.233.174.204 (64.233.174.204)  36.979 ms  36.492 ms  38.584 ms
11  64.233.175.151 (64.233.175.151)  37.497 ms  37.496 ms 64.233.174.189 (64.233.174.189)  37.503 ms
12  209.85.142.91 (209.85.142.91)  36.126 ms  36.735 ms  36.686 ms
13  lax02s21-in-f14.1e100.net (216.58.216.14)  34.280 ms  32.453 ms  31.764 ms
```

traceroute <host> first-ttl <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **first-ttl** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with the first time-to-live set to 3.

```
vyatta@vyatta#traceroute google.com first-ttl 3
traceroute to google.com (74.125.224.7), 30 hops max, 60 byte packets
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  36.929 ms  38.025 ms  38.016 ms
4  71.145.0.192 (71.145.0.192)  37.998 ms  41.153 ms  40.586 ms
5  12.83.39.189 (12.83.39.189)  43.315 ms 12.83.39.185 (12.83.39.185)  39.053 ms
                                                    12.83.39.189 (12.83.39.189)  43.311 ms
6  12.122.136.181 (12.122.136.181)  69.120 ms  69.564 ms  69.086 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  49.860 ms  53.125 ms  39.522 ms
9  72.14.232.33 (72.14.232.33)  35.172 ms  34.588 ms  35.129 ms
10  nuq04s18-in-f7.1e100.net (74.125.224.7)  37.639 ms  36.897 ms  32.858 ms
```

traceroute <host> gateway <address>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **gateway** *address* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

gateway *address*

Routes the request through a specified gateway.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with the gateway set.

```
vyatta@vyatta#traceroute google.com gateway
```

traceroute <host> icmp-echo

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **icmp-echo** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

icmp-echo

Uses ICMP echo for the traceroute probe.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com using ICMP echo for the traceroute probe.

```
vyatta@vyatta#traceroute google.com icmp-echo
traceroute to google.com (74.125.224.9), 30 hops max, 60 byte packets
 1  gateway.attlocal.net (10.0.6.1)  0.512 ms  0.510 ms  0.507 ms
 2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  3.175 ms  3.194 ms  3.194 ms
 3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  24.351 ms  25.192 ms  25.201 ms
 4  71.145.0.192 (71.145.0.192)  26.644 ms  27.905 ms  27.910 ms
 5  12.83.39.185 (12.83.39.185)  28.728 ms  32.328 ms  32.335 ms
 6  12.122.136.181 (12.122.136.181)  68.186 ms  67.898 ms  67.853 ms
 7  * * *
 8  216.239.49.168 (216.239.49.168)  25.336 ms  26.631 ms  27.460 ms
 9  72.14.232.33 (72.14.232.33)  26.151 ms  26.620 ms  27.186 ms
10  nuq04s18-in-f9.1e100.net (74.125.224.9)  27.531 ms  24.783 ms  24.752 ms
```

traceroute <host> icmp-extensions

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **icmp-extensions** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com showing ICMP extensions.

```
vyatta@vyatta#traceroute google.com icmp-extensions
traceroute to google.com (74.125.224.1), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.482 ms  0.458 ms  0.452 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  13.714 ms  13.703 ms  13.673 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  35.518 ms  34.284 ms  35.492 ms
4  71.145.0.192 (71.145.0.192)  34.201 ms  35.828 ms  35.385 ms
5  12.83.39.189 (12.83.39.189)  39.513 ms 12.83.39.185 (12.83.39.185)  39.510 ms 12.83.39.189
(12.83.39.189)  44.236 ms
6  12.122.136.181 (12.122.136.181)  47.009 ms  46.105 ms  46.052 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  31.725 ms  28.023 ms  28.467 ms
9  72.14.232.33 (72.14.232.33)  32.480 ms  31.081 ms  31.791 ms
10  nuq04s18-in-f1.1e100.net (74.125.224.1)  32.791 ms  26.412 ms  25.713 ms
```

traceroute <host> interface <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **interface** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

interface *value*

Specifies the interface that the device must use for traceroute requests.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com through interface dp0p1s2.

```
vyatta@vyatta#traceroute google.com interface dp0p1s2
```

traceroute <host> max-ttl <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **max-ttl** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

max-ttl *value*

Specifies the maximum number of hops for the probe.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with a maximum of 4 hops.

```
vyatta@vyatta#traceroute google.com max-ttl 4
traceroute to google.com (74.125.224.8), 4 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.362 ms  0.333 ms  0.340 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  9.559 ms  9.553 ms  9.529 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  30.061 ms  31.737 ms  31.773 ms
4  71.145.0.192 (71.145.0.192)  31.776 ms  31.763 ms  31.758 ms
```

traceroute <host> interval <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **interval** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

interval *value*

Specifies the time in seconds between traceroute requests from the device.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

max-ttl *value*

Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with 3 seconds between traceroute requests.

```
vyatta@vyatta#traceroute google.com interval 3
```

traceroute <host> max-ttl <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **max-ttl** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

max-ttl *value*

Specifies the maximum number of hops for the probe.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with a maximum of 4 hops.

```
vyatta@vyatta#traceroute google.com max-ttl 4
traceroute to google.com (74.125.224.8), 4 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.362 ms  0.333 ms  0.340 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  9.559 ms  9.553 ms  9.529 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  30.061 ms  31.737 ms  31.773 ms
4  71.145.0.192 (71.145.0.192)  31.776 ms  31.763 ms  31.758 ms
```

traceroute <host> no-fragment

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **no-fragment** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

no-fragment

Does not fragment the probe packets.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with no fragmented probe packets.

```
vyatta@vyatta#traceroute google.com no-fragment
traceroute to google.com (74.125.224.0), 30 hops max, 60 byte packets
1  pipsqueak.attlocal.net (10.0.6.1)  0.394 ms  0.564 ms  0.356 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  9.799 ms  9.771 ms  9.748 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  31.504 ms  30.107 ms  36.088 ms
4  71.145.0.192 (71.145.0.192)  36.139 ms  31.473 ms  31.441 ms
5  12.83.39.189 (12.83.39.189)  38.391 ms  37.441 ms 12.83.39.185 (12.83.39.185)  36.589 ms
6  12.122.136.181 (12.122.136.181)  83.786 ms  82.908 ms  82.880 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  27.049 ms  36.183 ms  33.859 ms
9  72.14.232.33 (72.14.232.33)  33.652 ms  33.064 ms  33.645 ms
10  nuq04s18-in-f0.1e100.net (74.125.224.0)  36.182 ms  36.165 ms  36.142 ms
```

traceroute <host> num-queries <num>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **num-queries** *number* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

num-queries *number*

Specifies the number of probe packets per hop. The default is 3.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with 4 probes per hop.

```
vyatta@vyatta#traceroute google.com num-queries 2
traceroute to google.com (74.125.224.9), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.411 ms  0.387 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  14.167 ms  14.145 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  36.338 ms  35.080 ms
4  71.145.0.192 (71.145.0.192)  36.862 ms  36.338 ms
5  12.83.39.189 (12.83.39.189)  49.387 ms 12.83.39.185 (12.83.39.185)  49.374 ms
6  12.122.136.181 (12.122.136.181)  41.428 ms  41.419 ms
7  * *
8  216.239.49.168 (216.239.49.168)  41.356 ms  49.198 ms
9  72.14.232.33 (72.14.232.33)  39.738 ms  39.720 ms
10  nuq04s18-in-f9.1e100.net (74.125.224.9)  34.504 ms  34.474 ms
```

traceroute <host> port <number>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **port** *number* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

port *number*

For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com through port 80.

```
vyatta@vyatta#traceroute google.com port 80
traceroute to google.com (74.125.224.1), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.383 ms  0.337 ms  0.327 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  1.689 ms  1.582 ms  1.461 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  22.674 ms  21.656 ms  21.901 ms
4  71.145.0.192 (71.145.0.192)  26.552 ms  21.609 ms  21.732 ms
5  12.83.39.185 (12.83.39.185)  23.492 ms 12.83.39.189 (12.83.39.189)  24.755 ms 12.83.39.185
(12.83.39.185) 23.333 ms
6  12.122.136.181 (12.122.136.181)  23.033 ms  22.736 ms  23.119 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  25.157 ms  24.775 ms  24.790 ms
9  72.14.232.33 (72.14.232.33)  25.859 ms  25.051 ms  25.207 ms
10  nuq04s18-in-f1.1e100.net (74.125.224.1)  25.102 ms  24.958 ms  25.077 ms
```

traceroute <host> seq-queries <number>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **seq-queries** *number* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

seq-queries *number*

Specifies the number of sequential probe packets.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with 2 sequential probe packets.

```
vyatta@vyatta#traceroute google.com seq-queries 2
traceroute to google.com (74.125.224.0), 30 hops max, 60 byte packets
1  pipsqueak.attlocal.net (10.0.6.1)  0.441 ms  0.416 ms  0.357 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  1.762 ms  2.295 ms  1.497 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  23.357 ms  22.565 ms  22.219 ms
4  71.145.0.192 (71.145.0.192)  23.964 ms  22.780 ms  21.566 ms
5  12.83.39.185 (12.83.39.185)  27.362 ms 12.83.39.189 (12.83.39.189)  26.326 ms 12.83.39.185
(12.83.39.185) 25.436 ms
6  12.122.136.181 (12.122.136.181)  113.918 ms  88.183 ms  49.824 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  26.184 ms  24.964 ms  24.673 ms
9  72.14.232.33 (72.14.232.33)  25.629 ms  25.079 ms  25.069 ms
10  nuq04s18-in-f0.1e100.net (74.125.224.0)  25.164 ms  25.286 ms  24.878 ms
```

traceroute <host> source-addr <host>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **source-addr** *host* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

source-addr *host*

Specifies an alternative source host by hostname, IPv4 address, or MAC address.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with source address *client1.attlocal.net*.

```
vyatta@vyatta#traceroute google.com source-addr client.attlocal.net
```

traceroute <host> tcp-syn

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **tcp-syn** [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

tcp-syn

Uses TCP SYN for the probes.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com using tcp-syn for probes.

```
vyatta@vyatta#traceroute google.com tcp-syn
traceroute to google.com (74.125.224.6), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.389 ms  0.341 ms  0.316 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  12.784 ms  12.767 ms  12.771 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  35.539 ms  35.529 ms  35.517 ms
4  71.145.0.192 (71.145.0.192)  35.512 ms  35.503 ms  33.653 ms
5  12.83.39.189 (12.83.39.189)  38.888 ms  36.371 ms  37.952 ms
6  12.122.136.181 (12.122.136.181)  63.947 ms  63.087 ms  63.045 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  28.752 ms  29.378 ms  29.368 ms
9  * * *
10  nuq04s18-in-f6.1e100.net (74.125.224.6)  29.669 ms  27.836 ms  29.669 ms
```

traceroute <host> tos <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **tos** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

tos *value*

Marks the packets with the specified Type of Service (TOS) value.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

version
Displays the timestamp during ping output.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com with packets marked with tos equal to 3.

```
vyatta@vyatta#traceroute google.com tos 3
traceroute to google.com (74.125.224.2), 30 hops max, 60 byte packets
1  gateway.attlocal.net (10.0.6.1)  0.374 ms  0.550 ms  0.353 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  7.270 ms  15.975 ms  7.238 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  29.199 ms  38.969 ms  28.635 ms
4  71.145.0.192 (71.145.0.192)  35.803 ms  36.212 ms  28.590 ms
5  12.83.39.189 (12.83.39.189)  34.061 ms  32.033 ms  42.496 ms
6  12.122.136.181 (12.122.136.181)  39.973 ms  41.665 ms  41.608 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  29.549 ms  26.607 ms  26.583 ms
9  72.14.232.33 (72.14.232.33)  28.133 ms  27.602 ms  29.294 ms
10  nuq04s18-in-f2.1e100.net (74.125.224.2)  28.553 ms  30.071 ms  28.028 ms
```

traceroute <host> version

Displays the route that packets take to a network host.

Syntax

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

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

version

Displays the timestamp during ping output.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

wait-time *value*
Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

Examples

The following example illustrates a traceroute to google.com showing timestamp during ping output.

```
vyatta@vyatta#traceroute google.com version
```

traceroute <host> wait-time <value>

Displays the route that packets take to a network host.

Syntax

traceroute [**ipv4** | **ipv6**] *host* **wait-time** *value* [**options**]

Parameters

ipv4

Explicitly force IPv4 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv4.

ipv6

Explicitly force IPv6 traceouting. By default, the program will try to resolve the name given, and choose the appropriate protocol automatically. If resolving a host name returns both IPv4 and IPv6 addresses, **traceroute** will use IPv6.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format h:h:h:h:h:h:h:h).

wait-time *value*

Specifies the time (seconds) to wait for a response from the probe. Default is 5 seconds.

options

The following entries are options. Multiple options can be included on the same command line.

as-path

Performs AS path lookups in routing registries and print results directly after the corresponding addresses.

bypass-routing

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it.

debug-socket

Enables socket level debugging.

first-ttl *value*

Specifies the first time-to-live value. Defaults to 1.

gateway *address*

Routes the request through a specified gateway.

icmp-echo

Uses ICMP echo for the traceroute probe.

icmp-extensions

Shows ICMP extensions (rfc4884). The general form is *CLASS/TYPE:* followed by a hexadecimal dump.

interface *value*

Specifies the interface that the device must use for traceroute requests.

interval *value*

Specifies the time in seconds between traceroute requests from the device.

max-ttl *value*
Specifies the maximum number of hops for the probe.

no-fragment
Does not fragment the probe packets.

num-queries *number*
Specifies the number of probe packets per hop. The default is 3.

port *number*
For UDP tracing, specifies the destination port base traceroute will use (the destination port number will be incremented by each probe). For ICMP tracing, specifies the initial icmp sequence value (incremented by each probe too). For TCP specifies just the (constant) destination port to connect.

seq-queries *number*
Specifies the number of sequential probe packets.

source-addr *host*
Specifies an alternative source host by hostname, IPv4 address, or MAC address.

tcp-syn
Uses TCP SYN for the probes.

tos *value*
Marks the packets with the specified Type of Service (TOS) value.

version
Displays the timestamp during ping output.

Modes

Operational mode

Usage Guidelines

Use this command to perform a "traceroute" operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP "Time exceeded" reply from a gateway.

Examples

The following example illustrates a traceroute to google.com waiting 2 seconds between probes.

```
vyatta@vyatta#traceroute google.com wait-time 2
traceroute to google.com (74.125.224.2), 30 hops max, 60 byte packets
1  pipsqueak.attlocal.net (10.0.6.1)  0.409 ms  0.452 ms  0.346 ms
2  75-25-153-214.uvs.sntcca.sbcglobal.net (75.25.153.214)  13.497 ms  13.517 ms  13.509 ms
3  76-198-128-2.lightspeed.mtvwca.sbcglobal.net (76.198.128.2)  35.910 ms  34.763 ms  35.876 ms
4  71.145.0.192 (71.145.0.192)  35.981 ms  35.979 ms  34.775 ms
5  12.83.39.189 (12.83.39.189)  39.654 ms  39.662 ms  44.599 ms
6  12.122.136.181 (12.122.136.181)  85.423 ms  84.396 ms  84.358 ms
7  * * *
8  216.239.49.168 (216.239.49.168)  33.291 ms  27.613 ms  28.017 ms
9  72.14.232.33 (72.14.232.33)  45.214 ms  45.849 ms  44.999 ms
10  nuq04s18-in-f2.1e100.net (74.125.224.2)  30.505 ms  29.739 ms  30.298 ms
```

traceroute <protocol> <host>

Displays the route that packets take to a network host.

Syntax

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

Parameters

ipv4

Displays the route that packets take to the IPv4 address of the host. This keyword is used when the host is specified as a host name rather than as an IP address.

ipv6

Displays the route that packets take to the IPv6 address of the host. This keyword is used when the host is specified as a host name rather than as an IP address.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), as an IPv4 or IPv6 address, or as a MAC address (format <h:h:h:h:h:h:h:h>).

option

Displays the route that packets take to the host. This keyword is used when the host is specified as a host name rather than as an IP address.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

traceroute <host>

Displays the route that packets take to a network host.

Syntax

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

Parameters

ipv4

Displays the route that packets take to the IPv4 address of the host. This keyword is used when the host is specified as a host name rather than as an IP address.

ipv6

Displays the route that packets take to the IPv6 address of the host. This keyword is used when the host is specified as a host name rather than as an IP address.

host

A host that is the destination for the trace. The host is specified as a name (if DNS is being used on the network), an IPv4 or IPv6 address, or MAC address.

Modes

Operational mode

Usage Guidelines

Use this command to perform a “traceroute” operation for a network host. This operation uses the IP protocol time-to-live (TTL) field and attempts to elicit an Internet Control Message Protocol (ICMP) TIME_EXCEEDED response from each gateway along the path to a host to track the route that a set of packets follows. It attempts to trace the route an IP packet follows to an Internet host by launching User Datagram Protocol (UDP) probe packets with a small time to live, then listening for an ICMP “Time exceeded” reply from a gateway.

ECMP

- ECMP overview 155

ECMP overview

ECMP is a technique that routes packets along multiple paths of equal cost. ECMP provides a load-balancing mechanism to ensure optimum usage of a routing path.

The Brocade vRouter supports the following load-balancing mechanisms:

- Modulo-n-hash
- Hash-threshold
- Highest Random Weight (HRW)

The Brocade vRouter calculates the key of the packet flow for every ECMP selection algorithm. The next-hop selection algorithm calculates the key of the flow and chooses the next hop.

The Brocade vRouter supports the HRW load-balancing mechanism by default. You can change the ECMP mode, if required.

ECMP is enabled on Border Gateway Protocol (BGP) by configuring the maximum number of ECMP routes for External BGP (eBGP) or Internal BGP (iBGP).

ECMP Commands

- protocols ecmp disable....157
- protocols ecmp maximum-paths....158
- protocols ecmp mode <mode>....159
- show dataplane route....161
- show dataplane route6....162

protocols ecmp disable

Disables ECMP routing.

Syntax

set protocols ecmp disable

Command Default

None.

Modes

Configuration mode.

Configuration Statement

```
protocols {
    ecmp {
        disable {}
    }
}
```

protocols ecmp maximum-paths

Sets the maximum number of next hops for ECMP routing.

Syntax

set protocols ecmp maximum-paths *number*

delete protocols ecmp maximum-paths

Command Default

None

Parameters

maximum-paths *number*

Sets the maximum number of next hops for ECMP routing.

Modes

Configuration mode.

Configuration Statement

```
protocols {
   ecmp {
      maximum-paths number{}
   }
}
```

Usage Guidelines

Use the **set** form of this command to set the maximum number of next hops for ECMP routing.

Use the **delete** form of this command to remove the maximum number of next hops for ECMP routing.

protocols ecmp mode <mode>

Sets the load-balancing mechanism for ECMP.

Syntax

set protocols ecmp mode *hash-threshold* | *hrw* | *modulo-n*

delete protocols ecmp mode *hash-threshold* | *hrw* | *modulo-n*

show protocols ecmp mode *hash-threshold* | *hrw* | *modulo-n*

Command Default

None

Parameters

hash-threshold

Sets the hash-threshold ECMP routing mode.

hrw

Sets the highest random weight ECMP routing mode. This mode is the default.

modulo-n

Sets the modulo-n-hash ECMP routing mode.

Modes

Configuration mode.

Configuration Statement

```
protocols {
   ecmp {
      mode {
         hash-threshold
         hrw
         modulo-n
         }
   }
}
```

Usage Guidelines

Use the **set** form of this command to set the load-balancing mechanism for ECMP.

Use the **delete** form of this command to remove the load-balancing mechanism. The ECMP load-balancing mechanism returns to its default setting, which is HRW.

Use the **show** form of this command to display the current load-balancing mechanism for ECMP.

Examples

The following example shows how to set the hash-threshold mode with maximum number of next hops as 34 for ECMP routing.

```
vyatta@vyatta# set protocols ecmp maximum-paths 34
[edit]
vyatta@vyatta# set protocols ecmp mode hash-threshold
[edit]
vyatta@vyatta# commit
```

The output for the ECMP configuration is as follows.

```
vyatta@vyatta# show protocols ecmp
 ecmp {
        maximum-paths 34
        mode hash-threshold
 }
[edit]
```

show dataplane route

Displays forward information base (FIB) table that contain all routes including ECMP routes.

Syntax

show dataplane route

Parameters

None.

Modes

Operational mode.

Usage Guidelines

Use this command to display the FIB table.

NOTE

FIB table is stored in the data plane.

show dataplane route6

Displays FIB table that contain all IPv6 routes including ECMP routes.

Syntax

show dataplane route6

Parameters

None

Modes

Operational mode.

Usage Guidelines

Use this command to display IPv6 FIB table.

Static Routes

- Static route configuration....163
- Static IPv6 route configuration....165

Static route configuration

This section presents the following topics:

- Static routes overview on page 163
- Configuring static routes on page 163
- Creating floating static routes on page 164
- Showing static routes in the routing table on page 165

Static routes overview

A static route is a manually configured route, which, in general, cannot be updated dynamically from information about the network topology learned by the Brocade vRouter. However, if a link fails, the router removes the routes, including static routes, from the Routing Information Base (RIB) that use this interface to reach the next hop.

Usually, static routes should be used only 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 learns from its configuration, or from its dynamic routing protocols, is stored in its RIB.

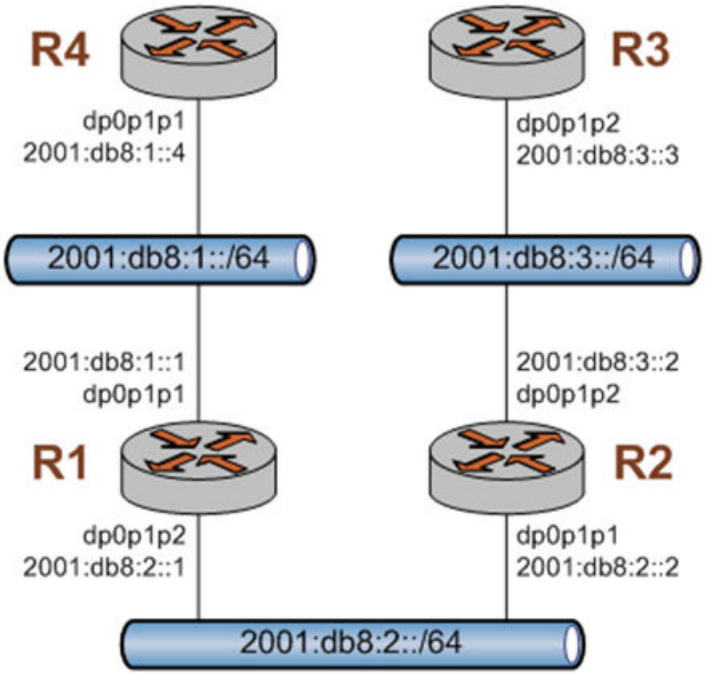
Unicast routes are directly used to determine the forwarding table for unicast packet forwarding.

Blackhole routes are static unreachable routes that can be configured to send ICMP unreachable responses on packets.

Configuring static routes

Figure 1presents sample configurations of basic static routes. When you are finished with Configuring static routes, the system is configured as shown in the figure. In the 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 1 Static routes



This section includes the following example:

- Configuring static routes

Table 1 shows how to create a static route to the 11.0.0.0/8 network that is directed toward 172.16.0.26.

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

TABLE 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`

View the configuration.	`vyatta@R1# show protocols static route`
	` route 11.0.0.0/8 {`
	`  next-hop 172.16.0.26 {`
	`  }`
	`}`

Creating floating static routes

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

NOTE

When configuring the administrative distance (AD) of a protocol, keep in mind when you specify the distance value of 255, the router will disbelieve the source and will not add the route to the routing table.

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** command, as shown in the following example.

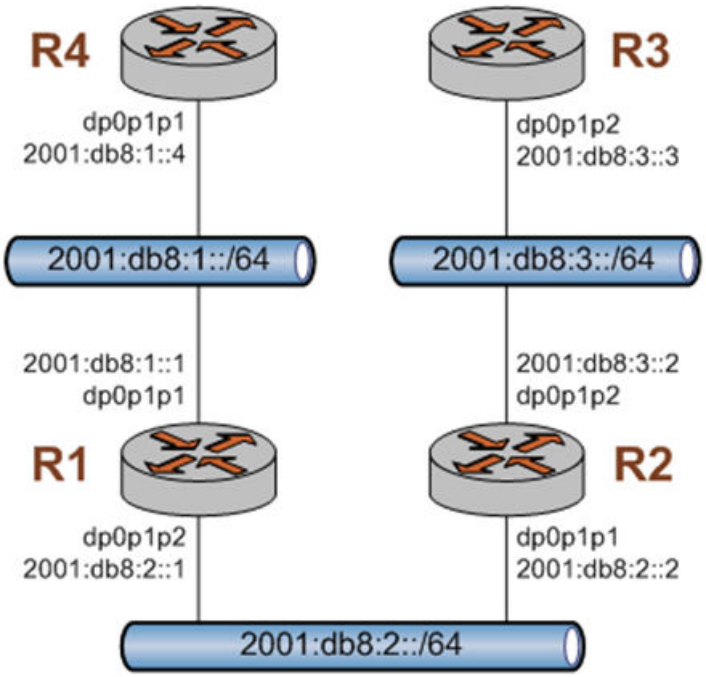
Showing static routes in the routing table

```
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, dp0p0p0
vyatta@R1:~$
```

Static IPv6 route configuration

Figure 2 shows an IPv6 network with three nodes. Verify that IPv6 forwarding is enabled on page 166 shows how to configure nodes that use static routes to enable R2 and R4 to communicate through R1.

FIGURE 2 Static IPv6 routing example



Verify that IPv6 forwarding is enabled

For R1 to be able to pass data between the dp0p0p0 and dp0p0p2 interfaces (that is, between R4 and R2), R1 must be configured to enable forwarding. To determine if forwarding is enabled, perform the following step in operational mode.

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 in the example below, the system must be configured to enable forwarding. To enable forwarding, perform the following steps in configuration mode.

Step	Command
Enable forwarding on R1.	`vyatta@R1# delete system ipv6 disable-forwarding`
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 is sent to R1. To configure the default route, perform the following steps in configuration mode.

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, dp0p0p0`
	`C>* ::1/128 is directly connected, lo`
	`C>* 2001:db8:1::/64 is directly connected, dp0p0p0`
	`C * fe80::/64 is directly connected, dp0p0p1`
	`C>* fe80::/64 is directly connected, dp0p0p0`
	`K>* ff00::/8 is directly connected, dp0p0p0`

Add a static IPv6 route

As an alternative to the default route created on R4, 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.

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, dp0p0p0`
	`C>* 2001:db8:2::/64 is directly connected, dp0p0p0`
	`C * fe80::/64 is directly connected, dp0p0p1`
	`C>* fe80::/64 is directly connected, dp0p0p0`
	`K>* ff00::/8 is directly connected, dp0p0p0`

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.

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 the **traceroute** command to verify that the route goes from R2 to R1 to R4. To confirm connectivity between R2 and R4 through R1 by using the **traceroute** command, perform the following step in operational mode.

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

- protocols static interface-route <subnet> next-hop-interface <interface> 170
- protocols static interface-route6 <subnet> next-hop-interface <interface> 171
- protocols static route <subnet> blackhole <distance> 172
- protocols static route <subnet> next-hop <address> 173
- protocols static route6 <subnet> blackhole 174
- protocols static route6 <subnet> next-hop <address> 175
- protocols static table <table> interface-route <subnet> next-hop-interface <interface> 177
- protocols static table <table> route <subnet> blackhole <distance> 179
- protocols static table <table> route <subnet> next-hop <address> 180
- protocols static table <table> route6 <subnet> next-hop <address> 182
- protocols static table <table> route6 <subnet> blackhole [distance] 184

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

Configures 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**]

Parameters

interface-route *subnet*

Multi-node. An interface-based static route. The format is a destination subnet of the form *address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*).

You can define multiple interface-based routes by creating multiple **interface-route** configuration nodes.

next-hop-interface *interface*

The next-hop interface.

disable

Disables the interface-based static route.

distance *distance*

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

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

Use the **delete** form of this command to remove the next-hop interface from a static route.

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

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

Configures the next-hop interface for an interface-based IPv6 static route.

Syntax

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

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

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

Parameters

interface-route6 *subnet*

Multi-node. An interface-based static route. The format is a destination subnet of the form *address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*).

You can define multiple interface-based routes by creating multiple **interface-route** configuration nodes.

next-hop-interface *interface*

The next-hop interface.

disable

Disables the interface-based IPv6 static route.

distance

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use the **set** form of this command to configure the next-hop interface for an IPv6 static route.

Use the **delete** form of this command to remove the next-hop interface from an IPv6 static route.

Use the **show** form of this command to view the next-hop interface for an IPv6 static route.

protocols static route <subnet> blackhole <distance>

Configures 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**]

Parameters

route *subnet*

Multi-node. A static route. The format is a destination subnet of the form *address*/*prefix* (*x.x.x.x*/*x*).
You can define multiple static routes by creating multiple **route** configuration nodes.

blackhole

A destination router that is offline and cannot receive traffic or provide messages to the source of the traffic.

distance *distance*

The black hole distance for this route. Routes with a smaller distance are selected before those with a larger distance.
The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

A black hole static route is a route for which the system silently discards packets that are matched.

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

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

Use the **show** form of this command to view a black hole static route.

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

Configures 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**]

Parameters

route *subnet*

Multi-node. A static route. The format is a destination subnet of the form *address*/*prefix* (*x.x.x.x*/*x*). You can define multiple static routes by creating multiple **route** configuration nodes.

address

The address of the next-hop router.

disable

Disables the static route.

distance

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use the **set** form of this command to configure the next hop for a static route.

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

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

protocols static route6 <subnet> blackhole

Configures a black hole IPv6 static route.

Syntax

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

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

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

Parameters

route6 *subnet*

Multi-node. An IPv6 static route. The format is a destination subnet of the form IPv6- *address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*). You can define multiple static routes by creating multiple **route** configuration nodes.

blackhole *distance*

The black hole distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

A black hole static route silently discards packets that are matched.

Use the **set** form of this command to configure a black hole IPv6 static route.

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

Use the **show** form of this command to view a black hole IPv6 static route.

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

Configures 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**]

Parameters

route6 *subnet*

Multi-node. An IPv6 static route. The format is a destination subnet of the form IPv6- *address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*). You can define multiple static routes by creating multiple **route6** configuration nodes.

next-hop *address*

The IPv6 address of the next hop router.

disable

Disables the IPv6 static route.

distance *distance*

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance.

interface

The outgoing interface used to reach the next-hop address. This interface is needed when the next-hop address is a link-local address (that is, it has a fe80::/64 prefix).

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

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

Use the **set** form of this command to configure the next hop for an IPv6 static route.

Use the **delete** form of this command to remove the next hop from an IPv6 static route.

Use the **show** form of this command to view the next hop for an IPv6 static route.

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

Configures the next-hop interface for an interface-based static route in an alternate routing table.

Syntax

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

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

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

Parameters

static
A static route in an alternate routing table.

table *table*
Multi-node. An alternate routing table to be used by policy-based routing rules.

interface-route *subnet*
Multi-node. An interface-based static route. The format is a destination subnet of the form *address*/*prefix* (*x.x.x.x*/*x*). You can define multiple interface-based routes by creating multiple **interface-route** configuration nodes.

next-hop-interface *interface*
The next-hop interface.

disable
Disables the interface-based static route.

distance *distance*
The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

```
protocols {
    static {
        table 1
            interface-route subnet {
                next-hop-interface interface {
                    disable
                    distance distance
                }
            }
        }
    }

}
```

Usage Guidelines

Use this command to configure interface-based static routes in an alternate routing table. The alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

Use the **set** form of this command to configure a next-hop interface.

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

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

protocols static table <table> route <subnet> blackhole <distance>

Configures a a black hole static route in an alternate routing table.

Syntax

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

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

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

Parameters

table *table*

Multi-node. An alternate routing table to be used by policy-based routing rules.

route *subnet*

Multi-node. Defines a static route. The format is a destination subnet of the form *address*/*prefix* (*x.x.x.x*/*x*). You can define multiple static routes by creating multiple **route** configuration nodes.

distance *distance*

The black hole distance for this route. Routes with a smaller distance are selected before those with a larger distance. The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use this command to configure a black hole static route in an alternate policy route table. A black hole route is a route for which the system silently discards packets that are matched.

The alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

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

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

Use the **show** form of this command to view a black hole static route.

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

Configures the next hop for a static route in an alternate routing table.

Syntax

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

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

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

Parameters

table *table*

Multi-node. An alternate routing table to be used by policy-based routing rules.

route *subnet*

Multi-node. Defines a static route. The format is a destination subnet of the form *address*/*prefix* (*x.x.x.x*/*x*). You can define multiple static routes by creating multiple **route** configuration nodes.

next-hop *address*

The address of the next-hop router.

disable

Disables the static route.

distance *distance*

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance. The distance ranges from 1 through 255. The default distance is 1.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use this command to configure static routes in an alternate routing table. The alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

Use the **set** form of this command to configure the next hop for a route in an alternate routing table.

Use the **delete** form of this command to remove the next hop from a static route in an alternate routing table.

Use the **show** form of this command to view the next hop for a static route in an alternate routing table.

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

Configures the next hop for an IPv6 static route in an alternate routing table.

Syntax

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

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

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

Parameters

table *table*

Multi-node. An alternate routing table to be used by policy-based routing rules.

route6 *subnet*

Multi-node. An IPv6 static route. The format is a destination subnet of the form IPv6- *address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*). You can define multiple static routes by creating multiple **route6** configuration nodes.

disable

Disables the IPv6 static route.

distance *distance*

The next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use this command to configure IPv6 static routes on the system in an alternate routing table. The alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

Use the **set** form of this command to configure the next hop for an IPv6 static route in an alternate routing table.

Use the **delete** form of this command to remove the next hop for an IPv6 static route in an alternate routing table.

Use the **show** form of this command to view the next hop for an IPv6 static route in an alternate routing table.

protocols static table <table> route6 <subnet> blackhole [distance]

Configures a black hole static route in an alternate routing table.

Syntax

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

delete protocols static table *table* **route6** *subnet* **blackhole distance**

show protocols static table *table* **route6** *subnet* **blackhole distance**

Parameters

table *table*

Multi-node. An alternate routing table to be used by policy-based routing rules.

route6 *subnet*

Multi-node. An IPv6 static route. The format is a destination subnet of the form IPv6-*address*/*prefix* (*h:h:h:h:h:h:h:h*/*x*). You can define multiple static routes by creating multiple **route** configuration nodes.

blackhole

A destination router that is offline and cannot receive traffic or provide messages to the source of the traffic.

distance *distance*

The black hole distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Modes

Configuration mode

Configuration Statement

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

}
```

Usage Guidelines

Use this command to configure a black hole IPv6 static route in an alternate routing table. A black hole route silently discards packets that are matched.

The alternate routing tables are used with policy-based routing. Refer to *Brocade 5600 vRouter Policy-based Routing Configuration Guide* for information on policy-based routing.

Use the **set** form of this command to configure a black hole IPv6 static route.

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

Use the **show** form of this command to view a black hole IPv6 static route.

Source Routes

- Source routing example.....187

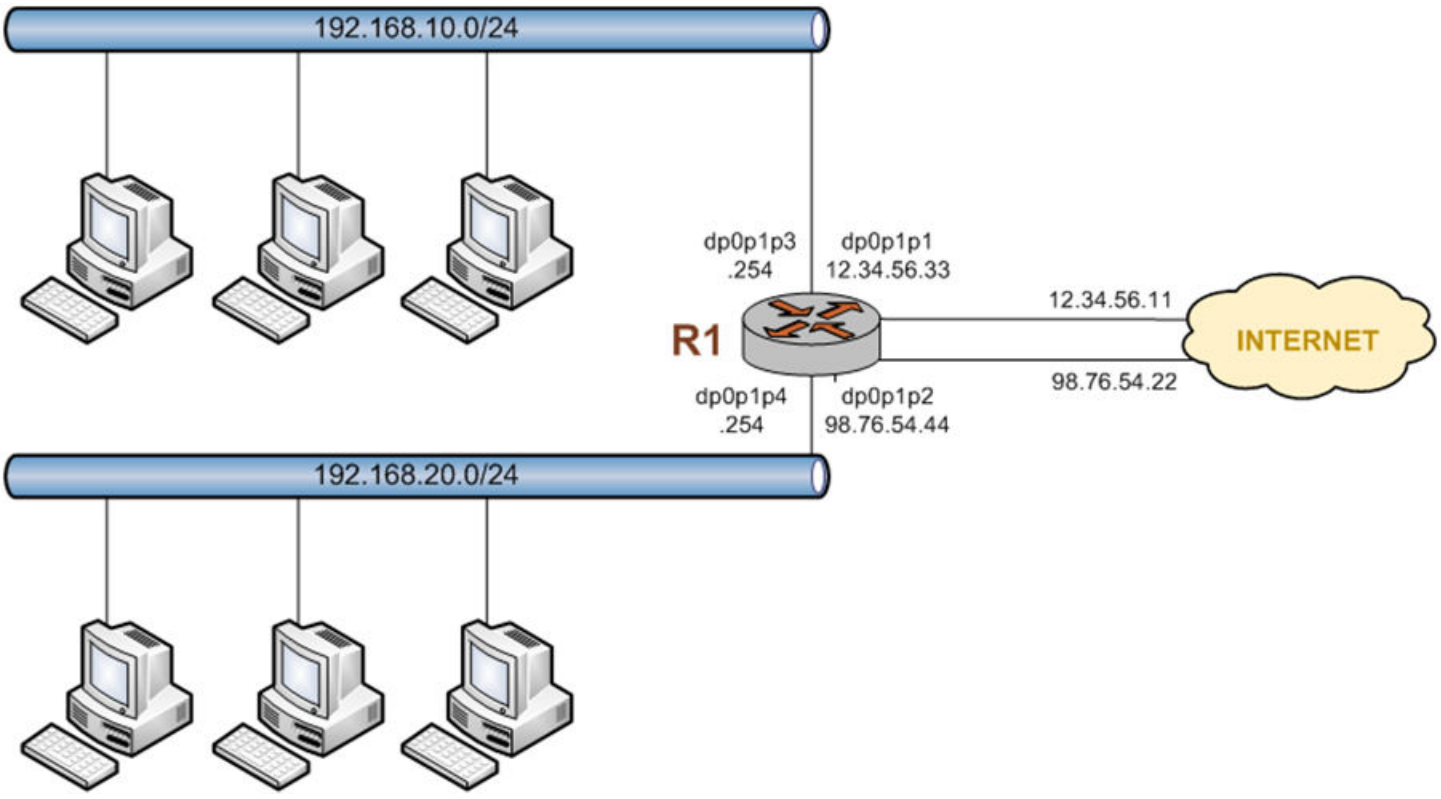
Source routing example

Figure 3 shows a simple site using PBR on the Brocade vRouter (R1) to route traffic from two different internal subnets to two Internet links.

In this example:

- All Internet-bound traffic from subnet 192.168.10.0/24 is routed out interface dp0p0p1.
- All Internet-bound traffic from subnet 192.168.20.0/24 is routed out interface dp0p0p2.

FIGURE 3 Source routing using PBR



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

Step	Command
Create the SRC-ROUTE policy.	`vyatta@R1# set policy route pbr SRC-ROUTE`

Create rule 10 and specify the destination address to match. In this case, any destination address will match.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 10 destination address 0.0.0.0/0`
Specify the source address to match. In this case, any address on subnet 192.168.10.0/24 will match.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 10 source address 192.168.10.0/24`
Specify that all packets that match should use alternate routing table 1.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 10 set table 1`
Create rule 20 and specify the destination address to match. In this case, any destination address will match.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 20 destination address 0.0.0.0/0`
Specify the source address to match. In this case, any address on subnet 192.168.20.0/24 will match.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 20 source address 192.168.20.0/24`
Specify that all packets that match should use alternate routing table 2.	`vyatta@R1# set policy route pbr SRC-ROUTE rule 20 set table 2`
Commit the change.	`vyatta@R1# commit`

Show the policy-based routing configuration.	`vyatta@R1# show policy`
	` route SRC-ROUTE {`
	`  rule 10 {`
	`  destination {`
	`  address 0.0.0.0/0`
	`  }`
	`  set {`
	`  table 1`
	`  }`
	`  source {`
	`  address 192.168.10.0/24`
	`  }`
	`  }`
	`  rule 20 {`
	`  destination {`
	`  address 0.0.0.0/0`
	`  }`
	`  set {`
	`  table 2`
	`  }`
	`  source {`
	`  address 192.168.20.0/24`
	`  }`
	`  }`
	` }`

Create the alternative routing table 1 and route default traffic to the first Internet connection.	`vyatta@R1# set protocols static table 1 route 0.0.0.0/0 nexthop 12.34.56.11`
Create the alternative routing table 2 and route default traffic to the second Internet connection.	`vyatta@R1# set protocols static table 2 route 0.0.0.0/0 nexthop 98.76.54.22`
Commit the change.	`vyatta@R1# commit`

Show the alternate routing table configuration.	`vyatta@R1# show protocols static`
	` table 1 {`
	`  route 0.0.0.0/0 {`
	`  next-hop 12.34.56.11 {`
	`  }`
	`  }`

	}
	table 2 {
	route 0.0.0.0/0 {
	next-hop 98.76.54.22 {
	}
	}
	}

Assign an address to dp0p0p1.	vyatta@R1# set interfaces dataplane dp0p0p1 address 12.34.56.33/24
Assign an address to dp0p0p2.	vyatta@R1# set interfaces dataplane dp0p0p2 address 98.76.54.44/24
Assign an address to dp0p0p3.	vyatta@R1# set interfaces dataplane dp0p0p3 address 192.168.10.254/24
Assign the policy to the interface connected to subnet 192.168.10.0/24.	vyatta@R1# set interfaces dataplane dp0p0p3 policy route SRC-ROUTE
Assign an address to dp0p0p4.	vyatta@R1# set interfaces dataplane dp0p0p4 address 192.168.20.254/24
Assign the policy to the interface connected to subnet 192.168.20.0/24.	vyatta@R1# set interfaces dataplane dp0p0p4 policy route SRC-ROUTE
Commit the change.	vyatta@R1# commit

Show the dataplane interface configuration.	vyatta@R1# show interfaces dataplane
	dataplane dp0p0p1 {
	address 12.34.56.33/24
	duplex auto
	hw-id 00:93:0b:23:ab:e1
	speed auto
	}
	dataplane dp0p0p2 {
	address 98.76.54.44/24
	duplex auto
	hw-id 00:93:0b:23:ab:e2
	speed auto
	}
	dataplane dp0p0p3 {
	address 192.168.10.254/24
	duplex auto
	hw-id 00:93:0b:23:ab:e3
	policy {
	route SRC-ROUTE
	}
	speed auto
	}
	dataplane dp0p0p4 {
	address 192.168.20.254/24
	duplex auto
	hw-id 00:93:0b:23:ab:e4
	policy {
	route SRC-ROUTE
	}
	speed auto
	}

List of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System

IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure

PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access