53-1003732-03
14 September 2015

Brocade 5600 vRouter Services

Reference Guide

Supporting Brocade 5600 vRouter 3.5R6

BROCADE

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

Contents

Preface

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.

Convention	Description
[ ]	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.
...	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 how to configure the DHCP, DHCPv6, DNS, flow monitoring, NHRP, sFlow, TWAMP, and port monitoring services on the Brocade 5600 vRouter (referred to as a virtual router, vRouter, or router in the guide).

DHCP

DHCP overview

DHCP allows dynamic assignment of reusable IP addresses and other configuration information to DHCP clients. This assignment reduces costs, configuration effort, and management burden associated with Internet access. On the other hand, it also increases network and service overhead.

In DHCP, the server assigns an IP address and other configuration parameters to a client for a limited period of time. This period of time is called the lease. The lease is valid for the period you configure on the Brocade vRouter or until the client explicitly relinquishes the address.

To use the DHCP service, you define a pool of IP addresses for each subnet assigned by the DHCP server. Each DHCP address pool is mapped to a subnet associated with the system. For each address pool, you can specify the length of time an address is valid (its lease duration). The default lease duration is 24 hours. You can also specify a number of different servers (for example DNS, WINS, SMTP, and others) that are available to clients on the subnet.

To create an IP address pool for clients on a subnet to which the router is not directly connected (that is, without having an interface into that network), you can use service dhcp-server listento interface <dp-interface> on page 47. See Configuring for networks indirectly connected to the system on page 21.

You can statically map an IP address to the MAC address of a device. The DHCP service listens on UDP port 67 for lease requests from DHCP clients. The request packet allows the system to determine the interface on which the client is located. It then assigns an IP address from the appropriate pool and binds it to the client.

In addition to providing a DHCP server, individual interfaces on the Brocade vRouter can be configured as DHCP clients. For details, see the Brocade 5600 Virtual Router documentation for the interface you are interested in configuring as a DHCP client.

The Brocade vRouter also supports DHCP relay.

A DHCP relay agent receives DHCP packets from DHCP clients and forwards them to a DHCP server. This allows you to place DHCP clients and DHCP servers on different networks; that is, across router interfaces.

The relay agent is configured with addresses of DHCP servers to which they should relay client DHCP message. The relay agent intercepts the broadcast, sets the gateway address (the **giaddr** field of the DHCP packet) and, if configured, inserts the Relay Agent Information option (option 82) in the packet and forwards it to the DHCP server.

The DHCP server echoes the option back verbatim to the relay agent in server-to-client replies, and the relay agent strips the option before forwarding the reply to the client.

DHCP configuration

DHCP configuration includes enabling the DHCP server, setting up the DHCP servers for failover, configuring DHCP address pools, setting up DHCP relay, and setting up additional DHCP configuration parameters.

Enabling the DHCP server

To use the DHCP server on the Brocade vRouter, you must enable the DHCP service.

To enable the DHCP service, perform the following steps in configuration mode.

TABLE 1 Enabling the DHCP service

Step	Command
Enable DHCP.	`vyatta@R1# set service dhcp-server`

In addition, at least one DHCP shared network (address pool) must be configured.

Creating a static mapping

Situations exist in which it makes sense to map a specific IP address to a specific host rather than dynamically assign an IP address from a pool of addresses. This mapping is known as a “static mapping.”

A static mapping is defined by using the **static-mapping** option of the service dhcp-server configuration node. The following example shows how to create a static mapping by associating the 172.16.0.101 IP address to the device with a MAC address of 00:15:c5:b3:2e:65.

TABLE 2 Creating a static mapping

Step	Command
Create a static mapping called **lab** and specify the static IP address.	`vyatta@R1# set service dhcp-server static-mapping lab ip-address 172.16.0.101`
Specify the associated MAC address within the lab static mapping.	`vyatta@R1# set service dhcp-server static-mapping lab mac-address 00:15:c5:b3:2e:65`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service dhcp-server`
	`static-mapping lab`
	`{`
	`ip-address 172.16.0.101`
	`mac-address 00:15:c5:b3:2e:65`
	`}`

Configuring DHCP address pools

Configure DHCP address pools for the system to act as a DHCP server for the network.

Configuring for networks directly connected to the system

Table 3 shows how to create three address pools:

- dp0p1p1_POOL. This address pool serves the 172.16.0.0/24 subnet, which is connected to the dp0p1p1 interface. The lease time remains at the default, 24 hours (86,400 seconds). This address pool uses the DNS name server at 172.16.0.34.
- dp0p1p2_30_POOL. This address pool serves the 10.10.30.0/24 subnet, which is connected directly to the dp0p1p2 interface. The lease time remains at the default, 24 hours (86,400 seconds). This address pool uses the DNS name server at 10.10.40.34, which is directly connected to dp0p1p2.40 (that is, dp0p1p2 virtual interface [vif] 40).
- dp0p1p2_40_POOL. This address pool serves the 10.10.40.0/24 subnet, which is connected to the dp0p1p2.40 interface. The lease time remains at the default, 24 hours (86,400 seconds). This address pool uses the DNS name server at 10.10.40.34, which is connected to dp0p1p2.40.

In all these pools, the range of addresses is configured for .100 through .199.

The following figure shows the sample address pool configuration.

FIGURE 1 DHCP address pool configuration

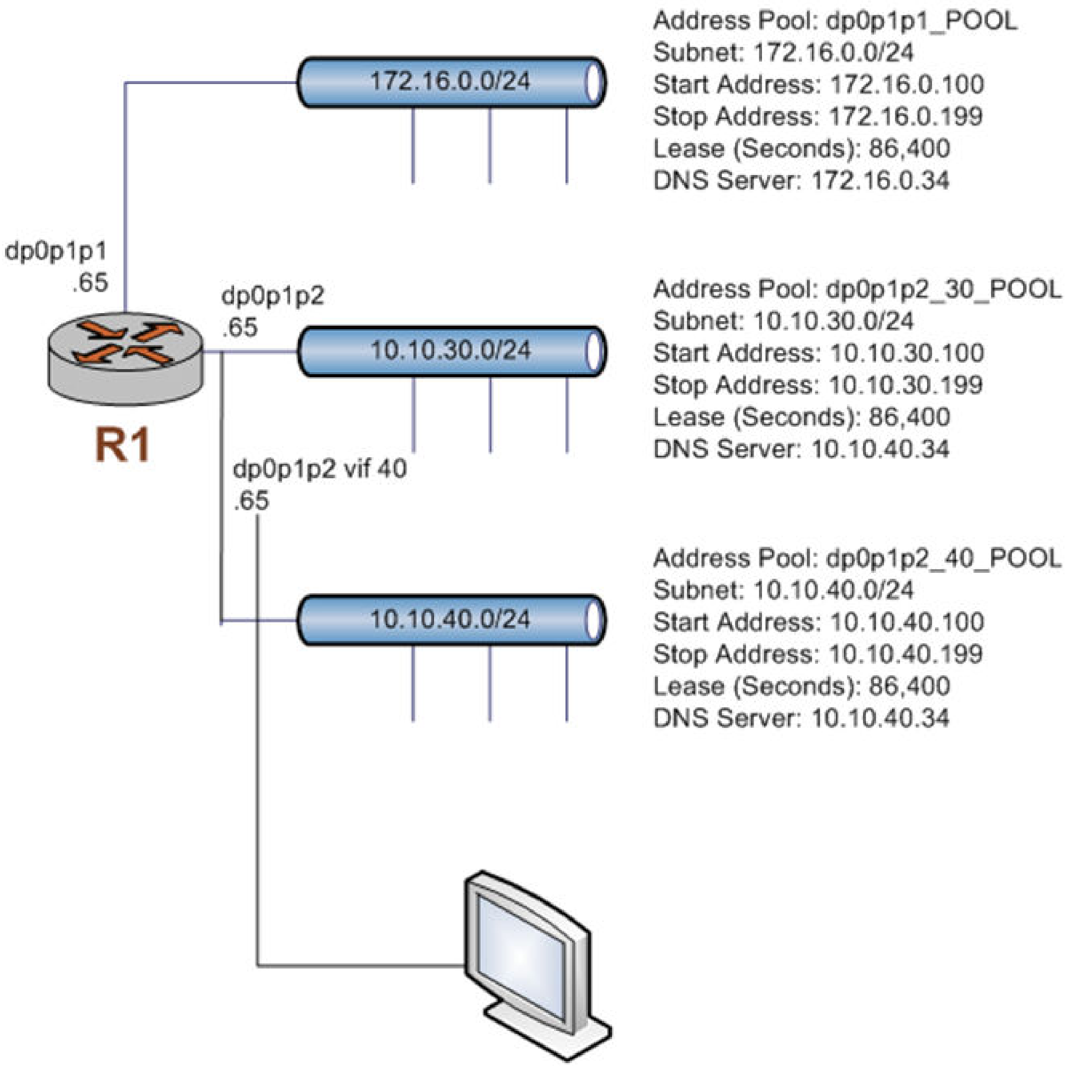


TABLE 3 Configuring DHCP address pools

Step	Command
Create the configuration node for dp0p1p1_POOL on the 172.16.0.0/24 subnet. Specify the start and stop IP addresses for the pool.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p1_POOL subnet 172.16.0.0/24 start 172.16.0.100 stop 172.16.0.199`
Specify the default router for dp0p1p1_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p1_POOL subnet 172.16.0.0/24 default-router 172.16.0.65`
Specify a DNS server for dp0p1p1_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p1_POOL subnet 172.16.0.0/24 dns-server 172.16.0.34`
Create the configuration node for dp0p1p2_30_POOL on the 10.10.30.0/24 subnet. Specify the start and stop IP addresses for the pool.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_30_POOL subnet 10.10.30.0/24 start 10.10.30.100 stop 10.10.30.199`
Specify the default router for dp0p1p2_30_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_30_POOL subnet 10.10.30.0/24 default-router 10.10.30.65`
Specify a DNS server for dp0p1p2_30_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_30_POOL subnet 10.10.30.0/24 dns-server 10.10.40.34`
Create the configuration node for dp0p1p2_40_POOL on the 10.10.40.0/24 subnet. Specify the start and stop IP addresses for the pool.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_40_POOL subnet 10.10.40.0/24 start 10.10.40.100 stop 10.10.40.199`
Specify the default router for dp0p1p2_40_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_40_POOL subnet 10.10.40.0/24 default-router 10.10.40.65`
Specify a DNS server for dp0p1p2_40_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_40_POOL subnet 10.10.40.0/24 dns-server 10.10.40.34`
Commit the changes.	`vyatta@R1# commit`

TABLE 3 Configuring DHCP address pools (Continued)

Show the configuration.

```
vyatta@R1# show service dhcp-server
        shared-network-name dp0p1p1_POOL {
        subnet 172.16.0.0/24 {
            default-router 172.16.0.65
            dns-server 172.16.0.34
            start 172.16.0.100 {
                stop 172.16.0.199
            }
        }
    }
    shared-network-name dp0p1p2_30_POOL {
        subnet 10.10.30.0/24 {
            default-router 10.10.30.65
            dns-server 10.10.40.34
            start 10.10.30.100 {
                stop 10.10.30.199
            }
        }
    }
    shared-network-name dp0p1p2_40_POOL {
        subnet 10.10.40.0/24 {
            default-router 10.10.40.65
            dns-server 10.10.40.34
            start 10.10.40.100 {
                stop 10.10.40.199
            }
        }
    }
```

Show the interface configuration.

```
vyatta@R1# show interfaces
 dataplane dp0p1p1 {
     address 172.16.0.65/24
     hw-id 00:0c:29:42:05:2b
 }
 dataplane dp0p1p2 {
     address 10.10.30.65/24
     hw-id 00:0c:29:42:05:35
     vif 40 {
         address 10.10.40.65/24
     }
 }
```

Configuring for networks indirectly connected to the system

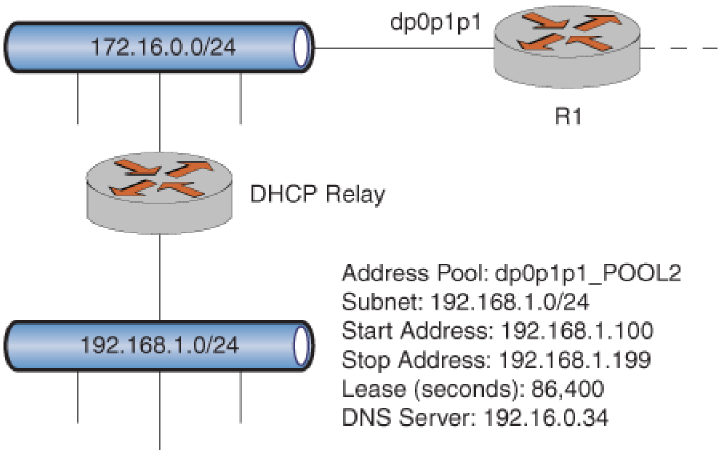
Table 4 shows how to create an address pool (dp0p1p1_POOL2) for clients that are indirectly connected to the R1 vRouter, as shown below.

The dp0p1p1_POOL2 address pool serves the 192.168.1.0/24 subnet, which is on a different subnet than the subnet to which the dp0p1p1 dataplane interface is connected.

The lease time remains at the default, 24 hours (86,400 seconds). This address pool uses the DNS name server at 172.16.0.34.

The following figure shows the sample address pool configuration.

FIGURE 2 DHCP address pool configuration for clients indirectly connected to the DHCP server host



To configure the dp0p1p1_POOL2 DHCP address pool, perform the following steps in configuration mode.

TABLE 4 Configuring DHCP address pools

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Configure the router interface to listen to DHCP messages.</td><td><pre>vyatta@R1#set service dhcp-server listento
interface dp0p1p1</pre></td></tr>
<tr><td>Create a shared network and associate it with the 192.168.1.0/24 subnet.</td><td><pre>vyatta@R1# set service dhcp-server shared-network-
name dp0p1p1_POOL2 subnet 192.168.1.0/24</pre></td></tr>
<tr><td>Show the configuration.</td><td><pre>vyatta@R1# show service dhcp-server
dhcp-server {
    listento {
        interface dp0p1p1
    }
    shared-network-name dp0p1p1_POOL2 {
        subnet 192.168.1.0/24 {
            dns-server 192.168.1.100
            lease 86400
            start 192.168.1.100 {
                stop 192.168.1.199
            }
        }
    }
}</pre></td></tr>
<tr><td>Show the interface configuration.</td><td><pre>vyatta@R1# show interfaces
interfaces {
    dataplane dp0p1p1 {
        address 172.16.0.0/24
    }
    loopback lo
}</pre></td></tr>
<tr><td>Commit the changes.</td><td><pre>vyatta@R1# commit</pre></td></tr>
</table>

Setting up DHCP servers for failover

The Brocade vRouter also provides a failover feature to allow for DHCP redundancy on a given subnet.

In a failover configuration, two DHCP servers act as failover peers, with one of the peers designated as the primary and the other as the secondary. For DHCP failover to work, the following conditions must be met.

- Both peers must be Brocade vRouters and must be running the same version of Vyatta software.
- Each server must be configured to point to the other as the failover peer.
- The time on the servers must be exactly synchronized.
- The start-stop range must have at least one IP address for each subnet that has not been either excluded (by using service dhcp-server shared-network-name name subnet ipv4net exclude ipv4 on page 62) or statically mapped (by using service dhcp-server static-mapping mapname on page 85).

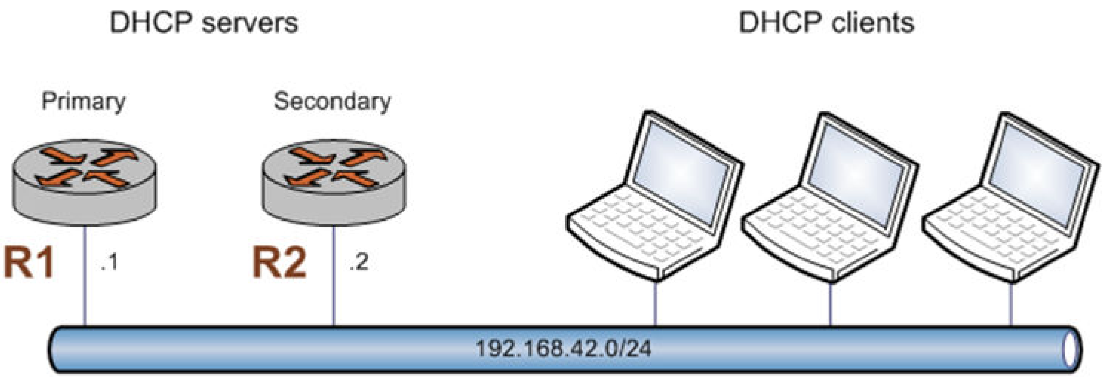
The system times should be synchronized before configuring DHCP failover. Use of NTP time synchronization is highly recommended. However, if difficulties arise because of incorrect system times, disable NTP, reset the times correctly, and then re-enable NTP.

Note that DHCP leases are assigned only in failover configurations if proper communication is established between the two failover peers. If the configuration is incorrect (if, for example, one failover peer is configured but the other is not), DHCP leases are not dispersed.

Also note that statically mapped addresses are not renewed by a failover server unless they are explicitly defined on that server by using service dhcp-server static-mapping mapname on page 85.

The following figure shows the sample DHCP server failover configuration.

FIGURE 3 DHCP server failover configuration



To configure R1 as the primary DHCP server in this failover scenario, perform the following steps in configuration mode on R1.

TABLE 5 Setting up DHCP failover on R1

Step	Command
Create the configuration node for DHCP1 on the 192.168.42.0/24 subnet. Specify the start and stop IP addresses for the pool.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 start 192.168.42.100 stop 192.168.42.199`
Specify the default router for DHCP1.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 default-router 192.168.42.254`

TABLE 5 Setting up DHCP failover on R1 (Continued)

Specify a DNS server for DHCP1.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 dns-server 192.168.42.253`
Specify the local IP address for the DHCP server for failover.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover local-address 192.168.42.1`
Specify the IP address of the peer DHCP server for failover.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover peer-address 192.168.42.2`
Specify the role that the DHCP server plays in the failover group.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover status primary`
Specify the name of the failover group.	`vyatta@R1# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover name FAILOVER`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service dhcp-server shared-network-name DHCP1`
	`    shared-network-name DHCP1 {`
	`        subnet 192.168.42.0/24 {`
	`            default-router 192.168.42.254`
	`            dns-server 192.168.42.253`
	`            failover {`
	`                local-address 192.168.42.1`
	`                name FAILOVER`
	`                peer-address 192.168.42.2`
	`                status primary`
	`            }`
	`            start 192.168.42.100 {`
	`                stop 192.168.42.199`
	`            }`
	`        }`
	`    }`

To configure R2 as the secondary DHCP server in this failover scenario, perform the following steps in configuration mode on R2.

TABLE 6 Setting up DHCP failover on R2

Step	Command
Create the configuration node for DHCP1 on the 192.168.42.0/24 subnet. Specify the start and stop IP addresses for the pool.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 start 192.168.42.100 stop 192.168.42.199`
Specify the default router for DHCP1.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 default-router 192.168.42.254`
Specify a DNS server for DHCP1.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 dns-server 192.168.42.253`
Specify the local IP address for the DHCP server for failover.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover local-address 192.168.42.2`

TABLE 6 Setting up DHCP failover on R2 (Continued)

Specify the IP address of the peer DHCP server for failover.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover peer-address 192.168.42.1`
Specify the role that the DHCP server plays in the failover group.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover status secondary`
Specify the name of the failover group.	`vyatta@R2# set service dhcp-server shared-network-name DHCP1 subnet 192.168.42.0/24 failover name FAILOVER`
Commit the changes.	`vyatta@R2# commit`

Show the configuration.	`vyatta@R2# show service dhcp-server shared-network-name DHCP1`
	`    shared-network-name DHCP1 {`
	`        subnet 192.168.42.0/24 {`
	`            default-router 192.168.42.254`
	`            dns-server 192.168.42.253`
	`            failover {`
	`                local-address 192.168.42.2`
	`                name FAILOVER`
	`                peer-address 192.168.42.1`
	`                status secondary`
	`            }`
	`            start 192.168.42.100 {`
	`                stop 192.168.42.199`
	`            }`
	`        }`
	`    }`

Setting up DHCP relay

Configure DHCP relay if you want the Brocade vRouter to forward DHCP requests to another DHCP server.

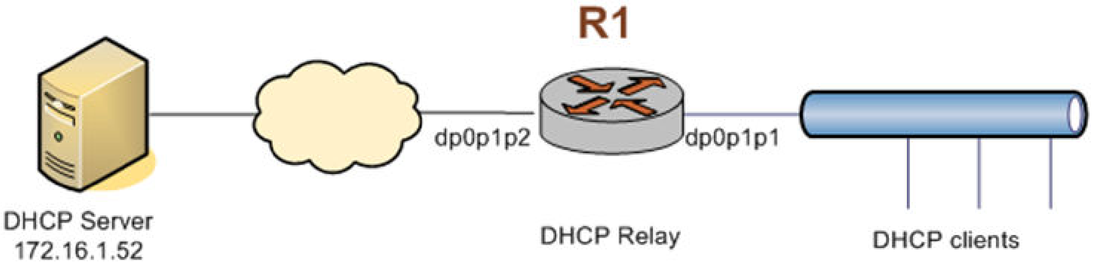
Every interface involved in the DHCP relay must be configured and must be capable of broadcasting. So, for example, if requests are coming in on the dp0p1p1 interface and the DHCP server specified in the configuration is reached through the dp0p1p2 interface, both dp0p1p1 and dp0p1p2 must be configured for DHCP.

Table 7 shows how to accomplish the following tasks:

- Configures both dp0p1p1 and dp0p1p2 for DHCP. The router is expected to receive client requests for the DHCP server through the dp0p1p1 interface. It forwards client-to-server DHCP messages to the DHCP server at 172.16.1.52 out through the dp0p1p2 interface.
- Enables relay options. This directs the system to add the Relay Agent Information option (option 82) to the DHCP message before forwarding, as specified by RFC 3046.
- Does not permit reformatting of DHCP messages by this system. If a packet is received that already contains relay information, the packet is discarded.
- Leaves other relay option parameters at default values. This means that the router uses port 67 for DHCP messaging, allows a maximum DHCP packet size of at most 576 bytes, and has a maximum hop count of 10 hops.

Figure 4 shows the sample DHCP relay configuration.

FIGURE 4 DHCP relay configuration



To configure DHCP relay, perform the following steps in configuration mode.

TABLE 7 Setting up DHCP relay

Step	Command
Enable DHCP relay on the dp0p1p1 interface on which client requests are received.	`vyatta@R1# set service dhcp-relay listen-interface dp0p1p1`
Enable DHCP relay on the dp0p1p2 interface on which client messages are forwarded to the DHCP server.	`vyatta@R1# set service dhcp-relay upstream-interface dp0p1p2`
Specify the IP address of the DHCP server.	`vyatta@R1# set service dhcp-relay server 172.16.1.52`
Set the router to discard messages containing relay information. Leave other parameters at default values.	`vyatta@R1# set service dhcp-relay relay-options relay-agents-packets discard`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service dhcp-relay`
	`    listen-interface dp0p1p1`
	`    upstream-interface dp0p1p2`
	`    server 172.16.1.52`
	`    relay-options {`
	`        relay-agents-packets discard`
	`    }`

Setting additional DHCP configuration parameters

CAUTION

This feature is advanced and should be used only by expert users in special situations.

The Brocade vRouter DHCP server commands provide a set of commonly used DHCP server features. However, many additional features are available. Information regarding the available DHCP server features are located on the dhcpd.conf man page. To access it, type the following at the Vyatta command prompt:

man dhcpd.conf

To access these additional features, use one of the following commands, depending on the required scope of the feature. The commands are listed from widest to narrowest scope.

- service dhcp-server global-parameters params on page 46
- service dhcp-server static-mapping mapname static-mapping-parameters params on page 89
- service dhcp-server shared-network-name name shared-network-parameters params on page 52
- service dhcp-server shared-network-name name subnet ipv4net subnet-parameters params on page 78

The precedence of scope of these commands is from narrowest to widest. That is, if more than one command is specified and a given host address falls within the scope of both, it is governed by parameters specified in the command with the narrowest scope.

Multiple parameter strings can be specified. Each parameter string that is specified adds a separate line to the dhcpd.conf file.

Note that no validation is done by the Brocade vRouter before passing the parameter string from these commands to the DHCP server process (dhcpd). Because of this lack of validation, it is imperative that the syntax described in the dhcpd.conf documentation be strictly followed. Failure to do so could result in a failure of the DHCP server. It is advisable to check the system log for errors when using these parameter strings. In addition, the **show system processes** command can be used to determine if the dhcpd process is still running.

The following example shows how the additional DHCP server parameters can be accessed. To configure additional DHCP server parameters, perform the following steps in configuration mode.

TABLE 8 Setting up a DHCP server with additional parameters

Step	Command
Enable the DHCP server and define an option that does not already have a keyword defined in the dhcpd process. See the dhcpd man page for further information.	`vyatta@R1# set service dhcp-server global-parameters ‘option rfc3442-static-route code 121 = string;’`
Specify the value to be used for the option for all shared networks, subnets, and static mappings defined in the DHCP server configuration.	`vyatta@R1# set service dhcp-server global-parameters ‘option rfc3442-static-route 01:01:01:01:01:01:01:01;’`
Specify an IP address to statically map to a host with a specific MAC address.	`vyatta@R1# set service dhcp-server static-mapping MAP1 ip-address 172.16.117.15`
Specify the MAC address of a host to be statically mapped to an IP address.	`vyatta@R1# set service dhcp-server static-mapping MAP1 mac-address 09:09:09:09:09:09`
Override the global value of the parameter defined previously for a specific host.	`vyatta@R1# set service dhcp-server static-mapping MAP1 static-mapping-parameters ‘option rfc3442-static-route 01:01:01:01:01:01:01:02’`
Specify that the DHCP server is authoritative for the specified shared network.	`vyatta@R1# set service dhcp-server shared-network-name NET1 authoritative enable`
Specify the subnet and address pool to use.	`vyatta@R1# set service dhcp-server shared-network-name NET1 subnet 172.16.117.0/24 start 172.16.117.10 stop 172.16.117.20`
Commit the changes.	`vyatta@R1# commit`

TABLE 8 Setting up a DHCP server with additional parameters (Continued)

Show the configuration.

```
vyatta@R1# show service dhcp-server
 global-parameters “option rfc3442-static-route
code 121  = string;”
 global-parameters “option rfc3442-static-route
01:01:01:01:01:01:01:01;”
shared-network-name NET1 {
      authoritative enable
       subnet 172.16.117.0/24 {
         start 172.16.117.10 {
            stop 172.16.117.20
            }
        }
    }
static-mapping MAP1 {
         ip-address 172.16.117.15
        mac-address 09:09:09:09:09:09
        static-mapping-parameters “option-rfc3422-
static-route 01:01:01:01:01:01:02”
}
```

DHCP Commands

release dhcp interface <interface>

Releases the current DHCP client lease from an interface.

Syntax **release dhcp interface** *interface*

Parameters *interface*

An interface that uses DHCP to obtain an IP address.

Modes Operational mode

Usage Guidelines Use this command to release the current DHCP client lease from an interface. The interface must be a DHCP client that obtained an IP address from a DHCP server.

renew dhcp interface <interface>

Renews the current DHCP client lease on an interface.

Syntax **renew dhcp interface** *interface*

Parameters *interface*

An interface that uses DHCP to obtain an IP address.

Modes Operational mode

Usage Guidelines Use this command to renew the current DHCP client lease on an interface. The interface must be a DHCP client that obtained an IP address from a DHCP server.

reset dhcp server lease ip <address>

Removes the DHCP lease for an IP address.

Syntax **reset dhcp server lease ip** [*ipv4-address* | *ipv6-address*]

Parameters *ipv4*

An IPv4 address.

ipv6

An IPv6 address.

Modes Operational mode

Usage Guidelines Use this command to remove the DHCP lease for an IP address. The command applies to leases provided by the DHCP server. The server is configured by using service dhcp-server on page 43.

reset dhcp server leases

Removes all DHCP leases.

Syntax **reset dhcp server leases**

Modes Operational mode

Usage Guidelines Use this command to remove all DHCP leases. The command applies to leases provided by the DHCP server. The server is configured by using service dhcp-server on page 43.

restart dhcp relay-agent

Restarts the DHCP relay agent.

Syntax **restart dhcp relay-agent**

Modes Operational mode

Usage Guidelines Use this command to stop the DHCP relay agent if it is running, then start it if it is configured. This command can be used if the DHCP relay agent is not operating properly.

restart dhcp server

Restarts the DHCP server.

Syntax **restart dhcp server**

Modes Operational mode

Usage Guidelines Use this command to stop and restart the DHCP server. This command can be used if the DHCP relay agent is not operating properly.

service dhcp-relay

Configures the system to relay DHCP client messages to an off-network DHCP server.

Syntax

set service dhcp-relay

delete service dhcp-relay

show service dhcp-relay

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-relay {
    }
}
```

Usage Guidelines

Use this command to configure the system as a DHCP relay agent.

A DHCP relay agent receives DHCP packets from DHCP clients and forwards them to a DHCP server. This allows you to place DHCP clients and DHCP servers on different networks; that is, across router interfaces.

The relay agent is configured with addresses of DHCP servers to which they should relay client DHCP messages. The relay agent intercepts the broadcast, sets the gateway address (the **giaddr** field of the DHCP packet) and, if configured, inserts the Relay Agent Information option (option 82) in the packet and forwards it to the DHCP server.

The DHCP server echoes the option back verbatim to the relay agent in server-to-client replies, and the relay agent strips the option before forwarding the reply to the client.

All interfaces involved in the DHCP relay for both clients and servers must be explicitly defined by using service dhcp-relay listen-interface <interface> on page 38.

Use the **set** form of this command to define DHCP relay configuration.

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

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

service dhcp-relay listen-interface <interface>

Enables DHCP relay on an interface for receiving DHCP requests from DHCP clients.

Syntax

set service dhcp-relay listen-interface *dp-interface*

delete service dhcp-relay listen-interface *dp-interface*

show service [**dhcp-relay listen-interface**]]

Parameters

interface

A dataplane interface on the router. At least one interface must be specified.

You can assign multiple interfaces to be used for DHCP by creating multiple **listen-interface** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-relay {
        listen-interface interface
    }
}
```

Usage Guidelines

Use this command to enable DHCP-relay on an interface to receive DHCP requests from DHCP clients.

At least one DHCP-relay server must be configured.

Use the **set** form of this command to specify the interface.

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

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

service dhcp-relay relay-options

Specifies whether to add the Relay Agent Information option (option 82) to the client-to-server packet.

Syntax

set service dhcp-relay relay-options [**hop-count** *count* | **max-size** *size* | **port** *port* | **relay-agents-packets** *policy*]

delete service dhcp-relay relay-options [**hop-count** | **max-size** | **port** | **relay-agents-packets**]

show service dhcp-relay relay-options [**hop-count** | **max-size** | **port** | **relay-agents-packets**]

Parameters

hop-count *count*

Optional. Sets the hop count for outgoing relayed messages. After the hop count is reached, the packet is discarded. The hop count should be set high enough that relayed packets are able to reach the DHCP server. The count ranges from 0 through 255. The default count is 10.

max-size *size*

Optional. Sets the maximum size of the DHCP packet to be created after appending the relay agent information option. If, after appending the information, the packet exceeds this size, the packet is forwarded without appending the information. This size should be set to the lowest MTU size in your network. The size ranges from 64 through1400. The default size is 576.

If this option is not configured, the router does not forward DHCP packets that exceed the MTU of the interface on which relaying is configured.

port *port*

Optional. Specifies the port on this interface that relays DHCP client messages. This should be done only for debugging because the behavior changes; responses are broadcast rather than being sent to port 68 of the requesting client. The port ranges from 1 through 65535.

relay-agents-packet *policy*

Optional. Sets the reforwarding policy for a DHCP relay agent. The router takes this action if the DHCP message already contains relay information. The policy is one of the following:

append: The DHCP relay agent may append its own set of relay options to the packet, leaving the supplied option field intact.

discard: If the packet already contains relay information, it is discarded.

forward: The packet is forwarded regardless of whether it contains relay information.

replace: The DHCP relay agent may replace the supplied option field with its own set of relay options.

The default policy is **forward**.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-relay {
        relay-options {
            hop-count count
            max-size size
            port port
            relay-agents-packets policy
        }
        }
}
```

Usage Guidelines

Use this command to configure the Relay Agent Information option (option 82) in the client-to-server packet, as specified by RFC 3046, and configure DHCP relay options.

Setting the port to a value other than 67 should be done only for debugging. When this is done, DHCP requests from clients are still accepted on port 67, but the responses from DHCP servers are forwarded to broadcast address 255.255.255.255 port 0 rather than on port 68, where DHCP clients listen.

Use the **set** form of this command to set DHCP relay options.

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

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

service dhcp-relay server <ipv4>

Specifies the IP address of a DHCP server.

Syntax

set service dhcp-relay server *ipv4*

delete service dhcp-relay server *ipv4*

show service dhcp-relay server

Parameters

ipv4

Mandatory. Multinode. The IP address of a DHCP server.

You can relay messages to more than one DHCP server by creating multiple **server** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-relay {
        server ipv4 {
        }
    }
}
```

Usage Guidelines

Use this command to specify the IP address of a DHCP server.

Use the **set** form of this command to specify the IP address of a DHCP server in a DHCP relay configuration.

Use the **delete** form of this command to remove DHCP server configuration in a DHCP relay configuration.

Use the **show** form of this command to view DHCP server configuration in a DHCP relay configuration.

service dhcp-relay upstream-interface <interface>

Specifies an interface for forwarding DHCP requests to the DCHP server.

Syntax

set service dhcp-relay upstream-interface *dp-interface*

delete service dhcp-relay upstream-interface *interface*

show service [**dhcp-relay upstream-interface**]]

Parameters

interface

A dataplane interface to forward DHCP requests. At least one interface must be specified.

You can assign multiple interfaces to be used for DHCP forwarding by creating multiple **upstream-interface** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-relay {
        upstream-interface interface
    }
}
```

Usage Guidelines

Use this command to specify an interface for forwarding DHCP requests to the DHCP server.

At least one DHCP-relay server must be configured.

Use the **set** form of this command to specify an interface for forwarding DHCP requests to the DHCP server.

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

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

service dhcp-server

Enables DHCP server functionality.

Syntax

set service dhcp-server

delete service dhcp-server

show service dhcp-server

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
    }
}
```

Usage Guidelines

Use this command to configure a pool of addresses the system can use for DHCP.

At least one address pool must be configured for DHCP to be available as a service.

At least one address pool must lie within a configured subnet on any of the broadcast interfaces.

Each subnet that is specified contains a distinct address pool. A given interface can support more than one address pool (that is, more than one subnet).

Use the **set** form of this command to enable DHCP server functionality.

Use the **delete** form of this command to remove DHCP server functionality.

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

service dhcp-server disabled <state>

Disables the DHCP server without discarding configuration.

Syntax

set service dhcp-server disabled *state*

delete service dhcp-server disabled

show service dhcp-server disabled

Command Default

DHCP server functionality is disabled.

Parameters

state

The administrative state of the DHCP server. The state is either of the following:

true: Disables the DHCP server without discarding configuration.

false: Enables the DHCP server.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        disabled state
    }
}
```

Usage Guidelines

Use this command to disable the DHCP server without discarding configuration.

Use the **set** form of this command to specify whether the DHCP server should be disabled.

Use the **delete** form of this command to restore the default state, that is, DHCP server functionality is disabled.

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

service dhcp-server dynamic-dns-update enable <state>

Specifies whether to dynamically update DNS.

Syntax

set service dhcp-server dynamic-dns-update enable *state*

delete service dhcp-server dynamic-dns-update enable

show service dhcp-server dynamic-dns-update enable

Command Default

DNS updates are not sent by the DHCP server.

Parameters

state

The state of dynamic DNS updates. The state is either of the following:

true: Sends updates dynamically.

false: Does not send updates.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        dynamic-dns-update {
            enable state
        }
    }
}
```

Usage Guidelines

Use this command to control DNS updates from the DHCP server.

Use the **set** form of this command to specify whether dynamic DNS updates should be sent.

Use the **delete** form of this command to restore the default state, that is, DNS updates are not sent.

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

service dhcp-server global-parameters <params>

Specifies additional global DHCP server parameters.

Syntax

set service dhcp-server global-parameters *params*

delete service dhcp-server global-parameters *params*

show service dhcp-server global-parameters

Parameters

params

A string of parameters to be used by the DHCP server. The string must be enclosed in single quotation marks (‘).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        global-parameters params
    }
}
```

Usage Guidelines

DANGER

This feature is advanced and should be used by only expert users in special situations.

Use this command to specify additional global DHCP server parameters that are not available with the **service dhcp-server** commands. The Brocade vRouter DHCP server commands are a subset of those that are available for DHCP server configuration. This command provides access to all DHCP server configuration parameters. More information regarding DHCP server configuration is located on the dhcpd.conf man page. To access the page, type the following at the Vyatta command prompt:

man dhcpd.conf

The Brocade vRouter does no validation before passing the parameter string to the DHCP server process (dhcpd). Because of this nonvalidation, it is imperative that the syntax described in the dhcpd.conf documentation be strictly followed. Failure to do so could result in a failure of the DHCP server. It is advisable to check the system log for errors when using these parameter strings. In addition, the **show system processes** command determines if the dhcpd process is still running.

The scope of these parameters is global. They apply to all shared-networks, subnets, and static-mappings unless parameters with a narrower scope are specified by using the **shared-network-parameters**, **subnet-parameters**, or **static-mapping-parameters** version of this command.

Multiple parameter strings can be specified. Each parameter string that is specified adds a separate line to the dhcpd.conf file.

Use the **set** form of this command to specify additional global DHCP server parameters.

Use the **delete** form of this command to remove additional global DHCP server parameters.

Use the **show** form of this command to display additional global DHCP server parameters.

service dhcp-server listento interface <dp-interface>

Allows the DHCP server to create address pools for clients that are indirectly connected to a dataplane network interface through a DHCP relay server.

Syntax

set service dhcp-server listento interface *dp-interface*

delete service dhcp-server listento interface

show service dhcp-server

Parameters

dp-interface

A dataplane interface on the router. It must have a valid IP address.

Modes

Configuration mode.

Configuration Statement

```
service {
   dhcp-server {
      listento {
         interface dp-interface
      }
   }
}
```

Usage Guidelines

Use this command to enable the DHCP server to create IP address pools for clients that are not directly connected to the router. For example, if clients on B subnet connect to the router through a DHCP relay server, the DHCP relay server connects to the router through a dataplane interface on the A subnet, and the dataplane interface has a valid IP address, using this command allows the DHCP server to create IP address pools for clients that are on the B subnet.

Use the **set** form of this command to create an IP address pool for clients that are indirectly connected to the router through a dataplane network interface.

Use the **delete** form of this command to remove a dataplane interface from the DHCP server configuration. If no dataplane interfaces are configured, the DHCP server cannot create address pools.

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

service dhcp-server shared-network-name <name>

Specifies the name for a DHCP address pool.

Syntax

set service dhcp-server shared-network-name *name*

delete service dhcp-server shared-network-name *name*

show service dhcp-server shared-network-name *name*

Parameters

name

Mandatory. Multinode. The name for a DHCP address pool.

You can define multiple address pools by creating multiple **shared-network-name** configuration nodes, each with a different name.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
        }
    }
}
```

Usage Guidelines

Use this command to create a DHCP server address pool with the specified name.

Use the **set** form of this command to create a DHCP address pool.

Use the **delete** form of this command to remove a DHCP address pool.

Use the **show** form of this command to display a DHCP address pool.

service dhcp-server shared-network-name <name> authoritative <state>

Specifies whether the DHCP server is the authoritative server.

Syntax

set service dhcp-server shared-network-name *name* **authoritative** *state*

delete service dhcp-server shared-network-name *name* **authoritative**

show service dhcp-server shared-network-name *name* **authoritative**

Command Default

The DHCP server is not authoritative.

Parameters

name

Mandatory. A DHCP address pool.

state

Specifies whether the DHCP server is the authoritative server. The state is either of the following:

enable Enables authoritative state.

disable Disables authoritative state.

The default state is **disable**.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            authoritative state
        }
    }
}
```

Usage Guidelines

Use this command to set the DHCP server as the authoritative server.

Setting the server as authoritative sets the server as a master server and allows it to protect itself from rogue DHCP servers or misconfigured DHCP clients. If the server is authoritative, it sends a DHCPNAK to a misconfigured client; otherwise, the client cannot update its IP address until after the old lease expires.

Use the **set** form of this command to enable or disable the authoritative state for the DHCP server.

Use the **delete** form of this command to restore the default authoritative state, which is not authoritative.

Use the **show** form of this command to display whether the DHCP server is authoritative.

service dhcp-server shared-network-name <name> description <desc>

Provides a description of a shared network.

Syntax

set service dhcp-server shared-network-name *name* **description** *desc*

delete service dhcp-server shared-network-name *name* **description**

show service dhcp-server shared-network-name *name* **description**

Parameters

name

Mandatory. A DHCP address pool.

desc

A description of a shared network.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            description desc
        }
    }
}
```

Usage Guidelines

Use this command to provide a description of a shared network.

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

Use the **delete** form of this command to delete the description of a shared network.

Use the **show** form of this command to display the description of a shared network.

service dhcp-server shared-network-name <name> disable

Disables DHCP configuration for a shared network.

Syntax

set service dhcp-server shared-network-name *name* **disable**

delete service dhcp-server shared-network-name *name* **disable**

show service dhcp-server shared-network-name *name*

Command Default

A shared-network configuration is enabled.

Parameters

name

Mandatory. A DHCP address pool.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            disable
        }
    }
}
```

Usage Guidelines

Use this command to disable DHCP configuration of a shared network.

Use the **set** form of this command to disable DHCP configuration of a shared network.

Use the **delete** form of this command to enable DHCP configuration of a shared network.

Use the **show** form of this command to display DHCP configuration of a shared network.

service dhcp-server shared-network-name <name> shared-network-parameters <params>

Specifies additional shared-network DHCP server parameters.

Syntax

set service dhcp-server shared-network-name *name* **shared-network-parameters** *params*

delete service dhcp-server shared-network-name *name* **shared-network-parameters** *params*

show service dhcp-server shared-network-name *name* **shared-network-parameters**

Parameters

name

Mandatory. A DHCP address pool.

params

A string of parameters to be used by the DHCP server. The string must be enclosed in single quotation marks (’).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            shared-network-parameters params
        }
    }
}
```

Usage Guidelines

NOTE

This feature is advanced and should be used by only expert users in special situations.

Use this command to specify additional shared-network DHCP server parameters that are not available with the **service dhcp-server** commands. The Brocade vRouter DHCP server commands are a subset of those that are available for DHCP server configuration. This command provides access to all DHCP server configuration parameters. More information regarding DHCP server configuration is located on the dhcpd.conf man page. To access the page, type the following at the Vyatta command prompt:

man dhcpd.conf

The Brocade vRouter does no validation before passing the parameter string to the DHCP server process (dhcpd). Because of this nonvalidation, it is imperative that the syntax described in the dhcpd.conf documentation be strictly followed. Failure to do so could result in a failure of the DHCP server. It is advisable to check the system log for errors when using these parameter strings. In addition, the **show system processes** command determines if the dhcpd process is still running.

The scope of these parameters is for the specified shared network. They apply to all subnets, and static-mappings within this scope unless parameters with a narrower scope are specified by using the **subnet-parameters** or **static-mapping-parameters** version of this command.

Multiple parameter strings can be specified. Each parameter string that is specified adds a separate line to the dhcpd.conf file.

Use the **set** form of this command to specify additional shared-network DHCP server parameters.

Use the **delete** form of this command to remove additional shared-network DHCP server parameters.

Use the **show** form of this command to display additional shared-network DHCP server parameters.

service dhcp-server shared-network-name <name> subnet <ipv4net>

Specifies the IPv4 network to be served by a DHCP address pool.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net*

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network to be served with the addresses defined in the specified address pool. The format of the network designation is *ip-addr*/*prefix*.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the IPv4 network to be served with the addresses that are defined in this named rule. DHCP requests from devices on this subnet are served static address assignments or an address from the defined range.

Use the **set** form of this command to specify the DHCP address pool subnet.

Use the **delete** form of this command to remove DHCP address pool subnet configuration.

Use the **show** form of this command to view DHCP address pool subnet configuration.

service dhcp-server shared-network-name <name> subnet <ipv4net> bootfile-name <bootfile>

Specifies a bootstrap file from which diskless PCs can boot.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-name** *bootfile*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-name**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-name**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

bootfile

The name of a bootstrap file to be used to boot.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                bootfile-name bootfile
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a bootstrap file from which diskless PCs can boot.

Use the **set** form of this command to specify a bootstrap file.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> bootfile-server <addr>

Specifies a bootstrap server from which diskless PCs can boot.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-server** *addr*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-server**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **bootfile-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

addr

The IPv4 address or host name of the bootfile server.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                bootfile-server addr
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a bootstrap server from which diskless PCs can boot.

Use the **set** form of this command to specify a bootstrap server.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> client-prefix-length <prefix>

Specifies the length of a subnet prefix to be assigned to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **client-prefix-length** *prefix*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **client-prefix-length**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **client-prefix-length**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

prefix

Optional. The length of a subnet prefix that is assigned to each client. By default, the prefix length defined in the **subnet** parameter is assigned. The prefix length ranges from 0 through 32.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                client-prefix-length prefix
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the length of a subnet prefix that is assigned to each client.

Use the **set** form of this command to specify the length of a prefix subnet that is assigned to each client.

Use the **delete** form of this command to delete the length of a subnet prefix.

Use the **show** form of this command to display the length of a subnet prefix.

service dhcp-server shared-network-name <name> subnet <ipv4net> ping-check

Pings the IP address of the shared network subnet to confirm if the address is not configured on another node. Ping-check is valid only for IPv4.

Syntax

set service dhcp-server shared-network-name*name* **subnet** *ipv4net* **ping-check**

delete service dhcp-server shared-network-name*name* **subnet** *ipv4net* **ping-check**

show service dhcp-server shared-network-name*name* **subnet** *ipv4net* **ping-check**

Parameters

name

Mandatory. A DHCP address pool.

You can define multiple address pools by creating multiple **shared-network-name** configuration nodes, each with a different name.

ipv4net

Mandatory. Multinode. The IPv4 network to be served with the addresses defined in the specified address pool. The format of the network designation is *ip-addr*/*prefix*.

Modes

Configuration mode

Configuration Statement

```
service {
   dhcp-server {
       shared-network-name name {
          subnet ipv4net {
             ping-check {

             }
          }
       }
   }
}
```

Usage Guidelines

Use this command to ping the IP address of the shared network subnet to confirm if the address is not configured on another node. Ping-check is valid only for IPv4.

Use the **set** form of this command to specify the IPv4 subnet for ping-check.

Use the **delete** form of this command to remove the IPv4 subnet for ping-check.

Use the **show** form of this command to view the IPv4 subnet for ping-check.

Examples

The following example shows how to use the ping-check option for a shared network called `foo` and the IPv4 subnet address of `12.1.1.0/24`. This option generates an ICMP echo request before offering an address to the client.

```
vyatta@vyatta# set service dhcp-server shared-network-name foo subnet 12.1.1.0/24
ping-check
```

service dhcp-server shared-network-name <name> subnet <ipv4net> default-router <ipv4>

Specifies the address of the default router for DHCP clients on a subnet.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **default-router** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **default-router**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **default-router**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

Optional. The address of the default router for DHCP clients on this subnet. The default router should be on the same subnet as the client. The format is an IP address.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                default-router ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of the default router for DHCP clients on a subnet.

Use the **set** form of this command to specify the address of the default router for DHCP clients on a subnet.

Use the **delete** form of this command to remove the default-router configuration.

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

service dhcp-server shared-network-name <name> subnet <ipv4net> dns-server <ipv4>

Specifies the address of a DNS server for DHCP clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **dns-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **dns-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **dns-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

ipv4

Optional. Multinode. The IPv4 address of a DNS server.

You can specify more than one DNS server by entering this command multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                dns-server ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of a DNS server that is available to DHCP clients.

Use the **set** form of this command to specify the address of a DNS server.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> domain-name <domain-name>

Provides the domain name for DHCP clients on a subnet.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **domain-name** *domain-name*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **domain-name**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **domain-name**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

domain-name

Optional. The domain name to be given to DHCP clients on this subnet. A domain name can include letters, numbers, hyphens (-), and one period (.). For example, brocade.com.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                domain-name domain-name
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the domain name for DHCP clients on a subnet.

Use the **set** form of this command to specify the client domain name.

Use the **delete** form of this command to remove client domain name configuration.

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

service dhcp-server shared-network-name <name> subnet <ipv4net> exclude <ipv4>

Excludes an IP address from a DHCP address pool.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **exclude** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **exclude** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **exclude**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

Optional. Multinode. An IP address to exclude from the lease range.

You can exclude more than one IP address by creating multiple **exclude** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                exclude ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to exclude an IP address from a DHCP address pool. An excluded address is never leased to DHCP clients. The exception is an IP address that is statically mapped by using service dhcp-server static-mapping mapname on page 85. This address is not excluded.

Use the **set** form of this command to exclude an IP address from the lease range.

Use the **delete** form of this command to delete an IP address from the list of excluded addresses.

Use the **show** form of this command to display excluded IP addresses.

service dhcp-server shared-network-name <name> subnet <ipv4net> failover

Enables DHCP failover functionality for a DHCP address pool on a subnet.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover**

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                failover {
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to enable DHCP failover for an address pool on a given network, allowing two DHCP servers to share an address pool.

In a failover configuration, two DHCP servers act as failover peers, with one of the peers designated as the primary and the other as the secondary. For DHCP failover to work, the following conditions must be met.

- Both peers must be Brocade vRouters and must be running the same version of Brocade vRouter software.
- Each server must be configured to point to the other as the failover peer.
- The time on the servers must be exactly synchronized.
- At least one IP address must exist in the start-stop range for each subnet that has not been either excluded (by using service dhcp-server shared-network-name name subnet ipv4net exclude ipv4 on page 62) or statically mapped (by using service dhcp-server static-mapping mapname on page 85).

The system times should be synchronized before configuring DHCP failover. Use of NTP time synchronization is highly recommended. However, if difficulties arise because of incorrect system times, then disable NTP, reset the times correctly, and re-enable NTP.

Note that DHCP leases are assigned only in failover configurations if proper communication is established between the two failover peers. If the configuration is incorrect (if, for example, one failover peer is configured but the other is not), DHCP leases are not dispersed.

Note also that statically mapped addresses are not renewed by a failover server unless they are explicitly defined on that server by using service dhcp-server static-mapping mapname on page 85.

Use the **set** form of this command to define DHCP failover configuration.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> failover local-address <ipv4>

Specifies the IP address of the local failover peer.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover local-address** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover local-address**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover local-address**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

The IP address of the local failover peer.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                failover {
                    local-address ipv4
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the IP address of the local failover peer.

Use the **set** form of this command to specify the IP address of the local failover peer.

Use the **delete** form of this command to remove local failover IP address configuration.

Use the **show** form of this command to view local failover IP address configuration.

service dhcp-server shared-network-name <name> subnet <ipv4net> failover name <peer-name>

Specifies the DHCP failover IP address for the local peer.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover name** *peer-name*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover name**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover name**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

peer-name

The DHCP failover peer name for the local peer.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                failover {
                    name peer-name
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a name for the local peer in a DHCP failover pair.

Use the **set** form of this command to specify the DHCP failover peer name.

Use the **delete** form of this command to remove the local peer name configuration.

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

service dhcp-server shared-network-name <name> subnet <ipv4net> failover peer-address <ipv4>

Specifies the IP address of the local failover peer.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover peer-address** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover peer-address**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover peer-address**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

The IP address of the local failover peer.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                failover {
                    peer-address ipv4
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the IP address of the local failover peer.

Use the **set** form of this command to specify the IP address of the local failover peer.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> failover status <status>

Specifies the DHCP failover status of the local system.

Syntax **set service dhcp-server shared-network-name** *name* **subnet** *ipv4net* **failover status** *status*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover status**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **failover status**

Parameters *name*

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

status

The DHCP failover status of a peer in the failover configuration. The status is either of the following:

- **primary**—Indicates the local system is the primary peer.
- **secondary**—Indicates the local system is the secondary peer.

Modes Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                failover {
                    status status
                }
            }
        }
    }
}
```

Usage Guidelines Use this command to specify the DHCP failover status of the local system.

Use the **set** form of this command to specify the DHCP failover status as primary or secondary.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> ip-forwarding enable <state>

Specifies whether a client configures its IP layer for packet forwarding.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ip-forwarding enable** *state*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ip-forwarding enable**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ip-forwarding enable**

Command Default

The DHCP server does not direct a client to configure its IP layer for packet forwarding.

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

state

Whether a client configures its IP layer for packet forwarding. The state is either of the following:

- **true** —Indicates that a client does configure its IP layer for packet forwarding.
- **false** —Indicates that a client does not configure its IP layer for packet forwarding.

The default state is **false**.

Modes

Configuration mode

Configuration Statement

```
service {
      dhcp-server {
            shared-network-name name {
                  subnet ipv4net {
                        ip-forwarding {
                              enable state
                        }
                  }
            }
      }
}
```

Usage Guidelines

Use this command to specify whether the DHCP server directs a client to configure its IP layer for packet forwarding.

Use the **set** form of this command to specify whether a client configures its IP layer for packet forwarding.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> lease <seconds>

Specifies how long the address assigned by the DHCP server is valid.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **lease** *seconds*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **lease**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **lease**

Command Default

The default number of seconds is 86,400 (24 hours).

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

seconds

Optional. The number of seconds the address that is assigned by the DHCP server is valid. The number of seconds ranges from 120 through 4294967296.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                lease seconds
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify how long the address assigned by the DHCP server is valid.

Use the **set** form of this command to specify how long the address assigned by the DHCP server is valid.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> ntp-server <ipv4>

Specifies the address of a Network Time Protocol (NTP) server that is available to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ntp-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ntp-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **ntp-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

Optional. The IP address of an NTP server. Multiple NTP server addresses can be specified in separate commands. The NTP servers should be specified in a preferred order.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                ntp-server ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of an NTP server that is available to clients.

Use the **set** form of this command to specify the address of an NTP server.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> pop-server <ipv4>

Specifies the address of a Post Office Protocol 3 (POP3) server that is available to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **pop-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **pop-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **pop-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

ipv4

Optional. The IP address of a POP3 server. Multiple POP3 server addresses can be specified in separate commands. The POP3 servers should be specified in a preferred order.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                pop-server ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of a POP3 server that is available to clients.

Use the **set** form of this command to specify the address of a POP3 server.

Use the **delete** form of this command to remove the POP3 server configuration.

Use the **show** form of this command to view the POP3 server configuration.

service dhcp-server shared-network-name <name> subnet <ipv4net> server-identifier <ipv4>

Specifies the address for the DHCP server identifier.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **server-identifier** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **server-identifier**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **server-identifier**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

Optional. The address for the DHCP server identifier.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                server-identifier ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address for the DHCP server identifier.

The server identifier is a field in a DHCP message that identifies the DHCP server as the destination address from clients to servers. When the DHCP server includes this field in a DHCPOffer, a client uses it to distinguish between multiple lease offers. The server identifier must be an address that can be reached from the client.

Use the **set** form of this command to specify the address for the DHCP server identifier.

Use the **delete** form of this command to remove the address for the DHCP server identifier.

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

service dhcp-server shared-network-name <name> subnet <ipv4net> smtp-server <ipv4>

Specifies the address of a Simple Mail Transfer Protocol (SMTP) server that is available to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **smtp-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **smtp-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **smtp-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

ipv4

Optional. The IP address of an SMTP server. Multiple SMTP server addresses can be specified in separate commands. The SMTP servers should be specified in a preferred order.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                smtp-server ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of an SMTP server that is available to clients.

Use the **set** form of this command to specify the address of an SMTP server.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> start <ipv4> stop <ipv4>

Specifies the range of addresses that are assigned to DHCP clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **start** *ipv4* **stop** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **start** [*ipv4* [**stop**]]

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **start** [*ipv4*]

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

start

Optional. Multinode. The beginning address in a range of addresses. This address is the first address in the range that can be assigned.

You can define multiple address ranges within an address pool by creating multiple **start** configuration nodes.

stop

Mandatory. The ending address in this range of addresses. This address is the last address in the range that can be assigned.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                start ipv4 {
                    stop ipv4
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the range of addresses that are assigned to DHCP clients.

Use the **set** form of this command to specify the range of addresses that are assigned to DHCP clients.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> static-route destination-subnet <ipv4net>

Specifies the destination subnet of a static route for clients to store in their routing cache.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route destination-subnet** *ipv4net2*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route destination-subnet**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route destination-subnet**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

ipv4net2

The destination IP subnet of a static route for clients to store in their routing table.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                static-route {
                    destination-subnet ipv4net2
                }
            }
        }
}
```

Usage Guidelines

Use this command to specify the destination subnet of a static route for clients to store in their routing cache. The other part of the static route is defined by using service dhcp-server shared-network-name name subnet ipv4net static-route router ipv4 on page 77. Only one static route can be defined for a given subnet.

Use the **set** form of this command to specify the destination subnet of a static route for clients to store in their routing cache.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> static-route router <ipv4>

Specifies the router for the destination of a static route that clients can store in their routing cache.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route router** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route router**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **static-route router**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

The IP address of the router for the destination of a static route for clients to store in their routing cache.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                static-route {
                    router ipv4
                }
            }
        }
}
```

Usage Guidelines

Use this command to specify the router for the destination of a static route that clients can store in their routing cache. The other part of the static route is defined by using service dhcp-server shared-network-name <name> subnet <ipv4net> static-route destination-subnet <ipv4net> on page 76.

Use the **set** form of this command to specify the router for the destination of a static route that clients can store in their routing cache.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> subnet-parameters <params>

Specifies additional subnet parameters for a DHCP server.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **subnet-parameters** *params*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **subnet-parameters** *params*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **subnet-parameters**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

params

A string of parameters to be used by the DHCP server. The string must be enclosed in single quotation marks (’).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                subnet-parameters params
            }
        }
    }
}
```

Usage Guidelines

DANGER

This feature is advanced and should be used by only expert users in special situations.

Use this command to specify additional subnet parameters for a DHCP server that are not available with the **service dhcp-server** commands. The Brocade vRouter DHCP server commands are a subset of those that are available for DHCP server configuration. This command provides access to all DHCP server configuration parameters. More information regarding DHCP server configuration is located on the dhcpd.conf man page. To access the page, type the following at the Vyatta command prompt:

man dhcpd.conf

The Brocade vRouter does no validation before passing the parameter string to the DHCP server process (dhcpd). Because of this nonvalidation, it is imperative that the syntax described in the dhcpd.conf documentation be strictly followed. Failure to do so could result in a failure of the DHCP server. It is advisable to check the system log for errors when using these parameter strings. In addition, the **show system processes** command determines if the dhcpd process is still running.

The scope of these parameters is for the specified subnet. They apply to all **static-mappings** within this scope unless parameters with a narrower scope are specified by using the **static-mapping-parameters** version of this command.

Double quotation marks (“) cannot be used within a string enclosed in single quotation marks (‘). Instead, replace all double quotation marks with &quot;.

Multiple parameter strings can be specified. Each parameter string that is specified adds a separate line to the dhcpd.conf file.

Use the **set** form of this command to specify additional subnet parameters for a DHCP server.

Use the **delete** form of this command to delete additional subnet parameters from a DHCP server.

Use the **show** form of this command to display additional subnet parameters for a DHCP server.

service dhcp-server shared-network-name <name> subnet <ipv4net> tftp-server-name <servername>

Specifies the name of a Trivial File Transfer Protocol (TFTP) server that is available to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **tftp-server-name** *servername*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **tftp-server-name**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **tftp-server-name**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool.
The format is *ip-addr*/*prefix*.

servername

The name of a TFTP server that is available to clients.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                tftp-server-name servername
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the name of a TFTP server that is available to clients.

Use the **set** form of this command to specify the name of a TFTP server.

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

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

service dhcp-server shared-network-name <name> subnet <ipv4net> time-offset <seconds>

Specifies a time offset in seconds from Universal Time Coordinated (UTC) of the subnet of a client.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-offset** *seconds*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-offset**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-offset**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

seconds

Time offset in seconds from UTC of the subnet of a client.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                time-offset seconds
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a time offset in seconds from UTC of the subnet of a client.

Use the **set** form of this command to specify a time offset.

Use the **delete** form of this command to remove a time offset.

Use the **show** form of this command to display a time offset.

service dhcp-server shared-network-name <name> subnet <ipv4net> time-server <ipv4>

Specifies the address of an RFC868 time server that is available to clients.

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **time-server**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr*/*prefix*.

ipv4

Optional. The IP address of an RFC868 time server. Multiple time server addresses can be specified in separate commands. The time servers should be specified in a preferred order.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                time-server ipv4
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of an RFC 868 time server that is available to clients.

Use the **set** form of this command to specify the address of a time server.

Use the **delete** form of this command to delete the address of a time server.

Use the **show** form of this command to display the address of a time server.

service dhcp-server shared-network-name <name> subnet <ipv4net> wins-server <ipv4>

Specifies the address of a Windows Internet Naming Server (WINS) that is available to DHCP clients.

Syntax **set service dhcp-server shared-network-name** *name* **subnet** *ipv4net* **wins-server** *ipv4*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **wins-server** *ipv4*

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **wins-server**

Parameters *name*

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

ipv4

Optional. Multinode. The address of a WINS. The WINS provides name-resolution services that Microsoft DHCP clients can use to correlate host names to IP addresses.

You can specify more than one WINS by entering this command multiple times. The format is an IP address.

Modes Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                wins-server ipv4
            }
        }
    }
}
```

Usage Guidelines Use this command to specify the address of a WINS that is available to DHCP clients.

Use the **set** form of this command to specify the address of a WINS server.

Use the **delete** form of this command to delete the configuration of a WINS.

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

service dhcp-server shared-network-name <name> subnet <ipv4net> wpad-url <url>

Specifies the Web Proxy Autodiscovery (WPAD) URL

Syntax

set service dhcp-server shared-network-name *name* **subnet** *ipv4net* **wpad-url** *url*

delete service dhcp-server shared-network-name *name* **subnet** *ipv4net* **wpad-url**

show service dhcp-server shared-network-name *name* **subnet** *ipv4net* **wpad-url**

Parameters

name

Mandatory. A DHCP address pool.

ipv4net

Mandatory. Multinode. The IPv4 network served by the DHCP address pool. The format is *ip-addr/prefix*.

url

Optional. The WPAD URL.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv4net {
                wpad-url url
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the WPAD URL.

Use the **set** form of this command to specify the WPAD URL.

Use the **delete** form of this command to delete the WPAD URL.

Use the **show** form of this command to display the WPAD URL.

service dhcp-server static-mapping <mapname>

Statically maps a DHCP client, based on its MAC address, to an IP address.

Syntax

set service dhcp-server static-mapping *mapname*

delete service dhcp-server static-mapping *mapname*

show service dhcp-server static-mapping *mapname*

Parameters

mapname

Optional. Multinode. Allows you to statically map an IP address within an address pool to the MAC address of a device on the network.

You can define multiple static mappings of this type by creating multiple **static-mapping** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        static-mapping mapname {
        }
    }
}
```

Usage Guidelines

Use this command to statically map a DHCP client, based on its MAC address, to an IP address.

Use the **set** form of this command to statically map a DHCP client, based on its MAC address, to an IP address.

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

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

service dhcp-server static-mapping <mapname> disable

Disables DHCP configuration for a static mapping.

Syntax

set service dhcp-server static-mapping *mapname* **disable**

delete service dhcp-server static-mapping *mapname* **disable**

show service dhcp-server static-mapping *mapname*

Command Default

The static-mapping configuration is enabled.

Parameters

mapname

Optional. Multinode. Allows you to statically map an IP address within an address pool to the MAC address of a device on the network.

You can define multiple static mappings of this type by creating multiple **static-mapping** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        static-mapping mapname {
        disable
        }
    }
}
```

Usage Guidelines

Use this command to disable DHCP configuration of a static mapping.

Use the **set** form of this command to disable DHCP configuration of a static mapping.

Use the **delete** form of this command to enable configuration of a static mapping.

Use the **show** form of this command to display the configuration of a static mapping.

service dhcp-server static-mapping <mapname> ip-address <ipv4>

Specifies a static IP address for a DHCP client.

Syntax

set service dhcp-server static-mapping *mapname* **ip-address** *ipv4*

delete service dhcp-server static-mapping *mapname* **ip-address**

show service dhcp-server static-mapping *mapname* **ip-address**

Parameters

mapname

Multinode. Allows you to statically map an IP address within an address pool to the MAC address of a device on the network.

You can define multiple static mappings of this type by creating multiple **static-mapping** configuration nodes.

ipv4

Mandatory. The IP address to be statically assigned to the device.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        static-mapping mapname {
        ip-address ipv4
        }
    }
}
```

Usage Guidelines

Use this command to specify a static IP address for a DHCP client, based on its MAC address.

Use the **set** form of this command to specify a static IP address for a DHCP client, based on its MAC address.

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

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

service dhcp-server static-mapping <mapname> mac-address <mac>

Specifies the MAC address of a DHCP client to which an IP address is assigned.

Syntax

set service dhcp-server static-mapping *mapname* **mac-address** *mac*

delete service dhcp-server static-mapping *mapname* **mac-address**

show service dhcp-server static-mapping *mapname* **mac-address**

Parameters

mapname

Multinode. Allows you to statically map an IP address within an address pool to the MAC address of a device on the network.

You can define multiple static mappings of this type by creating multiple **static-mapping** configuration nodes.

mac

Mandatory. The MAC address to be statically mapped to the specified IP address.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        static-mapping mapname {
        mac-address mac
        }
    }
}
```

Usage Guidelines

Use this command to specify the MAC address of a DHCP client to which an IP address is assigned.

Use the **set** form of this command to specify the MAC address of the DHCP client to which an IP address is assigned.

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

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

service dhcp-server static-mapping <mapname> static-mapping-parameters <params>

Specifies additional static-mapping parameters for a DHCP server.

Syntax

set service dhcp-server static-mapping *mapname* **static-mapping-parameters** *params*

delete service dhcp-server static-mapping *mapname* **static-mapping-parameters** *params*

show service dhcp-server static-mapping *mapname* **static-mapping-parameters**

Parameters

mapname

Optional. Multinode. Allows you to statically map an IP address within an address pool to the MAC address of a device on the network.

You can define multiple static mappings of this type by creating multiple **static-mapping** configuration nodes.

params

A string of parameters to be used by the DHCP server. The string must be enclosed in single quotation marks (’).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        static-mapping mapname {
        static-mapping-parameters params
        }
    }
}
```

Usage Guidelines

NOTE

This feature is advanced and should be used by only expert users in special situations.

Use this command to specify additional static-mapping parameters for a DHCP server that are not available with the **service dhcp-server** commands. The Vyatta DHCP server commands are a subset of those that are available for DHCP server configuration. This command provides access to all DHCP server configuration parameters. More information regarding DHCP server configuration is located on the dhcpd.conf man page. To access the page, type the following at the Vyatta command prompt:

man dhcpd.conf

The Brocade vRouter does no validation before passing the parameter string to the DHCP server process (dhcpd). Because of this nonvalidation, it is imperative that the syntax described in the dhcpd.conf documentation be strictly followed. Failure to do so could result in a failure of the DHCP server. It is advisable to check the system log for errors when using these parameter strings. In addition, the **show system processes** command determines if the dhcpd process is still running.

The scope of these parameters is for the specified map name. They apply to all **static-mappings** within this scope unless parameters with a narrower scope are specified by using the **static-mapping-parameters** version of this command.

Multiple parameter strings can be specified. Each parameter string that is specified adds a separate line to the dhcpd.conf file.

Use the **set** form of this command to specify additional static-mapping parameters for a DHCP server.

Use the **delete** form of this command to remove additional static-mapping parameters from a DHCP server.

Use the **show** form of this command to display the additional static-mapping parameters for a DHCP server.

show dhcp client leases

Displays DHCP information for an interface that is configured as a DHCP client.

Syntax

show dhcp client leases [**interface** *interface*]

Parameters

interface

The identifier of an interface. Supported interface types are the following:

- **lo** : A loopback interface.
- **dp***x***p***y***p***z*—The name of a dataplane interface, where

  — **dp***x* specifies the dataplane identifier (ID). Currently, only dp0 is supported.

  — p*y* specifies a physical or virtual PCI slot index (for example, p129).

  — **p***z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- **dp***x***em***y* —The name of a dataplane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- **dp***x***s***y* —The name of a dataplane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.
- **dp***x***P***n***p***y***p***z* —The name of a dataplane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name dataplane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

Modes

Operational mode

Usage Guidelines

Use this command to display DHCP information for an interface that is configured as a DHCP client.

When used with no option, this command displays DHCP information for all interfaces that are configured as DHCP clients. When an interface is specified, this command displays DHCP information for that interface.

To configure an interface as a DHCP client, refer to documentation for that interface.

Examples

The following example shows how to display DHCP information for all interfaces that are configured as DHCP clients.

```
vyatta@R1> show dhcp client leases
interface  : dp0p1p1
ip address : 192.168.1.157      [Active]
subnet mask: 255.255.255.0
router     : 192.168.1.254
name server: 192.168.1.254 74.150.163.68 74.150.163.100
dhcp server: 192.168.1.254
lease time : 86400
last update: Wed Feb 17 02:18:20 GMT 2010
expiry     : Thu Feb 18 02:18:18 GMT 2010
reason     : BOUND
vyatta@R1>
```

show dhcp server leases

Displays current DHCP lease information.

Syntax **show dhcp server leases** [**expired** | **pool** *pool-name*]

Parameters **expired**

Displays expired leases.

pool *pool-name*

Displays lease information for the specified address pool.

Modes Operational mode

Usage Guidelines Use this command to display current DHCP lease information for subscribers or expired leases.

When used with no option, this command displays all current lease information. When an address pool is specified, this command displays lease information for that address pool. When the **expired** option is specified, only expired leases are displayed.

DHCP is configured by using service dhcp-server on page 43.

Examples The following example shows how to display all current DHCP lease information.

```
vyatta@R1> show dhcp server leases
IP address       Hardware Address   Lease expiration     Pool     Client Name
----------       ----------------   ----------------     ----     -----------
192.168.11.101   00:12:3f:e3:af:67  2007/06/23 16:28:26  POOL1    Laptop 9
vyatta@R1>
```

show dhcp server statistics

Displays DHCP server statistics.

Syntax

show dhcp server statistics [**pool** *pool-name*]

Parameters

pool *pool-name*

Displays DHCP statistics for the specified address pool

Modes

Operational mode

Usage Guidelines

Use this command to see current lease information for DHCP subscribers.

When used with no option, this command displays all current lease information. When address pool is provided, this command displays lease information for the specified address pool.

DHCP is configured by using service dhcp-server on page 43.

Examples

The following example shows how to display all DHCP server statistics.

```
vyatta@R1> show dhcp server statistics
pool                                    pool size   # leased    # avail
----                                    ---------   --------    -------
POOL1                                   100         1           99
vyatta@R1>
```

DHCPv6

DHCPv6 overview

In general, Dynamic Host Configuration Protocol (DHCP) allows dynamic assignment of reusable IP addresses and other configuration information to DHCP clients. DHCP is described in DNS on page 139. DHCPv6 provides a stateful address auto-configuration service and a stateful auto-configuration protocol for the IPv6 suite of protocols.

Although it bears many features in common with DHCP and shares a common architectural model, DHCPv6 is a separate protocol and is configured independently of DHCP. It is defined in separate protocol specification documents and the functions it provides differ in significant ways from those provided by DHCP. For example, DHCP and DHCPv6 use different UDP port numbers and they provide different sets of configuration parameters.

The Brocade vRouter provides DHCPv6 server functionality, DHCPv6 client-side functionality (currently only available on dataplane interfaces), and a DHCPv6 relay function.

There are two common usage scenarios for DHCPv6 server. The first is one in which addresses are assigned by using SLAAC and the DHCPv6 server is used only to assign parameters to the clients. The second is one in which both addresses and parameters are supplied by the DHCPv6 server. In either case, default router discovery is provided by the Neighbor Discovery (ND) protocol and so the DHCPv6 server does not need to provide that parameter.

DHCPv6 configuration

DHCPv6 configuration includes enabling the DHCPv6 server, creating a static mapping, configuring a DHCPv6 address pool, setting up a DHCPv6 relay, and setting up a DHCPv6 client.

Enabling the DHCPv6 server

To use the DHCPv6 server on the Brocade vRouter, you must enable the DHCPv6 service. To enable the DHCPv6 service, perform the following steps in configuration mode.

TABLE 9 Enabling the DHCPv6 service

Step	Command
Enable the DHCPv6 server.	`vyatta@R1# set service dhcpv6-server`
Commit the information.	`vyatta@R1# commit`

TABLE 9 Enabling the DHCPv6 service (Continued)

Show the configuration.	`vyatta@R1# show service`
	`dhcpv6-server {`
	`}`

Creating a static mapping

Situations exist in which it makes sense to map a specific IPv6 address to a specific host rather than dynamically assign an IP address from a pool of addresses. This mapping is known as a static mapping.

A static mapping is defined by using the **static-mapping** option of the **service dhcp-server** configuration node. The following example shows how to create a static mapping.

Table 12 on page 98 shows how to map the 2001:db8:100::101 IP address to the device with a MAC address of 00:15:c5:b3:2e:65.

TABLE 10 Creating a static mapping

Step	Command
Create a static mapping called "lab" and specify the static IP address.	`vyatta@R1#set service dhcpv6-server static-mapping lab ipv6-address 2001:db8:100::101`
Specify the host identifier string (c5b32e65—low order 4 bytes of MAC address) within the static mapping called "lab".	`vyatta@R1#set service dhcpv6-server static-mapping lab identifier c5b32e65`
Commit the information.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service dhcp-server shared-network-name LAB-NET`
	`static-mapping lab`
	`{`
	`ipv6-address 2001:db8:100::101`
	`identifier c5b32e65`
	`}`

Configuring DHCPv6 address pools

Configure DHCPv6 address pools if you want the system to act as a DHCPv6 server for the network.

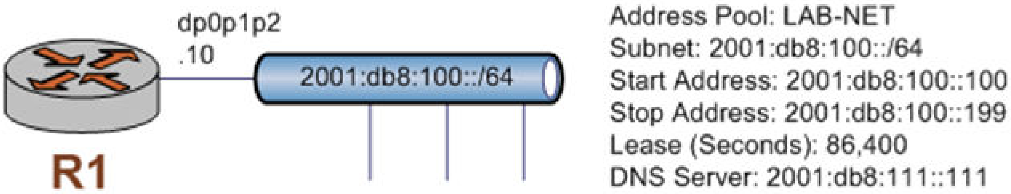
Configuring for networks directly connected to the system

The following example shows how to create an address pool within the LAB-NET shared network.

LAB-NET. This shared network serves the 2001:db8:100::/64 subnet, which is connected directly to the dp0p1p2 interface. The lease time remains at the default, 24 hours (86,400 seconds). The address pool uses the DNS name server at 2001:db8:111::111, which is on a separate subnet (not shown). The range of addresses is configured for .100 through .199.

Figure 5 shows the sample address pool configuration.

FIGURE 5 DHCPv6 address pool configuration


dp0p1p2
.10
2001:db8:100::/64
R1
Address Pool: LAB-NET
Subnet: 2001:db8:100::/64
Start Address: 2001:db8:100::100
Stop Address: 2001:db8:100::199
Lease (Seconds): 86,400
DNS Server: 2001:db8:111::111

To configure the DHCPv6 address pool, perform the following steps in configuration mode.

TABLE 11 Configuring a DHCPv6 address pool

Step	Command
Create the configuration node for LAB-NET on the 2001:db8:100::/64 subnet. Specify the start and stop IPv6 addresses for the pool.	`vyatta@R1# set service dhcpv6-server shared-network-name LAB-NET subnet 2001:db8:100::/64 address-range start 2001:db8:100::100 stop 2001:db8:100::199`
Specify a DNS server for LAB-NET.	`vyatta@R1# set service dhcp-server shared-network-name LAB-NET subnet 2001:db8:100::/64 name-server 2001:db8:111::111`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	
	```
	vyatta@R1# show service dhcpv6-server
	   shared-network-name LAB-NET {
	           subnet 2001:db8:100::/64 {
	               address-range {
	                   start 2001:db8:100::100 {
	                       stop 2001:db8:100::199
	                   }
	               }
	               name-server 2001:db8:111::111
	           }
	   }
	```

Show the interface configuration.	
	```
	vyatta@R1# show interfaces
	   dataplane dp0p1p2 {
	     address 2001:db8:100::10/64
	     hw-id 00:0c:29:42:05:35
	   }
	```

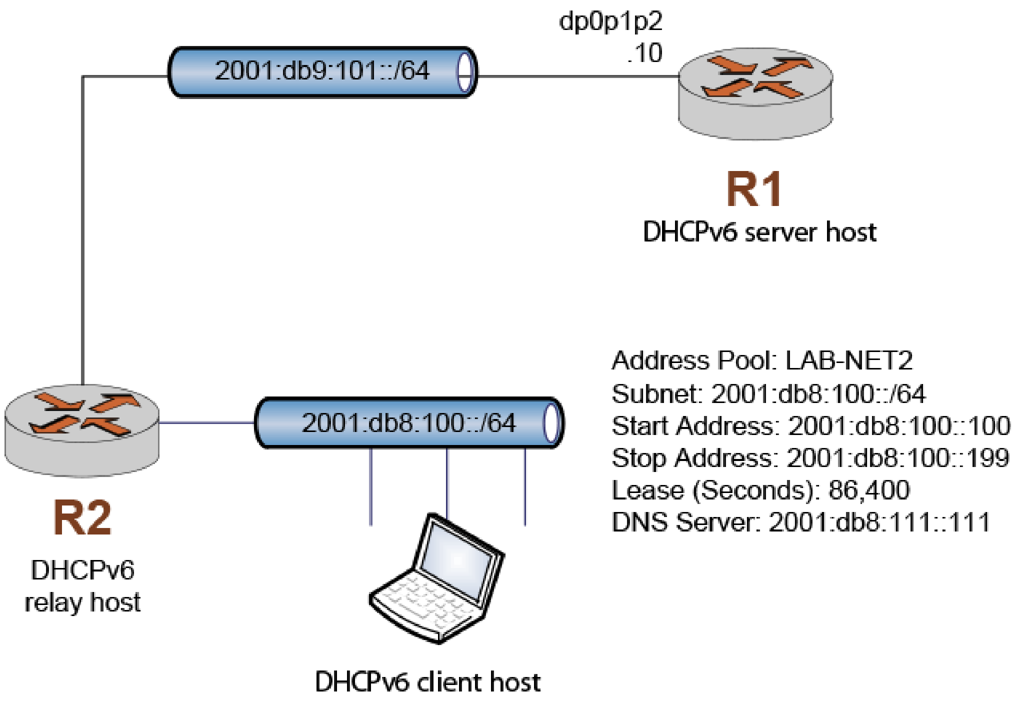
Configuring for networks indirectly connected to the system

The following example shows how to create an address pool within the LAB-NET2 shared network, which is indirectly connected through a DHCP relay server to the DHCP server on the Brocade vRouter.

- LAB-NET2. This shared network serves the 2001:db8:100::/64 subnet, which is connected to a DHCP relay (R2), which is directly connected to the dp0p1p2 interface. The lease time remains at the default, 24 hours (86,400 seconds). The address pool uses the DNS name server at 2001:db8:111::111, which is on a separate subnet (not shown). The range of addresses is configured for .100 through .199.

The following figure shows the sample address pool configuration.

FIGURE 6 DHCPv6 address pool configuration using the listento option



NOTE
To configure the DHCPv6 address pool, perform the following steps in configuration mode.

TABLE 12 Configuring a DHCPv6 address pool

Step	Command
Configure the router interface to listen to DHCP messages.	`vyatta@R1# set service dhcpv6-server listento interface dp0p1p2`
Create the configuration node for LAB-NET on the 2001:db8:100::/64 subnet. Specify the start and stop IPv6 addresses for the pool.	`vyatta@R1# set service dhcpv6-server shared-network-name LAB-NET2 subnet 2001:db8:100::/64 address-range start 2001:db8:100::100 stop 2001:db8:100::199`
Specify a DNS server for LAB-NET.	`vyatta@R1# set service dhcp-server shared-network-name LAB-NET subnet 2001:db8:100::/64 name-server 2001:db8:111::111`
Commit the changes.	`vyatta@R1# commit`

TABLE 12 Configuring a DHCPv6 address pool (Continued)

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Show the configuration.</td><td>

```
vyatta@R1# show service dhcpv6-server
dhcpv6-server {
       listento {
               interface dp0p1p2
       }
} shared-network-name LAB-NET2 {
     subnet 2001:db8:100::/64 {
        address-range {
           start 2001:db8:100::100 {
              stop 2001:db8:100::199
           }
        }
        name-server 2001:db8:111::111
        lease 86400
     }
  }
```

</td></tr>
<tr><td>Show the interface configuration.</td><td>

```
vyatta@R1# show interfaces
   dataplane dp0p1p2 {
     address 2001:db9:101::0/64
     hw-id 00:0c:29:42:05:35
   }
```

</td></tr>
</table>

Setting up DHCPv6 relay

Configure DHCPv6 relay if you want the Brocade vRouter to forward DHCPv6 requests to another DHCPv6 server.

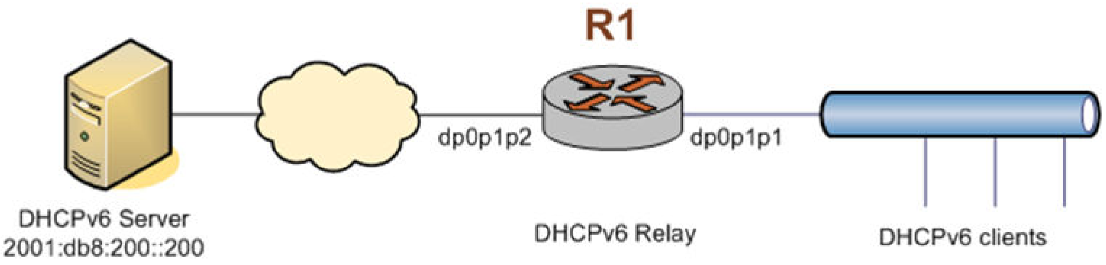
The DHCPv6 relay agent listens for requests sent by DHCPv6 clients and forwards them on to DHCPv6 servers. Because the client request packets and the relayed requests are often carried in IPv6 multicast packets, you must explicitly specify the interfaces on which the relay agent is to listen for requests and the interfaces on which it is to relay those requests.

Table 13 shows how to accomplish the following tasks:

- Configures both dp0p1p1 and dp0p1p2 for DHCPv6 relay. The system is expected to receive client requests for the DHCPv6 server through the dp0p1p1 interface. It forwards client-to-server DHCPv6 messages to the DHCPv6 server at 2001:db8:200::200 out through the dp0p1p2 interface. The DHCPv6 server refers to the interface on which client requests are received as the “listening interface,” and refers to the interface on which requests are relayed out as the “upstream interface.”
- Leaves other relay option parameters at default values. This means that R1 uses port 547 for DHCP messaging and has a maximum hop count of 10 hops.

Figure 7 shows the sample DHCPv6 relay configuration.

FIGURE 7 DHCPv6 relay configuration



To configure DHCPv6 relay, perform the following steps in configuration mode.

TABLE 13 Setting up DHCPv6 relay

Step	Command
Enable DHCPv6 relay to listen on the dp0p1p1 interface.	`vyatta@R1# set service dhcpv6-relay listen-interface dp0p1p1`
Enable DHCPv6 relay to forward requests on the dp0p1p2 interface specifying the DHCPv6 server address.	`vyatta@R1# set service dhcpv6-relay upstream-interface dp0p1p2 address 2001:db8:200::200`
Commit the changes.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show service dhcpv6-relay`
	`   listen-interface dp0p1p1 {`
	`   }`
	`   upstream-interface dp0p1p2 {`
	`           address 2001:db8:200::200`
	`   }`

Setting up DHCPv6 client

Configure DHCPv6 client if you want the Brocade vRouter to acquire an IPv6 address, parameters, or both from a DHCPv6 server. Refer to the “Ethernet Interfaces” chapter of the *Brocade 5600 vRouter LAN Interfaces Reference Guide* for more information on configuring a DHCPv6 client.

To configure a DHCPv6 client, perform the following steps in configuration mode.

TABLE 14 Setting up a DHCPv6 client on an Ethernet interface

Step	Command
Enable DHCPv6 client on the dp0p1p1 interface.	`vyatta@R1# set interface dataplane dp0p1p1 address dhcpv6`
Commit the change.	`vyatta@R1# commit`

Show the configuration.	`vyatta@R1# show interface dataplane dp0p1p1`
	`   address dhcpv6`
	`   hw-id b6:cc:6a:95:22:b2`

DHCPv6 Commands

release dhcpv6 interface <interface>

Releases the current DHCPv6 client lease on an interface.

Syntax **release dhcpv6 interface** *interface*

Parameters *interface*

An interface that uses DHCPv6 to obtain an IP address.

Modes Operational mode

Usage Guidelines Use this command to release the DHCPv6 client lease on an interface. The interface must be configured to obtain an address through DHCPv6. If the DHCPv6 client is in the process of acquiring an address, it stops that process. The client does not attempt to acquire a new address through DHCPv6.

renew dhcpv6 interface <interface>

Renews the current DHCPv6 client lease on an interface.

Syntax **renew dhcpv6 interface** *interface*

Parameters *interface*

An interface that uses DHCPv6 to obtain an IP address.

Modes Operational mode

Usage Guidelines Use this command to renew the DHCPv6 client lease on an interface. The interface must be configured to obtain an address through DHCPv6 server.

reset dhcpv6 server leases

Removes all DHCPv6 leases.

Syntax

reset dhcp server leases [**leases** | **lease ipv6** *ipv6-address*]

Parameters

ipv6-address

The IPv6 address with leases to be removed.

Modes

Operational mode.

Usage Guidelines

This command applies to leases provided by the DHCPv6 server. The server is configured by using service dhcpv6-server on page 114.

Use the **leases** command option to remove all DHCPv6 leases.

Use the **lease ipv6** command option to remove DHCPv6 leases from a particular IPv6 lease.

restart dhcpv6 relay-agent

Restarts the DHCPv6 relay agent.

Syntax **restart dhcpv6 relay-agent**

Modes Operational mode

Usage Guidelines Use this command to stop the DHCPv6 relay agent if it is running, then start it if it is configured. This command can be used if the DHCPv6 relay agent is not operating properly.

restart dhcpv6 server

Restarts the DHCPv6 server.

Syntax **restart dhcpv6 server**

Modes Operational mode

Usage Guidelines Use this command to stop and restart the DHCPv6 server. This command can be used if the DHCPv6 relay agent is not operating properly.

service dhcpv6-relay

Configures the system to relay DHCPv6 client messages to a DHCPv6 server.

Syntax

set service dhcpv6-relay

delete service dhcpv6-relay

show service dhcpv6-relay

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
    }
}
```

Usage Guidelines

Use this command to configure the system as a DHCPv6 relay agent.

You must configure the interfaces on which the system receives requests from DHCPv6 clients and the interfaces that send requests to DHCPv6 servers. The relay agent relays responses sent by the DHCPv6 servers back to the clients that sent the original request.

Use the **set** form of this command to define DHCPv6 relay configuration.

Use the **delete** form of this command to remove DHCPv6 relay configuration.

Use the **show** form of this command to view DHCPv6 relay configuration.

service dhcpv6-relay listen-interface <interface>

Specifies an interface for accepting DHCPv6 requests.

Syntax

set service dhcpv6-relay listen-interface *interface* [**address** *ipv6*]

delete service dhcpv6-relay listen-interface *interface* [**address**]

show service dhcpv6-relay listen-interface *interface* [**address**]

Parameters

interface

Mandatory. Multinode. An interface to accept DHCPv6 requests. At least one interface must be specified.

You can assign multiple interfaces to be used for DHCPv6 by creating multiple **listen-interface** configuration nodes.

ipv6

Optional. An IPv6 address on the specified interface on which to listen. If an address is not specified, one of the non-link-local addresses configured on the interface is used.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
        listen-interface interface {
            address ipv6
        }
    }
}
```

Usage Guidelines

Use this command to specify an interface for accepting DHCPv6 requests.

Use the **set** form of this command to specify an interface to accept DHCPv6 requests.

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

Use the **show** form of this command to view the specified value.

service dhcpv6-relay listen-port <port>

Specifies a port for accepting DHCPv6 requests.

Syntax

set service dhcpv6-relay listen-port *port*

delete service dhcpv6-relay listen-port *port*

show service dhcpv6-relay listen-port *port*

Command Default

The DHCPv6 Relay agent listens on port 547.

Parameters

port

Optional. The port on which to listen for DHCPv6 requests.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
        listen-port port
    }
}
```

Usage Guidelines

Use this command to specify a port for accepting DHCPv6 requests.

Use the **set** form of this command to specify a port to use to accept DHCPv6 requests.

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

Use the **show** form of this command to view the specified value.

service dhcpv6-relay max-hop-count <count>

Specifies the maximum number of hops before discarding DHCPv6 packets.

Syntax

set service dhcpv6-relay max-hop-count *count*

delete service dhcpv6-relay max-hop-count *count*

show service dhcpv6-relay max-hop-count *count*

Command Default

The maximum hop count is 10.

Parameters

count

Optional. The maximum hop count before discarding DHCPv6 packets. The default count is 10.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
        max-hop-count count
    }
}
```

Usage Guidelines

Use this command to specify the maximum number of hops before discarding DHCPv6 packets. This count is used to prevent loops.

Use the **set** form of this command to specify the maximum number of hops before discarding DHCPv6 packets.

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

Use the **show** form of this command to view the specified value.

service dhcpv6-relay upstream-interface <interface>

Specifies an interface for forwarding DHCPv6 requests.

Syntax

set service dhcpv6-relay upstream-interface *interface* [**address** *ipv6*]

delete service dhcpv6-relay upstream-interface *interface* [**address**]

show service dhcpv6-relay upstream-interface *interface* [**address**]

Parameters

interface

Mandatory. Multinode. An interface to forward DHCPv6 requests. At least one interface must be specified.

You can assign multiple interfaces to be used for DHCPv6 forwarding by creating multiple **upstream-interface** configuration nodes.

ipv6

Optional. An IPv6 address on the specified interface through which to forward queries. If an address is not specified, the queries are sent to the all DHCP relay agents and servers multicast group.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
        upstream-interface interface {
            address ipv6
        }
    }
}
```

Usage Guidelines

Use this command to specify an interface for forwarding DHCPv6 requests.

Use the **set** form of this command to specify an interface to use to forward DHCPv6 requests.

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

Use the **show** form of this command to view the specified value.

service dhcpv6-relay use-interface-id-option

Specifies that the relay agent is to insert the DHCPv6 interface ID option.

Syntax

set service dhcpv6-relay use-interface-id-option

delete service dhcpv6-relay use-interface-id-option

show service dhcpv6-relay use-interface-id-option

Command Default

The DHCPv6 interface ID option is not inserted if a single listening interface is defined, but is inserted automatically if more than one listening interface is defined.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-relay {
        use-interface-id-option
    }
}
```

Usage Guidelines

Use this command to specify that DHCPv6 is to insert the interface ID option. Note that this option is automatically inserted when two or more listening interfaces are configured, so this parameter affects just system behavior when only one listening interface is configured.

Use the **set** form of this command to specify that DHCPv6 is to insert the interface ID option.

Use the **delete** form of this command to return the system to its default behavior.

Use the **show** form of this command to view the specified value.

service dhcpv6-server

Enables DHCPv6 server functionality.

Syntax

set service dhcpv6-server

delete service dhcpv6-server

show service dhcpv6-server

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
    }
}
```

Usage Guidelines

Use the **set** form of this command to enable DHCPv6 server functionality.

Use the **delete** form of this command to remove DHCPv6 server functionality.

Use the **show** form of this command to view DHCPv6 server configuration.

service dhcpv6-server listento interface <dp-interface>

Allows the DHCP server to create address pools for clients that are indirectly connected to a dataplane network interface through a DHCP relay server.

Syntax

set service dhcpv6-server listento interface *dp-interface*

delete service dhcpv6-server listento interface

show service dhcpv6-server

Parameters

dp-interface

A dataplane interface on the router. It must have a valid IP address.

Modes

Configuration mode.

Configuration Statement

```
service {
    dhcpv6-server {
        listento {
        interface dp-interface
        }
    }
}
```

Usage Guidelines

Use this command to enable the DHCP server to create IP address pools for clients that are not directly connected to the router. For example, if clients on the B subnet connect to the router through a DHCP relay server, the DHCP relay server connects to the router through a dataplane interface on the A subnet, and the dataplane interface has a valid IP address, using this command allows the DHCP server to create IP address pools for clients that are on the B subnet.

Use the **set** form of this command to create an IP address pool for clients that are indirectly connected to the router through a dataplane network interface.

Use the **delete** form of this command to remove a data-plane interface from the DHCP server configuration. If no dataplane interfaces are configured, the DHCP server cannot create address pools.

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

service dhcpv6-server preference <preference>

Specifies the DHCPv6 server preference.

Syntax

set service dhcpv6-server preference *preference*

delete service dhcpv6-server preference

show service dhcpv6-server preference

Command Default

The DHCPv6 server preference is not set.

Parameters

preference

Optional. The preference for the DHCPv6 server. The preference ranges from 0 through 255.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        preference preference
    }
}
```

Usage Guidelines

Use this command to specify the DHCPv6 server preference to DHCPv6 clients. When clients receive advertise messages from multiple servers that include preferences, they choose the server with the highest preference.

Use the **set** form of this command to specify the DHCPv6 server preference.

Use the **delete** form of this command to restore the default state, that is, the DHCPv6 server preference is not set.

Use the **show** form of this command to display the DHCPv6 server preference.

service dhcpv6-server shared-network-name <name>

Assigns a name to a physical subnet.

Syntax

set service dhcpv6-server shared-network-name *name*

delete service dhcpv6-server shared-network-name *name*

show service dhcpv6-server shared-network-name *name*

Parameters

name

Multinode. The name for a physical subnet.

You can define multiple subnets by creating multiple **shared-network-name** configuration nodes, each with a different name.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
        }
    }
}
```

Usage Guidelines

Use this command to assign a name to a physical subnet. The subnet created may be directly connected to the system. The name is arbitrary and need not match any name used for this subnet elsewhere within the system.

Use the **set** form of this command to assign a name to a physical subnet.

Use the **delete** form of this command to delete the name of a physical subnet.

Use the **show** form of this command to display the name of a physical subnet.

service dhcpv6-server shared-network-name <name> subnet <ipv6net>

Specifies an IPv6 subnet to which the DHCPv6 server provides access.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net*

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet to which the DHCPv6 server provides access. The format is *ipv6-addr/prefix*.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify an IPv6 subnet to which the DHCPv6 server provides access. The DHCPv6 server responds to clients on this subnet by using the parameters and addresses defined in this subtree.

Use the **set** form of this command to specify the DHCPv6 subnet.

Use the **delete** form of this command to remove DHCPv6 subnet configuration.

Use the **show** form of this command to view DHCPv6 subnet configuration.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> address-range

Specifies a range of IPv6 addresses that can be assigned to clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range**

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range**

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                address-range {
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a range of IPv6 addresses that can be assigned to clients. If no address range is provided, the DHCPv6 server operates in a stateless mode on this subnet, which means that it does not assign dynamic IPv6 addresses and thus does not maintain state information about those assignments.

Use the **set** form of this command to create the address-range configuration node.

Use the **delete** form of this command to remove the address-range configuration.

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

service dhcpv6-server shared-network-name <name> subnet <ipv6net> address-range prefix <pool-ipv6net>

Specifies a pool of IPv6 addresses that can be assigned to clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range prefix** *ipv6net* [**temporary**]

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range prefix** *ipv6net* [**temporary**]

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range prefix** *ipv6net*

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

pool-ipv6net

Optional. An IPv6 address prefix that defines a pool of consecutive addresses available for assignment to clients. The specified prefix must be a subset of the subnet prefix.

temporary

Optional. If specified, indicates that the range can be used for assigning privacy addresses (RFC 4941).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                address-range {
                    prefix pool-ipv6net {
                        temporary
                    }
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a pool of IPv6 addresses that can be assigned to clients.

Use the **set** form of this command to create the address-range prefix configuration.

Use the **delete** form of this command to remove the address-range prefix configuration.

Use the **show** form of this command to view the address-range prefix configuration.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> address-range start <start-ipv6>

Specifies the beginning and ending addresses in a range of IPv6 addresses that can be assigned to clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range start** *start-ipv6* [**stop** *stop-ipv6* | **temporary**]

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range start** *start-ipv6* [**stop** | **temporary**]

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **address-range start** *start-ipv6* [**stop** | **temporary**]

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

start-ipv6

Optional. Multinode. The beginning address in a range of consecutive IPv6 addresses that are available for assignment to clients.

stop-ipv6

Optional. The ending address in a range of consecutive IPv6 addresses that are available for assignment to clients. If not specified, only the beginning address is available for assignment.

temporary

Optional. If specified, indicates that the range can be used for assigning privacy addresses (RFC 4941).

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                address-range {
                    start start ipv6 {
                        stop stop ipv6
                            temporary
                    }
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the beginning and ending addresses in a range of IPv6 addresses that can be assigned to clients.

Use the **set** form of this command to create the address-range configuration.

Use the **delete** form of this command to remove the address-range configuration.

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

service dhcpv6-server shared-network-name <name> subnet <ipv6net> description <desc>

Provides a description of a subnet.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **description** *desc*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **description**

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **description**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

desc

A description of the specified subnet.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcp-server {
        shared-network-name name {
            subnet ipv6net {
                description desc
            }
        }
    }
}
```

Usage Guidelines

Use this command to provide a description of a subnet.

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

Use the **delete** form of this command to delete the description of a subnet.

Use the **show** form of this command to display the description of a subnet.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> domain-search <domain>

Specifies a domain name to include in the domain search list.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **domain-search** *domain*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **domain-search** *domain*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **domain-search**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

domain

Multinode. A domain name to include in the domain search list.

You can specify more than one domain name by including this parameter multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                domain-search domain
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a domain name to include in the domain search list. Hosts use the domain search list when resolving host names in DNS. Values are listed in the option, and communicated to the client, in the order entered.

Use the **set** form of this command to specify a domain name.

Use the **delete** form of this command to delete a domain name.

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

service dhcpv6-server shared-network-name <name> subnet <ipv6net> lease-time

Sets the client lease time.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **lease-time** {**default** *default-time* | **maximum** *max-time* | **minimum** *min-time*}

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **lease-time** {**default** | **maximum** | **minimum**}

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **lease-time** {**default** | **maximum** | **minimum**}

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

default-time

The default lease time in seconds. The time is assigned to a client if the client does not request a specific lease time.

max-time

The maximum lease time in seconds that is assigned to a lease. If the client requests a time larger than the maximum lease time, the maximum time is used.

min-time

The minimum lease time in seconds that is assigned to a lease. If the client requests a time smaller than the minimum lease time, the minimum time is used.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                lease-time {
                    default default-time
                    maximum max-time
                    minimum min-time
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the client lease time.

Use the **set** form of this command to specify the lease time.

Use the **delete** form of this command to delete the lease time.

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

service dhcpv6-server shared-network-name <name> subnet <ipv6net> name-server <ipv6>

Specifies the address of a recursive DNS server (RDNSS) for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **name-server** *ipv6*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **name-server** *ipv6*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **name-server**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

ipv6

Multinode. The IPv6 address of an RDNSS.

You can specify more than one server by specifying this parameter multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                name-server ipv6
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of an RDNSS that is available to DHCPv6 clients. Addresses are listed in the order they are specified.

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

Use the **delete** form of this command to delete the address of an RDNSS.

Use the **show** form of this command to display the address of an RDNSS.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> nis-domain <nis-domain-name>

Specifies the Network Information Service (NIS) domain for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-domain** *nis-domain-name*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-domain**

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-domain**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

nis-domain-name

The name of the NIS domain for DHCPv6 clients.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                nis-domain nis-domain-name
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify an NIS domain for DHCPv6 clients.

Use the **set** form of this command to specify the NIS domain.

Use the **delete** form of this command to delete the NIS domain.

Use the **show** form of this command to display the NIS domain.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> nisplus-domain <nisplus-domain-name>

Specifies the Network Information Service Plus (NIS+) domain for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-domain** *nisplus-domain-name*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-domain**

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-domain**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

nisplus-domain-name

The name of an NIS+ domain for DHCPv6 clients.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                nisplus-domain nisplus-domain-name
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify an NIS+ domain for DHCPv6 clients.

Use the **set** form of this command to specify the NIS+ domain.

Use the **delete** form of this command to delete the NIS+ domain.

Use the **show** form of this command to display the NIS+ domain.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> nisplus-server <ipv6>

Specifies the address of the Network Information Service Plus (NIS+) server for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-server** *ipv6*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-server** *ipv6*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nisplus-server**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

ipv6

Multinode. The address of the NIS+ server for DHCPv6 clients.

You can specify more than one address by issuing this statement multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                nisplus-server ipv6
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of the NIS+ server for DHCPv6 clients. Addresses are listed in the order they are specified.

Use the **set** form of this command to specify the NIS+ server address.

Use the **delete** form of this command to delete the address of the NIS+ server.

Use the **show** form of this command to display the address of the NIS+ server.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> nis-server <ipv6>

Specifies the address of the NIS server for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-server** *ipv6*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-server** *ipv6*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **nis-server**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr/prefix*.

ipv6

Multinode. The address of the NIS server for DHCPv6 clients.

You can specify more than one address by entering this command multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                nis-server ipv6
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of the NIS server for DHCPv6 clients. Addresses are listed in the order they are specified.

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

Use the **delete** form of this command to delete the address of the NIS server.

Use the **show** form of this command to display the address of the NIS server.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> sip-server-address <ipv6>

Specifies the address of the Session Initiation Protocol (SIP) server for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-address** *ipv6*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-address** *ipv6*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-address**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

ipv6

Multinode. The address of the SIP server for DHCPv6 clients.

You can specify more than one address by entering this command multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                sip-server-address ipv6
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of the SIP server for DHCPv6 clients. Addresses are listed in the order they are specified.

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

Use the **delete** form of this command to delete the address of the SIP server.

Use the **show** form of this command to display the address of the SIP server.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> sip-server-name <sip-server-name>

Specifies the name of the Session Initiation Protocol (SIP) server for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-name** *sip-server-name*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-name** *sip-server-name*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sip-server-name**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

sip-server-name

Multinode. The name of the SIP server for DHCPv6 clients.

You can specify more than one name by entering this command multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                sip-server-name sip-server-name
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the name of the SIP server for DHCPv6 clients. Addresses are listed in the order they are specified.

Use the **set** form of this command to specify the SIP server name.

Use the **delete** form of this command to delete the name of the SIP server.

Use the **show** form of this command to display the name of the SIP server.

service dhcpv6-server shared-network-name <name> subnet <ipv6net> sntp-server-address <ipv6>

Specifies the address of the Simple Network Time Protocol (SNTP) for DHCPv6 clients.

Syntax

set service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sntp-server-address** *ipv6*

delete service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sntp-server-address** *ipv6*

show service dhcpv6-server shared-network-name *name* **subnet** *ipv6net* **sntp-server-address**

Parameters

name

The name of a physical subnet.

ipv6net

Optional. Multinode. An IPv6 subnet served by the DHCPv6 server. The format is *ipv6-addr*/*prefix*.

ipv6

Multinode. The address of the SNTP server for DHCPv6 clients.

You can specify more than one address by entering this command multiple times.

Modes

Configuration mode

Configuration Statement

```
service {
    dhcpv6-server {
        shared-network-name name {
            subnet ipv6net {
                sntp-server-address ipv6
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the address of the SNTP server for DHCPv6 clients. Addresses are listed in the order they are specified. SNTP is a subset of NTP and includes extensions to operate over IPv6. It is specified in: http://tools.ietf.org/html/rfc4330.

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

Use the **delete** form of this command to delete the address of the SNTP server.

Use the **show** form of this command to display the address of the SNTP server.

service dhcpv6-server static-mapping <mapping-name>

Specifies the IPv6 address for a client.

Syntax

set service dhcpv6-server static-**mapping** *mapping*-*name* [**ipv6-address** *ipv6* | **identifier** *identifier*]

delete service dhcpv6-server static-**mapping** *mapping*-*name* [**ipv6-address** | **identifier**]

show service dhcpv6-server static-**mapping** *mapping*-*name* [**ipv6-address** | **identifier**]

Parameters

mapping-name

A name to identify the static mapping.

ipv6

The IPv6 address that is assigned to a client.

identifier

The character string that identifies a client. It is compared against the ia-na option sent by the client. This value is typically the low-order 4 bytes of the MAC address of the client.

Modes

Configuration mode.

Configuration Statement

```
service {
    dhcpv6-server {
        static-mapping mapping-name {
        ipv6-address ipv6
        identifier identifier
        }
    }
}
```

Usage Guidelines

Use this command to specify the IPv6 address for a client.

Use the **set** form of this command to create the prefix-delegation configuration.

Use the **delete** form of this command to remove the prefix-delegation configuration.

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

show dhcpv6 client leases

Displays DHCPv6 information for interfaces that are configured as DHCPv6 clients.

Syntax **show dhcpv6 client leases**

Modes Operational mode

Usage Guidelines Use this command to display current DHCPv6 client information for interfaces that are configured as DHCPv6 clients. If an address has been acquired, the command shows the lease parameters associated with that address, including the unique ID, assigned IPv6 address, and time remaining on the lease.

To configure an interface as a DHCPv6 client, refer to the documentation for that interface.

show dhcpv6 relay-agent status

Displays the status of the DHCPv6 relay agent.

Syntax **show dhcpv6 relay-agent status**

Modes Operational mode

Usage Guidelines Use this command to display the status of the DHCPv6 relay agent. This status includes an indication of whether the DHCPv6 relay agent is configured. If it is configured, the command indicates whether the DHCPv6 relay agent is running.

show dhcpv6 server leases

Displays the status of all leases assigned by the DHCPv6 server.

Syntax **show dhcpv6 server leases**

Modes Operational mode

Usage Guidelines Use this command to display the status of all leases assigned by the DHCPv6 server. For each lease, it shows the unique ID of the client, assigned IPv6 address, and time remaining on the lease.

show dhcpv6 server status

Displays the status of the DHCPv6 server.

Syntax **show dhcpv6 server status**

Modes Operational mode

Usage Guidelines Use this command to display the status of the DHCPv6 server. This status includes an indication of whether the DHCPv6 server is configured. If it is configured, the command indicates whether the DHCPv6 server is running. The command notes whether any address ranges are configured. If none are configured, the server can assign only parameters; it cannot assign addresses.

DNS

DNS overview

DNS is an Internet directory service that provides mappings between human-readable domain names and numeric IP addresses. DNS mappings are recorded in resource records that are stored on name servers distributed throughout the Internet. A device needing to access a host across the Internet sends a DNS query to a name server. The name server consults its resource records and returns an answer with the IP address of the specified name.

The DNS system has billions of resource records. If the requested record is not local to the consulted name server, the name server consults another name server, and so on, until the requested information is located and returned.

There are billions of resource records in the DNS system. To keep the data manageable, the records are divided into zones, which contain resource records for a DNS domain or subdomain.

The Brocade vRouter supports three main DNS-related features:

- System DNS on page 139
- Dynamic DNS on page 139
- DNS forwarding on page 140

System DNS

In system DNS, you define the list of name servers that the Brocade vRouter can use to resolve host names to IP addresses. This list is created by using the **system name-server** command. (The **system name-server** command is described in *Brocade 5600 vRouter Basic System Reference Guide*; for your convenience, an example of system DNS is provided in Configuring access to a name server on page 140.)

Dynamic DNS

Originally, DNS mappings were statically specified in “zone files,” which were periodically loaded onto DNS servers. These zone files worked reasonably well at a time when most hosts were configured with static IP addresses. However, since the 1990s, many network endpoints have been assigned IP addresses using dynamic protocols such as DHCP. Until 1997, devices with DHCP-assigned IP addresses essentially could not participate in the DNS system.

In 1997, the Internet Engineering Task Force (IETF) published RFC 2136, *Dynamic Updates in the Domain Name System* , describing the dynamic DNS update protocol. Dynamic DNS (DDNS) provides a mechanism for DNS entries to be established and removed dynamically. Devices using dynamic DNS can notify a domain name server in real time of changes to host name, IP address, or other DNS-related information.

This feature is particularly useful for systems in which a dynamic IP address is provided by the ISP. Whenever the IP address changes, the Brocade vRouter updates a DDNS service provider with the change. The DDNS provider is responsible for propagating this change to other DNS servers. The Brocade vRouter supports a number of DDNS providers.

DNS forwarding

In many environments that use consumer-level ISP connections, the ISP both assigns the client router with its IP address and notifies the client router of the DNS server to use. In many cases, the IP address of the DNS server itself is assigned through DHCP and changes periodically; the ISP notifies the client router of the change in DNS server IP address through periodic updates. This makes it problematic to statically configure a DNS server IP address on the DHCP server of the client router for its LAN clients.

In cases like these, the Brocade vRouter can use DNS forwarding (also called DNS relay) to maintain connectivity between hosts on its network and the DNS server of the ISP.

When DNS forwarding is used, the client router offers its own client-side IP address (which is static) as the DNS server address to the hosts on its network, so that all client DNS requests are made to the client-side address of the client router. When DNS requests are made, the client router forwards them to the ISP DNS server; answers are directed back to the client router and forwarded through to the client hosts. If the ISP changes the address of its DNS server, the client router simply records the new address of the server. The server address remains unchanged from the perspective of the LAN clients.

Another advantage to DNS forwarding is that DNS requests are cached in the Brocade vRouter (until either the time-to-live value in the DNS record expires or the cache fills). Subsequent requests for a cached entry are responded to locally, with a corresponding reduction in WAN traffic.

DNS configuration examples

This section presents the following topics:

- Configuring access to a name server on page 140
- Configuring dynamic DNS on page 141
- Configuring DNS forwarding on page 143
- Statically configured entries and DNS forwarding on page 145

Configuring access to a name server

To be able to translate host names (such as www.brocade.com) to IP addresses (such as 69.59.150.141), the system must be able to access a DNS server.

Configuring access to a DNS server is a function of basic system management, and is described in *Brocade 5600 vRouter Basic System Reference Guide*. For your convenience, the configuration example is repeated here.

Table 15 configures a static IP address for the DNS server at address 12.34.56.100. To configure the Brocade vRouter in this way, perform the following steps.

TABLE 15 Configuring static access to a DNS name server

Step	Command
Specify the IP address of the DNS server.	`vyatta@R1# set system name-server 12.34.56.100`

Configuring dynamic DNS

Figure 8 on page 142 shows a typical DDNS scenario. In this scenario:

- The Brocade vRouter (R1) is connected to an ISP via dp0p1p1.
- The network domain is company.com.
- The Brocade vRouter host name is r1.company.com.
- The web server of the company is located behind the Brocade vRouter. Its host name is www.company.com.
- The ISP is providing dynamic IP addresses to its clients through DHCP.
- The IP address of the dp0p1p1 interface in the Brocade vRouter changes over time because of the dynamic assignment by the ISP.
- The web server of the company is behind a Network Address Translation (NAT) device on the Brocade vRouter, so its IP address (as viewed from the Internet) changes when the ISP assigns a new address to the dp0p1p1 interface.
- Because the web address of the server changes, responses to DNS queries for www.company.com must also change to the new IP address. DDNS resolves this problem.

DDNS allows the Brocade vRouter (R1) to update the DNS system with the new IP address information for any local host names (for example, r1.company.com, and www.company.com) whenever the IP address on dp0p1p1 changes. The setup process is as follows:

1. Sign up for DDNS service from one of the supported service providers:

   - DNS Park: www.dnspark.com
   - DSL Reports: www.dslreports.com
   - DynDNS: www.dyndns.com
   - easyDNS: www.easydns.com
   - namecheap: www.namecheap.com
   - Sitelutions: www.sitelutions.com
   - zoneedit: www.zoneedit.com

   The individual providers offer instructions for sign-up.

   NOTE
   Depending on the service provider, host names may need to include the domain name (for example, www instead of www.company.com).

2. Configure the Brocade vRouter (R1 in the example) with service provider information such as the service name, a login ID, and a password so that the system can determine how to log on and send updates to the DDNS service provider.
3. Configure the Brocade vRouter with the host names that must be updated in the DNS system when the IP address on dp0p1p1 changes.

Setting up Dynamic DNS

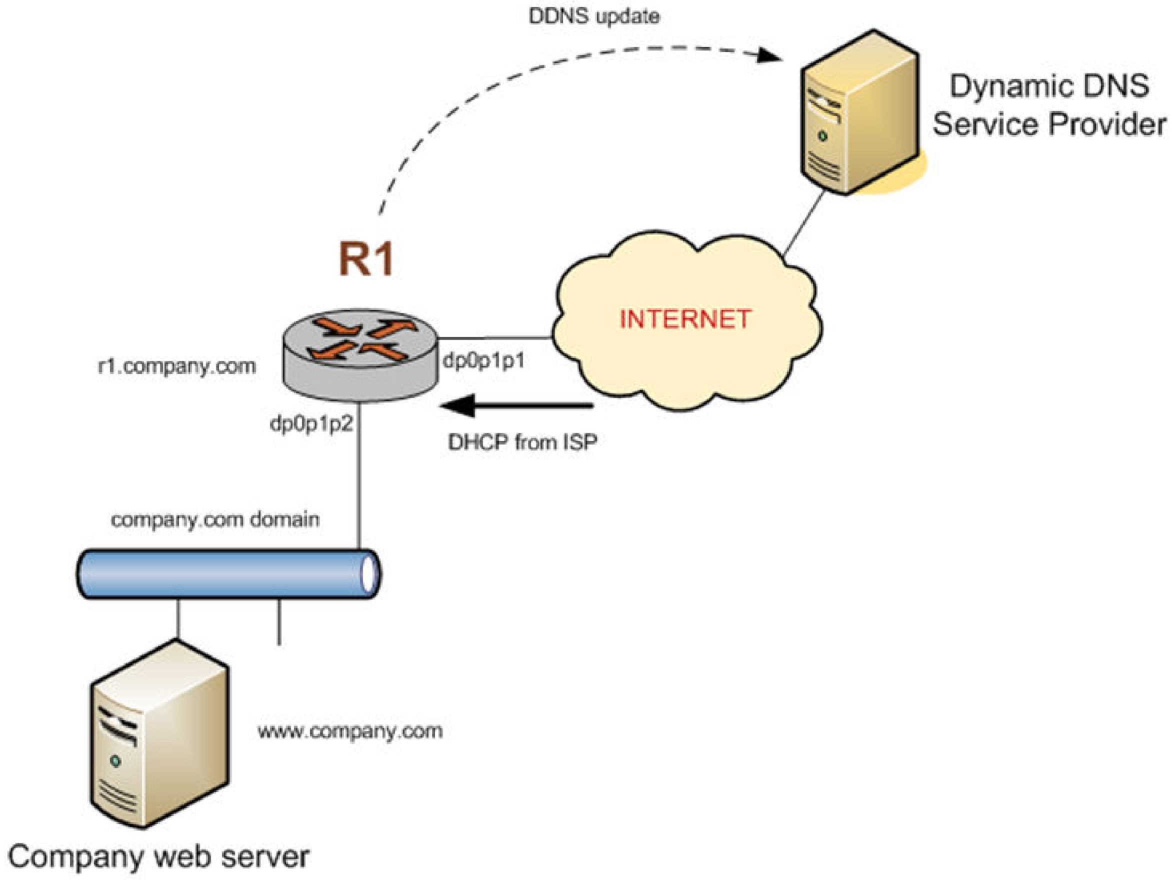


FIGURE 8 Dynamic DNS

The following example shows how to set up DDNS for DDNS service provider DynDNS. It is assumed for this example that you have already signed up with DynDNS). To configure the Brocade vRouter in this way, perform the following steps in configuration mode.

TABLE 16 Setting up dynamic DNS

Step	Command
Set the service provider.	`vyatta@R1# set service dns dynamic interface dp0p1p1 service dyndns`
Set the DDNS service provider login ID (for example, vtest).	`vyatta@R1# set service dns dynamic interface dp0p1p1 service dyndns login vtest`
Set the DDNS service provider password (for example, testpwd).	`vyatta@R1# set service dns dynamic interface dp0p1p1 service dyndns password testpwd`
Specify R1 as a host name whose DNS entry needs to be updated when the IP address on dp0p1p1 changes.	`vyatta@R1# set service dns dynamic interface dp0p1p1 service dyndns host-name r1.company.com`

TABLE 16 Setting up dynamic DNS (Continued)

Step	Command
Specify www as a host name whose DNS entry needs to be updated when the IP address on dp0p1p1 changes.	`vyatta@R1# set service dns dynamic interface dp0p1p1 service dyndns host-name www.company.com`
Commit the changes.	`vyatta@R1# commit OK`

Show the dynamic DNS configuration.	`vyatta@R1# show service dns dynamic`
	`interface dp0p1p1 {`
	`    service dyndns {`
	`         host-name r1.company.com`
	`         host-name www.company.com`
	`         login vtest`
	`         password testpwd`
	`    }`
	`}`

At this point, whenever the IP address on dp0p1p1 changes, the Brocade vRouter automatically logs onto the DynDNS service by using the vtest login ID and the testpwd password. It sends an update for the r1.company.com and www.company.com host names specifying the new IP address required to reach those hosts on the company.com domain. External users that query DNS for r1.company.com or www.company.com are subsequently answered with the new address from the DNS system.

NOTE
Dynamic DNS updates are logged. To see the updates, set up logging by using the system **syslog global facility** *facility* **level** *level* command, where *facility* is daemon and *level* is notice. See *Brocade 5600 vRouter Basic System Reference Guide* for details.

Configuring DNS forwarding

Configuring the Brocade vRouter for DNS forwarding has two main steps:

1. Specifying the DNS name servers to which to forward
2. Specifying the interfaces on which to listen for DNS requests

Specifying DNS Name Servers

Name server locations can be obtained in three ways:

- From the system name server list, defined by using the **set system name-server** command
- By DHCP
- By listing additional name servers by using service dns forwarding dhcp <interface> on page 157

By default, the Brocade vRouter forwards DNS requests to name servers on the system name server list plus name servers obtained through DHCP. You can override the default behavior by specifying any or all of the following:

- Specifically use system-defined name servers. To do this, use service dns forwarding system on page 160.
- Specifically use name servers received for the interface that is using DHCP client to get an IP. To do this, use service dns forwarding dhcp <interface> on page 157.
- List additional name servers by using service dns forwarding name-server <ipv4> on page 159.

These three options can be used in any combination; however, using any of them eliminates the default DNS forwarding behavior.

When DNS forwarding starts or restarts, it broadcasts a message to all the name servers in the pool and selects the first name server to answer. This name server is used unless it becomes unreachable, in which case the system sends another broadcast message to the remaining name servers in the pool.

Specifying the Listening Interfaces

The listening interfaces are the interfaces to which internal clients forward DNS requests. The DNS forwarding service listens for these requests and forwards them to the name server.

To set the listening interface, use service dns forwarding listen-on <interface> on page 158. You can specify more than one interface by issuing this command multiple times.

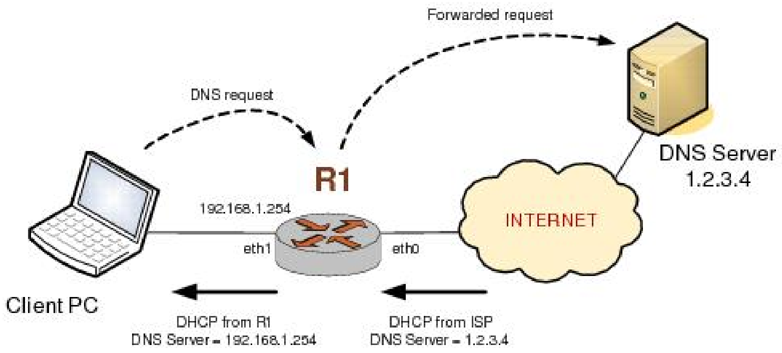
DNS Forwarding Scenario

After these steps are completed, DNS forwarding is set up. At this point, the Brocade vRouter DHCP server can be used to distribute the DNS forwarding interface address to DHCP clients. (For information about setting up a DHCP server on the Brocade vRouter, see DHCP on page 17.

Figure 9 shows a typical scenario in which DNS forwarding is deployed. In this scenario:

- The ISP is providing dynamic IP addresses to its customers, including a Brocade vRouter (R1) through DHCP.
- The Brocade vRouter (R1) is providing DHCP service to clients on its local network.
- Local clients send DNS requests to the Brocade vRouter.
- The DNS forwarding service on the Brocade vRouter forwards the requests to the the DNS server of the ISP.

FIGURE 9 Scenario using DNS forwarding



The following example shows how to set up the key parts of the Brocade vRouter for the preceding scenario. To configure the Brocade vRouter in this way, perform the following steps in configuration mode.

TABLE 17 Setting up DNS forwarding

Step	Command
Set IP address and prefix on dp0p1p2.	`vyatta@R1# set interfaces dataplane dp0p1p2 address 192.168.1.254/24`
Set dp0p1p1 as a DHCP client.	`vyatta@R1# set interfaces dataplane dp0p1p1 address dhcp`

TABLE 17 Setting up DNS forwarding (Continued)

Step	Command
Set up the DHCP server on R1 by creating the configuration node for dp0p1p2_POOL on subnet 192.168.1.0/24. Specify the start and stop IP addresses for the pool.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_POOL subnet 192.168.1.0/24 start 192.168.1.100 stop 192.168.1.199`
Specify the default router for dp0p1p2_POOL.	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_POOL subnet 192.168.1.0/24 default-router 192.168.1.254`
Create a DNS server list using DNS server information provided by the DHCP server of the ISP (on dp0p1p1).	`vyatta@R1# set service dns forwarding dhcp dp0p1p1`
Listen for DNS requests on dp0p1p2.	`vyatta@R1# set service dns forwarding listen-on dp0p1p2`
Specify a DNS server for dp0p1p2_POOL (in this case, it acts as a DNS forwarder).	`vyatta@R1# set service dhcp-server shared-network-name dp0p1p2_POOL subnet 192.168.1.0/24 dns-server 192.168.1.254`
Commit the changes.	`vyatta@R1# commit`

Show the DNS-related configuration.	`vyatta@R1# show service dns`
	`forwarding {`
	`        dhcp dp0p1p1`
	`        listen-on dp0p1p2`
	`}`

Statically configured entries and DNS forwarding

Because of difficulties interworking with network address translation (NAT) on the corporate gateway, it is sometimes difficult to obtain correct IP addresses for hosts on the corporate network. To work around this problem, you can create static entries on a local Brocade vRouter by using the **system static-host-mapping** command. Any entries configured in this way are compared with incoming DNS queries before the query is passed to DNS forwarding. If a match is found, the corresponding IP address is returned.

The following table shows how to set up the system to return an IP address of 12.34.56.78 if it receives a DNS query for either vyatta.com or vdut1.

TABLE 18 Setting up static entries

Step	Command
Create the static host-mapping configuration node.	`vyatta@R1# set system static-host-mapping host-name vyatta.com`
Provide an alias host name (this step is optional).	`vyatta@R1# set system static-host-mapping host-name vyatta.com alias vdut1`
Specify the IP address to be returned in response to the DNS query.	`vyatta@R1# set system static-host-mapping host-name vyatta.com inet 12.34.56.78`
Commit the changes.	`vyatta@R1# commit`

TABLE 18 Setting up static entries (Continued)

Step	Command

Show the static host-mapping configuration.	`vyatta@R1# show system static-host-mapping`
	` host-name vyatta.com{`
	`       alias vdut1`
	`       inet 12.34.56.78`
	` }`

DNS Commands

reset dns forwarding all

Resets all counters related to DNS forwarding and resets the DNS forwarding cache.

Syntax **reset dns forwarding all**

Modes Operational mode

Usage Guidelines Use this command to reset all counters related to DNS forwarding and remove all entries from the DNS forwarding cache.

reset dns forwarding cache

Removes all entries from the DNS forwarding cache.

Syntax **reset dns forwarding cache**

Modes Operational mode

Usage Guidelines Use this command to remove all entries from the DNS forwarding cache.

service dns dynamic interface <interface>

Enables support of dynamic DNS (DDNS) on an interface.

Syntax

set service dns dynamic interface *interface*

delete service dns dynamic interface *interface*

show service dns dynamic interface *interface*

Parameters

interface

Multinode. An interface that is to support DDNS.

You can have more than one interface that supports DDNS by creating multiple **interface** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
            }
        }
    }
}
```

Usage Guidelines

Use this command to enable support of DDNS on an interface.

Use the **set** form of this command to enable support of DDNS on an interface.

Use the **delete** form of this command to disable DDNS on an interface and remove all its dynamic DNS configuration.

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

service dns dynamic interface <interface> service <service>

Specifies a dynamic DNS (DDNS) service provider.

Syntax

set service dns dynamic interface *interface* **service** *service*

delete service dns dynamic interface *interface* **service** *service*

show service dns dynamic interface *interface* **service** *service*

Parameters

interface

Multinode. An interface that supports DDNS.

service

Multinode. The name of a DDNS service provider. The name is one of the following: **dnspark**, **dslreports**, **dyndns**, **easydns**, **namecheap**, **sitelutions**, or **zoneedit**.

You can specify more than one DDNS provider for each interface by creating multiple **service** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
                service service
            }
         }
    }
}
```

Usage Guidelines

Use this command to specify the organizations that provides DDNS service to the Brocade vRouter.

Use the **set** form of this command to specify a DDNS service provider.

Use the **delete** form of this command to remove a DDNS service provider.

Use the **show** form of this command to display information for a DDNS service provider.

service dns dynamic interface <interface> service <service> host-name <hostname>

Specifies the name of a host for which to update the DNS record of the dynamic DNS (DDNS) service provider.

Syntax

set service dns dynamic interface *interface* **service** *service* **host-name** *hostname*

delete service dns dynamic interface *interface* **service** *service* **host-name** *hostname*

show service dns dynamic interface *interface* **service** *service* **host-name**

Parameters

interface

Multinode. An interface that supports DDNS.

service

Multinode. The name of a DDNS service provider. The name is one of the following: **dnspark**, **dslreports**, **dyndns**, **easydns**, **namecheap**, **sitelutions**, or **zoneedit**.

hostname

The name of a host.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
                service service {
                    host-name hostname
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the name of a host for which to update the DNS record of the DDNS service provider.

Use the **set** form of this command to specify a host name.

Use the **delete** form of this command to remove the host name from the configuration.

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

service dns dynamic interface <interface> service <service> login <service-login>

Specifies a login ID to use to log on to a dynamic DNS (DDNS) service provider.

Syntax

set service dns dynamic interface *interface* **service** *service* **login** *service-login*

delete service dns dynamic interface *interface* **service** *service* **login**

show service dns dynamic interface *interface* **service** *service* **login**

Parameters

interface

Multinode. An interface that supports DDNS.

service

Multinode. The name of a DDNS service provider. The name is one of the following: **dnspark**, **dslreports**, **dyndns**, **easydns**, **namecheap**, **sitelutions**, or **zoneedit**.

login

A login ID.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
                service service {
                    login service-login
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the login ID to use to log on to a DDNS service provider.

Use the **set** form of this command to specify the login ID for a DDNS service provider.

Use the **delete** form of this command to remove the login ID for a DDNS service provider.

Use the **show** form of this command to display the login ID for a DDNS service provider.

service dns dynamic interface <interface> service <service> password <service-password>

Specifies the password to use to log on to a dynamic DNS (DDNS) service provider.

Syntax

set service dns dynamic interface *interface* **service** *service* **password** *service-password*

delete service dns dynamic interface *interface* **service** *service* **password**

show service dns dynamic interface *interface* **service** *service* **password**

Parameters

interface

Multinode. An interface that supports DDNS.

service

Multinode. The name of a DDNS service provider. The name is one of the following: **dnspark**, **dslreports**, **dyndns**, **easydns**, **namecheap**, **sitelutions**, or **zoneedit**.

password

A password.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
                service service {
                    password service-password
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the password to use to log on to a DDNS service provider.

Use the **set** form of this command to specify the password for a DDNS service provider.

Use the **delete** form of this command to remove the password for a DDNS service provider.

Use the **show** form of this command to display the password for a DDNS service provider.

service dns dynamic interface <interface> service <service> server <addr>

Specifies a server to which to send dynamic DNS (DDNS) updates.

Syntax

set service dns dynamic interface *interface* **service** *service* **server** *addr*

delete service dns dynamic interface *interface* **service** *service* **server**

show service dns dynamic interface *interface* **service** *service* **server**

Command Default

The default server of the DDNS service provider is used.

Parameters

interface

Multinode. An interface that supports DDNS.

service

Multinode. The name of a DDNS service provider. The name is one of the following: **dnspark**, **dslreports**, **dyndns**, **easydns**, **namecheap**, **sitelutions**, or **zoneedit**.

addr

An IP address or a host name. Only some DDNS service providers require this address or host name.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        dynamic {
            interface interface {
                service service {
                    server addr
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify the IP address or host name of the DDNS service provider's server that DDNS updates are sent to. This should be set only if the DDNS service provider requires it.

Use the **set** form of this command to specify the server to send DDNS updates to.

Use the **delete** form of this command to use the default DDNS service provider servers.

Use the **show** form of this command to view DDNS service provider server configuration.

service dns forwarding cache-size <size>

Specifies the size of the DNS forwarding service cache.

Syntax

set service dns forwarding cache-size *size*

delete service dns forwarding cache-size

show service dns forwarding cache-size

Command Default

A maximum of 150 DNS entries are stored in the DNS forwarding cache.

Parameters

size

Optional. The maximum number of DNS entries to be held in the DNS forwarding cache. The number ranges 0 through10000, where 0 means an unlimited number of entries are stored. The default number is150.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        forwarding {
            cache-size size
        }
    }
}
```

Usage Guidelines

Use this command to specify the size of the DNS forwarding cache.

Use the **set** form of this command to specify the size of the DNS forwarding cache.

Use the **delete** form of this command to restore the default size of the DNS forwarding cache, which is 150 entries.

Use the **show** form of this command to display the size of the DNS forwarding cache.

service dns forwarding dhcp <interface>

Specifies an interface on which DHCP updates to name server information are received.

Syntax

set service dns forwarding dhcp *interface*

delete service dns forwarding dhcp *interface*

show service dns forwarding dhcp *interface*

Command Default

The system forwards DNS requests to all configured name servers and all name servers specified through DHCP.

Parameters

interface

Multinode. An interface that is to receive name server information updates from a DHCP server.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        forwarding {
            dhcp interface
        }
    }
}
```

Usage Guidelines

Use this command to specify an interface that is to act as a DHCP client and receive updates to DNS name server information. The Brocade vRouter uses this information to forward DNS requests from its local clients to the name server.

To be configured to listen for updates to name server information, the interface must be configured to obtain its own IP address through DHCP; that is, it must be configured as a DHCP client. Refer to *Brocade 5600 vRouter LAN Interfaces Reference Guide* for information about configuring the IP address of a physical interface.

By default, the DNS forwarding service creates a pool of name servers to which it forwards DNS requests; this pool comprises any name servers statically configured for the system (by using the **system name-server** command) and those of which it is notified through DHCP. This command overrides the default behavior: when an interface is specified by using this command, the system attends to DHCP name server information updates arriving on the specified interface.

This command can be combined with service dns forwarding name-server <ipv4> on page 159, service dns forwarding system on page 160, or both commands to provide a larger pool of candidate name servers.

Use the **set** form of this command to specify an interface to be used as the source of updates to the DHCP name server.

Use the **delete** form of this command to restore the default method of receiving updates to the name server, that is, the system forwards DNS requests to all configured name servers and all name servers specified through DHCP.

Use the **show** form of this command to view DNS forwarding DHCP update configuration.

service dns forwarding listen-on <interface>

Specifies an interface on which to listen for client DNS requests.

Syntax

set service dns forwarding listen-on *interface*

delete service dns forwarding listen-on *interface*

show service dns forwarding listen-on *interface*

Parameters

interface

Mandatory. Multinode. An interface on which to listen for client DNS requests.

You can specify more than one interface to receive client DNS requests by creating multiple **listen-on** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        forwarding {
            listen-on interface
        }
    }
}
```

Usage Guidelines

Use this command to specify interfaces on which to listen for client DNS requests. Only queries received on interfaces specified with this command receive DNS answers. At least one interface must be specified for DNS forwarding to operate.

Use the **set** form of this command to specify an interface on which to listen for client DNS requests.

Use the **delete** form of this command to stop an interface from listening for client DNS requests.

Use the **show** form of this command to display the configuration of client DNS request listening.

service dns forwarding name-server <ipv4>

Specifies a name server to which to forward DNS requests.

Syntax

set service dns forwarding name-server *ipv4*

delete service dns forwarding name-server *ipv4*

show service dns forwarding name-server *ipv4*

Parameters

ipv4

Optional. Multinode. The IPv4 address of a name server to which to forward DNS requests.

You can forward DNS requests to more than one name server by creating multiple **name-server** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        forwarding {
            name-server ipv4
        }
    }
}
```

Usage Guidelines

Use this command to specify a name server to which client DNS requests are forwarded.

Use of this command is optional. By default, the DNS forwarding service creates a default pool of name servers that comprises those statically configured by using the **system name-server** command plus those of which it was notified using DHCP. This command overrides the defaults: when this command is used, the system forwards DNS requests to the specified name server or servers.

This command can be combined with service dns forwarding dhcp <interface> on page 157, service dns forwarding system on page 160, or both commands to provide a larger pool of candidate name servers.

Use the **set** form of this command to specify a name server to which to forward DNS requests.

Use the **delete** form of this command to remove a name server from the list of name servers to which to forward DNS requests. If the last specified server is removed, the default forwarding behavior is restored.

Use the **show** form of this command to display the name servers to which DNS requests are forwarded.

service dns forwarding system

Specifies DNS forwarding to system-configured name servers.

Syntax

set service dns forwarding system

delete service dns forwarding system

show service dns forwarding

Modes

Configuration mode

Configuration Statement

```
service {
    dns {
        forwarding {
            system
        }
    }
}
```

Usage Guidelines

Use this command to direct the system to forward DNS requests to name servers that are statically configured by using the **system name-server** command.

By default, the DNS forwarding service forwards DNS requests to a pool of name servers that comprises the statically configured name servers plus those of which it was notified by using DHCP. This command overrides the defaults: when this command is used, DNS requests are forwarded to statically configured name servers.

This command can be combined with service dns forwarding dhcp <interface> on page 157, service dns forwarding name-server <ipv4> on page 159, or both commands to provide a larger pool of candidate name servers.

Use the **set** form of this command to specify DNS forwarding to system-configured name servers.

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

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

show dns dynamic status

Displays the update status for all hosts configured for dynamic DNS (DDNS) updates.

Syntax

show dns dynamic status

Modes

Operational mode

Usage Guidelines

Use this command to display the update status for all host names that are configured for DDNS updates.

Examples

The following example shows how to display the update status for hosts that are configured for DDNS updates.

```
vyatta@R1> show dns dynamic status
show dns dynamic status
interface    : dp0p1p3
ip address   : 1.2.3.4
host-name    : test1.getmyip.com
last update  : Thu Sep 11 19:30:43 2008
update-status: good

interface    : dp0p1p3
ip address   : 1.2.3.5
host-name    : test2.getmyip.com
last update  : Thu Sep 11 19:30:43 2008
update-status: good

interface    : dp0p1p4
ip address   : 1.3.4.5
host-name    : test4
last update  : Thu Sep 11 19:34:16 2008
update-status: good
vyatta@R1>
```

show dns forwarding nameservers

Displays the name servers that are being used for DNS forwarding.

Syntax **show dns forwarding nameservers**

Modes Operational mode

Usage Guidelines Use this command to display the name servers that are being used for DNS forwarding and those that are available but are not being used for DNS forwarding.

Examples The following example shows how to display the name servers that are being used for DNS forwarding.

```
vyatta@R1> show dns forwarding nameservers
-----------------------------------------------
    Nameservers configured for DNS forwarding
-----------------------------------------------
10.0.0.30 available via 'system'
-----------------------------------------------
    Nameservers NOT configured for DNS forwarding
-----------------------------------------------
10.0.0.31 available via 'dhcp dp0p1p4'
vyatta@R1>
```

show dns forwarding statistics

Displays DNS forwarding statistics.

Syntax **show dns forwarding statistics**

Modes Operational mode

Usage Guidelines Use this command to display statistics related to DNS forwarding. The statistics restart each time a change occurs in name servers from any source (DHCP, system, or statically configured), a change in static host mapping (by using the **system static-host-mapping** command), or a change made to the DNS forwarding configuration.

Examples The following example shows how to display DNS forwarding statistics.

```
vyatta@R1> show dns forwarding statistics
----------------
Cache statistics
----------------
Cache size: 150
Queries forwarded: 5
Queries answered locally: 2
Total DNS entries inserted into cache: 23
DNS entries removed from cache before expiry: 0
---------------------
Nameserver statistics
---------------------
Server: 10.0.0.30
Queries sent: 5
Queries retried or failed: 0
vyatta@R1>
```

update dns dynamic interface <interface>

Sends a forced update to a dynamic DNS (DDNS) service provider on a specific interface.

Syntax
update dns dynamic interface *text*

Parameters
interface

An interface from which to send the forced update.

Modes
Operational mode

Usage Guidelines
Use this command to manually initiate a forced update to a DDNS service provider. The forced update provides the DDNS service provider with the status of the specified interface.

Note that this command should be used sparingly because frequent unnecessary updates could cause the host name to be blocked by the DDNS service provider.

Flow Monitoring

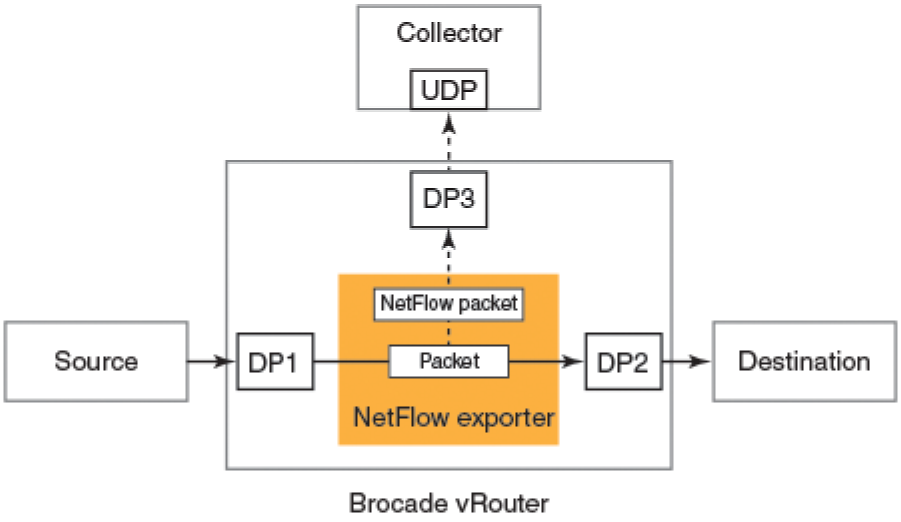
Flow monitoring overview

The Flow Monitoring service allows network administrators to collect IP flow information from Brocade vRouter.

Brocade vRouters support the NetFlow Version 9 export format, which is based on the Internet Engineering Task Force (IETF) standard. This standard defines how IP flow information is formatted and transferred from an exporter (in this case, a Brocade vRouter) to a NetFlow collector, a system that collects IPv4 and IPv6 flow information, as shown in the following figure.

The exporter periodically collects information about packets that flow through the router into a flow record. Then, the exporter packs the record into a UDP packet and sends it to the collector. For more information about the NetFlow Version 9 export format, refer to RFC 3954.

FIGURE 10 Brocade vRouter flow-monitoring architecture



NOTE
You can configure flow monitoring only for ingress traffic that is flowing through a dataplane interface.

The NetFlow Version 9 export format is based on templates, which allows for an extensible design of the record format. This means that future enhancements of the Flow Monitoring service do not require changes to the export protocol.

Table 19 describes the data that is extracted by the Flow Monitoring service from the sampled packets and exported to the NetFlow collector.

TABLE 19 Exported NetFlow version 9 IPv4 and IPv6 data

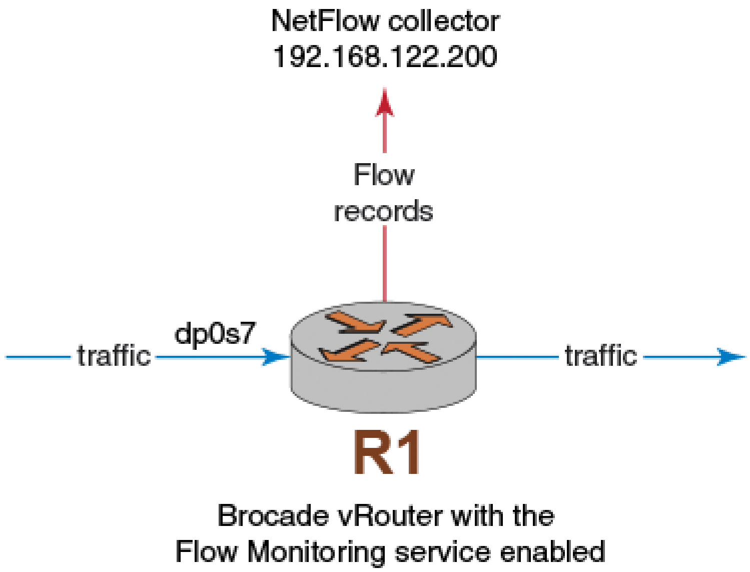
Field	Key or Nonkey Field	Description
IP ToS	Key	Type of service (ToS) value.
IP Protocol	Key	IP protocol.
IP Source Address	Key	Source IP address.
IP Destination Address	Key	Destination IP address.
Transport Source Port	Key	Source port number.
Transport Destination Port	Key	Destination port number.
Interface Input	Key	Dataplane interface on which the traffic is received.
Flow Sampler ID	Key	ID number of the flow sampler (if flow sampling is enabled).
IP Source AS	Nonkey	Source autonomous system number.
IP Destination AS	Nonkey	Destination autonomous system number.
IP Next Hop Address	Nonkey	IP address of the next hop.
IP Source Mask	Nonkey	Mask for the IP source address.
IP Destination Mask	Nonkey	Mask for the IP destination address.
TCP Flags	Nonkey	TCP flag.
Interface Output	Nonkey	Interface through which the traffic travels to its destination.
Counter Bytes	Nonkey	Number of bytes in the flow.
Counter Packets	Nonkey	Number of packets in the flow.
Time Stamp System Uptime First	Nonkey	System uptime (time, in milliseconds, since the service was restarted) when the first packet was switched.
Time Stamp System Uptime Last	Nonkey	System uptime (time, in milliseconds, since the service was restarted) when the last packet was switched.

NOTE
Currently, the IP Source AS, IP Destination AS, IP Source Mask, and IP Destination Mask fields are set to 0. In a future version, these fields will be set correctly.

Flow monitoring configuration

Table 20 shows how to configure the Flow Monitoring service on the R1 Brocade vRouter, as shown in the example in the following figure.

FIGURE 11 Brocade vRouter flow monitoring configuration example



To configure the Flow Monitoring service on R1, perform the following steps in configuration mode.

TABLE 20 Configuring the flow monitoring service

Step	Command

Create a flow-monitoring exporter and specify the IP address and UDP port of the NetFlow collector.
The Brocade vRouter supports only a single unnamed exporter. You cannot create multiple exporters.
	`vyatta@R1# set service flow-monitoring exporter udp-collector address 192.168.122.200`
	`vyatta@R1# set service flow-monitoring exporter udp-collector port 9995`

Create a packet selector named s1 and specify a sampling size of 100 packets. The sampling rate in this case is 1:100, which means that the router randomly selects one packet from every 100 packets that flow through any dataplane interface with which the selector is associated.	`vyatta@R1# set service flow-monitoring selector s1 randomly out-of 100`
Apply the s1 selector to the dp0s7 dataplane interface. The selector samples IPv4 and IPv6 flows passing through dp0s7.	`vyatta@R1# set interfaces dataplane dp0s7 flow-monitoring selector s1`
Commit the configuration.	`vyatta@R1# commit`
Save the configuration.	`vyatta@R1# save`

TABLE 20 Configuring the flow monitoring service (Continued)

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Verify the configuration.</td><td><pre>
vyatta@R1# show interfaces
interfaces {
    dataplane dp0s7 {
        address 10.0.1.220/24
        flow-monitoring {
            selector s1
        }
    }
    loopback lo
}
vyatta@R1# show services
service {
    flow-monitoring {
        selector s1 {
            randomly {
                out-of 100
            }
        }
        exporter {
            udp-collector {
                address 192.168.122.200
                port 9995
            }
        }
    }
}
</pre></td></tr>
<tr><td>Exit the configuration mode.</td><td><pre>
vyatta@R1# exit
</pre></td></tr>
<tr><td>Show exporter statistics.</td><td><pre>
vyatta@R1:~$ show flow-monitoring
dataplane statistics:
    interface dp0s7:
        monitor default:
            packets observed: 884
            samples taken: 88

export daemon statistics:
  /exporter/default:
    /monitor/mon1/ipv4:
      reports received: 88
      reports exported: 88
    /monitor/mon1/ipv6:
      reports received: 0
      reports exported: 0
</pre></td></tr>
</table>

NOTE
If a dataplane interface fails to gather samples because of resource constraints, show flow-monitoring on page 175 displays the number of lost samples in the samples lost field (under the samples taken field) in the output.

Flow Monitoring Commands

interfaces dataplane <interface> flow-monitoring selector <selector-name>

Associates a packet selector with a dataplane interface through which the traffic to be monitored flows.

Syntax

set interfaces dataplane *interface* **flow-monitoring selector** *selector-name*

delete interfaces dataplane *interface* **flow-monitoring selector** *selector-name*

show interfaces dataplane *interface* **flow-monitoring**

Parameters

interface

A dataplane interface. For more information about dataplane interfaces, refer to *LAN Interfaces Reference Guide*.

selector-name

The name of a packet selector.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    dataplane interface {
        flow-monitoring {
            selector selector-name
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to associate a packet selector with a dataplane interface through which the traffic to be monitored flows.

Use the **delete** form of this command to disassociate a packet selector from a dataplane interface.

Use the **show** form of this command to display the name of the configured selector for a dataplane interface.

service flow-monitoring exporter udp-collector address <ip-address>

Specifies the IPv4 or IPv6 address of the NetFlow collector.

Syntax

set service flow-monitoring exporter udp-collector address *ip-address*

delete service flow-monitoring exporter udp-collector address *ip-address*

show service flow-monitoring exporter udp-collector

Parameters

ip-address

The IPv4 or IPv6 address of the collector.

Modes

Configuration mode

Configuration Statement

```
service {
    flow-monitoring {
        exporter {
            udp-collector {
                address ip-address
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the IPv4 or IPv6 address of the NetFlow collector.

Use the **delete** form of this command to remove the IP address of the NetFlow collector.

Use the **show** form of this command to display the configured IP address of the NetFlow collector.

service flow-monitoring exporter udp-collector port <udp-port>

Specifies the UDP port of the NetFlow collector.

Syntax

set service flow-monitoring exporter udp-collector port *udp-port*

delete service flow-monitoring exporter udp-collector port *udp-port*

show service flow-monitoring exporter udp-collector

Command Default

The default UDP port number is 9995.

Parameters

udp-port

The UDP port number of the collector. The number ranges from 0 through 65535.

Modes

Configuration mode

Configuration Statement

```
service {
    flow-monitoring {
        exporter {
            udp-collector {
                port udp-port
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the UDP port of the NetFlow collector.

Use the **delete** form of this command to specify the default UDP port number, which is 9995.

Use the **show** form of this command to display the configured UDP port for the NetFlow collector.

service flow-monitoring selector <selector-name> randomly out-of <num-of-packets>

Creates a random-packet selector and specifies the size of the packet sample window from which to select a packet.

Syntax

set service flow-monitoring selector *selector-name* **randomly out-of** *num-of-packets*

delete service flow-monitoring selector *selector-name* **randomly out-of** *num-of-packets*

show service flow-monitoring selector

Command Default

The default size of the packet sample is 1000.

Parameters

selector-name

The name of the packet selector.

num-of-packets

The size of the packet sample window from which to select a packet. The size ranges from 10 through 10000.

Modes

Configuration mode

Configuration Statement

```
service {
    flow-monitoring {
        selector selector-name {
            randomly {
                out-of num-of-packets
            }
        }
    }
}
```

Usage Guidelines

This command randomly samples packets. For example, for a 1:100 sampling, the Brocade vRouter randomly calculates a number from 1 through 100, say 63. When the sixty-third packet arrives at the dataplane interface that is being monitored, the router extracts the NetFlow packet details and sends them to the NetFlow collector. The router ignores the remaining 37 packets of the sampling window (packets 64 through 100), then the router repeats the process of sampling packets.

Use the **set** form of this command to create a random-packet selector and specify the sample size.

Use the **delete** form of this command to delete the random-packet selector.

Use the **show** form of this command to display the details of the configured random-packet selector.

clear flow-monitoring

Clears the flow-monitoring statistics.

Syntax **clear flow-monitoring**

Modes Operational mode

Usage Guidelines Use this command to clear the flow-monitoring statistics that have been gathered by the Flow Monitoring service.

show flow-monitoring

Displays the flow-monitoring statistics.

Syntax **show flow-monitoring**

Modes Operational mode

Usage Guidelines Use this command to display the flow-monitoring statistics that have been gathered by the Flow Monitoring service.

NOTE
In the output of this command, if a 64-bit integer counter wraps, an asterisk (*) appears after the counter.

LLDP

LLDP overview

LLDP is an open standard for network devices to communicate link-layer topology and connection endpoint information on IEEE 802 (Ethernet) LANs and MANs. LLDP is described in the IEEE standards document 802.1AB, *Station and Media Access Control Connectivity Discovery*. It allows a station on the network to advertise information about its capabilities, configuration, and identity to other LLDP-enabled stations on the same physical network. This information is stored in the device as a standard management information base (MIB) as specified in RFC 2922. A network management system can query these MIBs using SNMP to model the topology of the network.

Configuring LLDP

To enable LLDP on a Brocade vRouter, you must enable the service by setting its configuration node, as in the following example.

TABLE 21 Enabling LLDP on a system

Step	Command
Create the LLDP service configuration node.	`vyatta@vyatta# set service lldp`
Commit the configuration.	`vyatta@vyatta# commit`

After the service is enabled, you can record information about the location, management address, and port of the device and the legacy protocols it supports. Additional information, including the configured capabilities and neighbors of the system, is extracted automatically from the system and stored in a MIB.

The following example shows how to configure a civic-based location for the system, as follows:

```
Suite 200 - 1301 Shoreway Road
Belmont, CA, USA
94002-4157
```

TABLE 22 Configuring a civic-based location for LLDP

Step	Command
Configure the language.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 0 ca-value English`

TABLE 22 Configuring a civic-based location for LLDP (Continued)

Step	Command
Configure the occupant.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 23 ca-value "Vyatta, Inc. Corporate Headquarters"`
Configure the suite number.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 26 ca-value “Suite 200”`
Configure the floor number.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 27 ca-value 2nd`
Configure the street address.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 6 ca-value “1301 Shoreway Road”`
Configure the city.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 3 ca-value Belmont`
Configure the country.	`vyatta@vyatta#set service lldp interface dp0p1p1 location country-code US`
Configure the ZIP code.	`vyatta@vyatta#set service lldp interface dp0p1p1 location civic-based ca-type 24 ca-value 94002-4157`
Commit the configuration.	`vyatta@vyatta#commit`

Show the LLDP configuration.	vyatta@vyatta#show service lldp
	
	 interface dp0p1p1 {
	    location {
	        civic-based {
	            ca-type 0 {
	                ca-value English
	            }
	            ca-type 3 {
	                ca-value Belmont
	            }
	            ca-type 6 {
	                ca-value “1301 Shoreway Road”
	            }
	            ca-type 23 {
	                ca-value "Vyatta, Inc. Corporate Headquarters"
	            }
	            ca-type 24 {
	                ca-value 94002-4157
	            }
	            ca-type 26 {
	                ca-value "Suite 200"
	            }
	            ca-type 27 {
	                ca-value 2nd
	            }
	            country-code US
	        }
	    }
	 }

The location can be coordinate-based rather than civic-based, as shown in the following example.

TABLE 23 Configuring the physical coordinates of a system

Step	Command
Configure the latitude coordinate.	`vyatta@vyatta#set service lldp interface dp0p1p4 location coordinate-based latitude 37.524449N`
Configure the longitude coordinate.	`vyatta@vyatta#set service lldp interface dp0p1p4 location coordinate-based longitude 122.267255W`
Commit the configuration.	`vyatta@vyatta#commit`

TABLE 23 Configuring the physical coordinates of a system (Continued)

Step	Command

Show the LLDP configuration for the dp0p1p4 interface.	`vyatta@vyatta#show service lldp interface dp0p1p4`
	` location {`
	`    coordinate-based {`
	`        latitude 37.524449N`
	`        longitude 122.267255W`
	`    }`
	` }`

Displaying LLDP information

When the system is enabled for LLDP, it can gather and display information about link-layer neighbors, as shown below.

Showing LLDP neighbors

```
vyatta@vyatta:~$show lldp neighbors

Capability Codes: R - Router, B - Bridge, W - Wlan r - Repeater, S - Station
                  D - Docsis, T - Telephone, O - Other

Device ID        Local     Proto  Cap   Platform             Port ID
---------        -----     -----  ---   --------             -------
R1               dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
R2               dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
HNF-BFD2         dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
HNF-BF           dp0p192p1 LLDP   RS    Vyatta Router      dp0p192p1
HNF-BFD2         dp0p224p1 LLDP   RS    Vyatta Router      dp0p224p1
```

The following example shows detailed information on LLDP neighbors.

Showing detailed information on LLDP neighbors

```
vyatta@vyatta:~$show lldp neighbors detail

-------------------------------------------------------------------------------
LLDP neighbors:
-------------------------------------------------------------------------------
Interface:    dp0p160p1, via: LLDP, RID: 1, Time: 0 day, 01:12:52
  Chassis:
         ChassisID:    mac 00:0c:29:e6:b1:7d
         SysName:      R1
         SysDescr:     Vyatta Router running on Brocade 5600 vRouter 3.5 R5
         MgmtIP:       10.37.108.1
         MgmtIP:       3000:1:2::1
         Capability:   Bridge, off
         Capability:   Router, on
         Capability:   Wlan, off
         Capability:   Station, on
  Port:
         PortID:       mac 00:0c:29:e6:b1:7d
         PortDescr:    dp0p160p1
         PMD autoneg:  supported: no, enabled: yes
           MAU oper type: 10GigBaseR - R PCS/PMA, unknown PMD.
  LLDP-MED:
        Device Type:  Network Connectivity Device
        Capability:   Capabilities
        Capability:   Policy
        Capability:   Location
        Capability:   MDI/PSE
        Capability:   MDI/PD
        Capability:   Inventory
        Inventory:
          Hardware Revision: None
          Software Revision: 3.14.48-1-amd64-vyatta
          Firmware Revision: 6.00
          Serial Number: VMware-56 4d dd 2d b2 65 00 db-1
          Manufacturer: VMware, Inc.
          Model:        VMware Virtual Platform
          Asset ID:     No Asset Tag
-------------------------------------------------------------------------------
Interface:    dp0p160p1, via: LLDP, RID: 2, Time: 0 day, 01:12:52
  Chassis:
    ChassisID:    mac 00:0c:29:44:dc:65
    SysName:      R2
    SysDescr:     Vyatta Router running on Brocade 5600 vRouter 3.5 R5
    MgmtIP:       10.37.108.3
    MgmtIP:       3000:2:4::1
    Capability:   Bridge, off
    Capability:   Router, on
    Capability:   Wlan, off
    Capability:   Station, on
  Port:
    PortID:       mac 00:0c:29:44:dc:65
    PortDescr:    dp0p160p1
    PMD autoneg:  supported: no, enabled: yes
      MAU oper type: 10GigBaseR - R PCS/PMA, unknown PMD.
  LLDP-MED:
    Device Type:  Network Connectivity Device
    Capability:   Capabilities
    Capability:   Policy
    Capability:   Location
    Capability:   MDI/PSE
    Capability:   MDI/PD
    Capability:   Inventory
  Inventory:
    Hardware Revision: None
    Software Revision: 3.14.48-1-amd64-vyatta
    Firmware Revision: 6.00
    Serial Number: VMware-56 4d 92 ca 84 63 bc f2-a
    Manufacturer: VMware, Inc.
    Model:        VMware Virtual Platform
    Asset ID:     No Asset Tag
-------------------------------------------------------------------------------
```

LLDP Commands

service lldp

Enables the LLDP service.

Syntax

set service lldp

delete service lldp

show service lldp

Modes

Configuration mode

Configuration Statement

```
service {
    lldp {}
}
```

Usage Guidelines

Use this command to enable LLDP on the system.

Use the **set** form of this command to enable the LLDP service.

Use the **delete** form of this command to disable the LLDP service.

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

service lldp interface <interface> location civic-based

Records a civic-based location for an LLDP-enabled device.

Syntax **set service lldp interface** *interface* **location civic-based** [**ca-type** *type* **ca-value** *value* | **country-code** *code*]

delete service lldp interface *interface* **location civic-based ca-type** *type* **ca-value**

show service lldp interface *interface* **location civic-based ca-type** *type* **ca-value**

Parameters *interface*

An interface to which the configured information applies. The name of any IEEE 802.1-compatible interface or the **all** keyword, where **all** refers to all 802.1-compatible interfaces.

ca-type *type*

Multinode. Records a civic address type as defined in the ANSI document. The type is one of the following:

- 0 — Language
- 1 — National subdivisions
- 2 — County, parish, district
- 3 — City, township
- 4 — City division, borough, ward
- 5 — Neighborhood, block
- 6 — Street
- 16 — Leading street direction
- 17 — Trailing street suffix
- 18 — Street suffix
- 19 — House number
- 20 — House number suffix
- 21 — Landmark or vanity address
- 22 — Additional location info
- 23 — Name
- 24 — Postal or ZIP code
- 25 — Building
- 26 — Unit
- 27 — Floor
- 28 — Room number
- 29 — Place type
- 128 — Script

You can record multiple civic address components by creating multiple **ca-type** and **ca-value** pairs, where each pair represents a different component.

ca-value *value*

The value for the specified civic address type.

country-code *code*

A two-letter code, as defined in ISO 3166, representing the country in which the device is located.

Modes Configuration mode

Configuration Statement

```
service {
    lldp {
        interface interface {
            location {
            civic-based {
                ca-type type {
                    ca-value value
                }
                country-code code
            }
        }
    }
}
```

Usage Guidelines

Use this command to record the components of a civic address identifying the location of the device. A civic address-based location requires a country code and at least one **ca-type** and **ca-value** pair.

Use the **set** form of this command to specify a component of a civic address.

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

Use the **show** form of this command to show civic address component configuration.

service lldp interface <interface> location coordinate-based

Records a coordinate-based location for an LLDP-enabled device.

Syntax

set service lldp interface *interface* **location coordinate-based** [**altitude** *altitude* | **datum** *datum* | **latitude** *latitude* | **longitude** *longitude*]

delete service lldp interface *interface* **location coordinate-based** [**altitude** | **datum** | **latitude** | **longitude**]

show service lldp interface *interface* **location coordinate-based** [**altitude** | **datum** | **latitude** | **longitude**]

Command Default

No location is configured.

Parameters

interface

An interface to which the configured information applies. The name of any IEEE 802.1-compatible interface can be specified or the **all** keyword, where **all** refers to all 802.1-compatible interfaces.

altitude *altitude*

Specifies the altitude, in meters, of the device. The default altitude is 0.

datum *datum*

Specifies the reference datum for the coordinate system. The datum is **WGS84**, **NAD83**, or **MLLW**. The default datum is **WGS84**.

latitude *latitude*

Specifies the latitude of the device. The format for the latitude is *deg.minD*. where *deg* is degrees, *min* is minutes to any level of precision, and *D* is N or S, representing North or South, respectively; for example 37.524449N.

longitude *longitude*

The longitude of the device. The format is *deg.minD* . where *deg* is degrees, *min* is minutes to any level of precision, and *D* is E or W, representing East or West, respectively; for example, 122.267255W.

Modes

Configuration mode

Configuration Statement

```
service {
    lldp {
        interface interface {
            location {
                coordinate-based {
                    altitude altitude
                    datum datum
                    latitude latitude
                    longitude longitude
                }
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify a location for a device based on its coordinates.

When a coordinate-based location is used, both the latitude and longitude must be configured; other values are optional.

Use the **set** form of this command to specify a coordinate-based location.

Use the **delete** form of this command to remove a coordinate-based location and restore any default values.

Use the **show** form of this command to show coordinate-based configuration.

service lldp interface <interface> location elin <phone-num>

Records an emergency line identification number (ELIN) for an LLDP-enabled device.

Syntax

set service lldp interface *interface* **location elin** *phone-num*

delete service lldp interface *interface* **location elin**

show service lldp interface *interface* **location elin**

Parameters

interface

An interface to which the configured information applies. The name of any IEEE 802.1-compatible interface or the **all** keyword, where **all** refers to all 802.1-compatible interfaces.

phone-num

An emergency line identification number. The number is a 10-to-25-digit phone number. A phone number with fewer than 10 digits must be padded with zeros; for example, 911 must be represented as 0000000911.

Modes

Configuration mode

Configuration Statement

```
service {
    lldp {
        interface interface {
            location {
                elin phone-num
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify an emergency call service ELIN.

Use the **set** form of this command to record an ELIN.

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

Use the **show** form of this command to display the ELIN.

service lldp legacy-protocols <protocol>

Specifies which legacy (proprietary) link-layer discovery protocols to support in addition to LLDP.

Syntax

set service lldp legacy-protocol *protocol*

delete service lldp legacy-protocol *protocol*

show service lldp legacy-protocol

Parameters

protocol

Multinode. A proprietary link-layer discovery protocol. The protocol is one of the following:

cdp —Cisco Discovery Protocol

edp —Extreme Discovery Protocol

fdp —Foundry Discovery Protocol

sonmp —Nortel Discovery Protocol

You can enable support for multiple legacy protocols by creating multiple legacy-protocol configuration nodes.

Modes

Configuration mode

Configuration Statement

```
service {
    lldp {
        legacy-protocol protocol
    }
}
```

Usage Guidelines

Use this command to specify a legacy (proprietary) link-layer discovery protocol to support in addition to LLDP.

If a frame from one of the specified protocols is received on an interface, the system begins to send frames for that protocol on the interface.

Use the **set** form of this command to enable support for a legacy protocol.

Use the **delete** form of this command to disable support for a legacy protocol.

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

service lldp management-address <ipv4>

Records the management address of the system.

Syntax

set service lldp management-address *ipv4*

delete service lldp management-address

show service lldp management-address

Command Default

The system automatically determines which address to advertise as the management address.

Parameters

ipv4

The IP address of the management system. The address must be an IPv4 address.

Modes

Configuration mode

Configuration Statement

```
service {
    lldp {
        management-address ipv4
    }
}
```

Usage Guidelines

Use this command to specify the IP address to be advertised as the management address by LLDP.

Use the **set** form of this command to set the management address.

Use the **delete** form of this command to restore the default behavior, that is, the system automatically determines which address to advertise as the management address.

Use the **show** form of this command to show LLDP management address configuration.

show lldp neighbors

Displays a summary of link layer neighbors that are running LLDP.

Syntax **show lldp neighbors** [**detail**]

Command Default When used with no option, this command displays a summary of information about link layer neighbors.

Parameters **detail**

Optional. Displays detailed information about link layer neighbors.

Modes Operational mode

Usage Guidelines Use this command to display information about link layer neighbors that are running LLDP.

If support for any legacy link-layer discovery protocol has been enabled (using service lldp legacy-protocols <protocol> on page 188), the system also displays neighbors discovered using that protocol.

Examples The following example shows how to display a summary of link layer neighbors.

```
vyatta@vyatta:~$show lldp neighbors

Capability Codes: R - Router, B - Bridge, W - Wlan r - Repeater, S - Station
                  D - Docsis, T - Telephone, O - Other

Device ID        Local     Proto  Cap   Platform             Port ID
---------        -----     -----  ---   --------             -------
R1               dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
R2               dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
HNF-BFD2         dp0p160p1 LLDP   RS    Vyatta Router      dp0p160p1
HNF-BF           dp0p192p1 LLDP   RS    Vyatta Router      dp0p192p1
HNF-BFD2         dp0p224p1 LLDP   RS    Vyatta Router      dp0p224p1
```

The following example shows how to display details of link layer neighbors.

```
vyatta@vyatta:~$show lldp neighbors detail
-------------------------------------------------------------------------------
LLDP neighbors:
-------------------------------------------------------------------------------
Interface:    dp0p1p1, via: CDPv1, RID: 3, Time: 0 day, 00:19:34
  Chassis:
    ChassisID:    local medusa
    SysName:      medusa
    SysDescr:     cisco 2511 running on
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-I-L), Version 12.0(14), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Tue 31-Oct-00 23:59 by linda
    MgmtIP:       10.1.0.9
    Capability:   Router, on
  Port:
    PortID:       ifname Ethernet0
    PortDescr:    Ethernet0
-------------------------------------------------------------------------------
Interface:    dp0p1p1, via: LLDP, RID: 4, Time: 0 day, 00:19:28
  Chassis:
    ChassisID:    mac 00:1b:21:44:70:44
    SysName:      tethys
    SysDescr:     Vyatta Series 3500 running on 999.larkspurse.04270036
    MgmtIP:       10.1.0.40
    Capability:   Bridge, off
    Capability:   Router, on
    Capability:   Wlan, off
  Port:
    PortID:       mac 00:24:e8:7b:ca:6c
    PortDescr:    dp0p1p1
    PMD autoneg:  supported: yes, enabled: yes
      Adv:          10Base-T, HD: yes, FD: yes
      Adv:          100Base-T, HD: yes, FD: yes
      Adv:          1000Base-T, HD: no, FD: yes
      MAU oper type: 100BaseTXFD - 2 pair category 5 UTP, full duplex mode
  LLDP-MED:
    Device Type:  Network Connectivity Device
    Capability:   Capabilities
    Capability:   Location
    Capability:   Inventory
    LLDP-MED Location Identification: Type: elin
      ECS ELIN:     0000000911
    Inventory:
      Software Revision: 2.6.32-1-586-vyatta
      Firmware Revision: 2.0.11
      Serial Number: JGSM3K1
      Manufacturer: Vyatta
      Model:        Series 3500
-------------------------------------------------------------------------------
Interface:    dp0p1p4, via: LLDP, RID: 6, Time: 0 day, 00:00:03
  Chassis:
    ChassisID:    mac 00:0c:29:8c:53:7c
    SysName:      R1
    SysDescr:     Vyatta Router running on Vyatta Subscription Edition 6.0 2010.03.22
    MgmtIP:       20.0.0.2
    Capability:   Bridge, off
    Capability:   Router, on
    Capability:   Wlan, off
  Port:
    PortID:       ifname dp0p1p1
    PortDescr:    bridge
    PMD autoneg:  supported: yes, enabled: yes
      Adv:          10Base-T, HD: yes, FD: yes
      Adv:          100Base-T, HD: yes, FD: yes
      Adv:          1000Base-T, HD: no, FD: yes
      MAU oper type: 1000BaseTFD - Four-pair Category 5 UTP, full duplex mode
  LLDP-MED:
    Device Type:  Network Connectivity Device
    Capability:   Capabilities
    Capability:   Location
    Capability:   Inventory
    LLDP-MED Location Identification: Type: address
      Country:      US
      Language:     English
      City, township: Belmont
      Street:       1301 Shoreway Road
      Name:         Vyatta, Inc. Corporate Headquarters
      Postal/ZIP code: 94002-4157
      Floor:        2nd
      Room number:  Suite 200
    Inventory:
      Hardware Revision: None
      Software Revision: 2.6.32-1-586-vyatta-virt
      Firmware Revision: 6.00
      Serial Number: VMware-56 4d 6b 88 64 cc 44 27-2
      Manufacturer: VMware, Inc.
      Model:        VMware Virtual Platform
      Asset ID:     No Asset Tag
-------------------------------------------------------------------------------
```

NHRP

NHRP overview

Next Hop Resolution Protocol (NHRP) is a software-addressing service commonly used in nonbroadcast multiaccess (NBMA) networks. In an NBMA network, NHRP provides the mapping between the NBMA next hop and the Layer 3 subnetwork address.

NHRP and NBMA networks

An NBMA network connects multiple hosts, but has no broadcast or multicast capability; data is transmitted only from one device to one other device. For full connectivity in such a network, connections must be meshed in some way.

A full mesh of connections is difficult to scale, so NBMA networks often employ some version of a hub-and-spoke network to reduce the complexity of the network. However, hub-and-spoke networks have challenges of their own.

- The hub becomes a single point of failure for the network.
- All network traffic passes through the hub, which becomes a processing bottleneck.

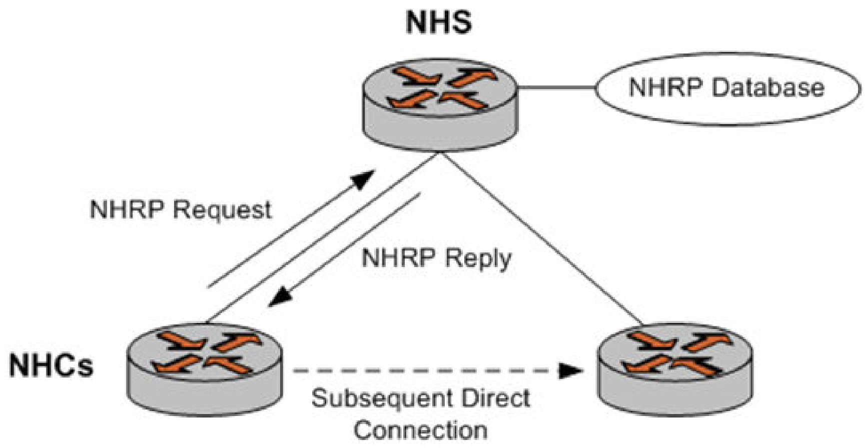
NHRP allows you to reduce the number of paths through the network and reduces the need for static configuration, helping provide the connectivity of a full mesh but greater scalability.

NHSs and NHCs

In an NHRP-enabled network, a router is configured with NHRP as a next-hop server (NHS). The NHS becomes a kind of route server, maintaining an NHRP database mapping the NBMA next hop to IP addresses.

The NHRP database is dynamically built and kept accurate using an ARP-like query-and-reply mechanism. Devices that need to communicate register dynamically with the NHS as a next-hop client (NHC). Having determined the existence of the NHC, the NHS adds it to the NBMA network without configuration. The NHC dynamically determines the locations of the other devices in the network from the NHS. The first communication of an NHC to a given device in the network initially flows through the NHS. During its first communication to a destination device, the NHC queries for the IP address of the destination device from the NHS and determines it. Thereafter, the NHC initiates a dynamic GRE/IPSec tunnel to the destination device and traffic flows directly from the NHC to the device. The tunnel is torn down when traffic is no longer flowing between the two devices.

FIGURE 12 NHRP



Supported standards

The Brocade 5600 vRouter implementation of NHRP complies with the RFC 2332: NBMA Next Hop Resolution Protocol (NHRP) standard.

NHRP configuration

NHRP is frequently used with multipoint Generic Routing Encapsulation (mGRE) to create networks of dynamically built point-to-point tunnels. When these tunnels are secured with IP Security (IPsec), the result is a dynamic multipoint virtual private network (DMVPN).

- Configuration examples using NHRP with mGRE are provided in *Brocade 5600 vRouter Tunnels Reference Guide*.
- Configuration examples using NHRP in a DMVPN are provided in *Brocade 5600 vRouter DMVPN Reference Guide*.

NHRP Commands

interfaces tunnel <tunx> nhrp

Enables NHRP on a tunnel interface.

Syntax

set interfaces tunnel *tunx* **nhrp**

delete interfaces tunnel *tunx* **nhrp**

show interfaces tunnel *tunx* **nhrp**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp
    }
}
```

Usage Guidelines

Use this command to enable NHRP on a tunnel interface.

Use the **set** form of this command to enable NHRP on a tunnel interface.

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

Use the **show** form of this command to display NHRP configuration on a tunnel interface.

interfaces tunnel <tunx> nhrp authentication <secret>

Specifies a password that authenticates NHRP packets.

Syntax

set interfaces tunnel *tunx* **nhrp authentication** *secret*

delete interfaces tunnel *tunx* **nhrp authentication**

show interfaces tunnel *tunx* **nhrp authentication**

Command Default

NHRP packets are not authenticated.

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

secret

Mandatory. A plain text password that authenticates packets. The password is a maximum of eight characters.

Modes

Configuration mode

Configuration Statement

```
interfaces {
   tunnel tunx {
      nhrp {
            authentication secret
      }
   }
}
```

Usage Guidelines

Use this command to specify a password that authenticates NHRP packets. The password is embedded in all outgoing NHRP packets. All incoming NHRP packets on the interface are discarded unless the password is present.

Use the **set** form of this command to specify a password that authenticates NHRP packets.

Use the **delete** form of this command to delete an authentication password.

Use the **show** form of this command to display an authentication password.

interfaces tunnel <tunx> nhrp dynamic-map nbma-domain <nbma-domain-name>

Specifies that protocol address-to-nonbroadcast multiaccess (NBMA) address mappings are to be determined dynamically by using the next-hop server whose fully qualified domain name (FQDN) is specified.

Syntax

set interfaces tunnel *tunx* **nhrp dynamic-map nbma-domain** *nbma-domain-name*

delete interfaces tunnel *tunx* **nhrp dynamic-map nbma-domain** *nbma-domain-name*

show interfaces tunnel *tunx* **nhrp dynamic-map nbma-domain**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

nbma-domain-name

Mandatory. The FQDN of the next-hop server.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            dynamic-map {
                nbma-domain nbma-domain-name
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify that protocol address-to-NBMA address mappings are to be determined dynamically by using the next-hop server whose FQDN is specified.

Use the **set** form of this command to specify the FQDN of the next-hop server.

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

Use the **show** form of this command to display the FQDN of the next-hop server.

interfaces tunnel <tunx> nhrp dynamic-map protocol-address <protocol-addr>/<prefix>

Specifies that protocol address-to-NBMA address mappings are to be determined dynamically by using the next-hop server whose protocol address is specified.

Syntax

set interfaces tunnel *tunx* **nhrp dynamic-map protocol-address** *protocol-addr/prefix*

delete interfaces tunnel *tunx* **nhrp dynamic-map protocol-address** *protocol-addr/prefix*

show interfaces tunnel *tunx* **nhrp dynamic-map protocol-address**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of the next-hop server. The protocol address is an IPv4 address.

prefix

Mandatory. The protocol address prefix of the next-hop server. The protocol address prefix is an IPv4 prefix.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            dynamic-map {
                protocol-address protocol-addr/prefix
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify that protocol address-to-NBMA address mappings are to be determined dynamically using the next-hop server whose NBMA IP address is specified.

Use the **set** form of this command to specify the NBMA IP address of the next-hop server.

Use the **delete** form of this command to remove the NBMA IP address of the next-hop server.

Use the **show** form of this command to display the NBMA IP address of the next-hop server.

interfaces tunnel <tunx> nhrp holding-time <time>

Specifies the hold time for NHRP Registration requests and Resolution replies sent from an interface or a shortcut target.

Syntax

set interfaces tunnel *tunx* **nhrp holding-time** *time*

delete interfaces tunnel *tunx* **nhrp holding-time**

show interfaces tunnel *tunx* **nhrp holding-time**

Command Default

The hold time is 7,200 seconds (two hours).

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

time

Mandatory. The hold time in seconds. The default time is 7200 seconds (two hours).

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            holding-time time
        }
    }
}
```

Usage Guidelines

Use this command to specify the hold time for NHRP Registration requests and Resolution replies sent from an interface or a shortcut target. The hold time is the amount of time that the system retains its knowledge of protocol address-to-NBMA address mappings. If not refreshed within the hold time, the mappings are discarded and need to be re-established.

Use the **set** form of this command to specify the hold time for NHRP Registration requests and Resolution replies sent from an interface or a shortcut target.

Use the **delete** form of this command to delete the hold time.

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

interfaces tunnel <tunx> nhrp map <protocol-addr>/<prefix> nbma-address <nbma-addr>

Statically maps a protocol address or protocol address prefix to the NBMA address of an NBMA peer.

Syntax

set interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix* **nbma-address** *nbma-addr*

delete interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix* **nbma-address** *nbma-addr*

show interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix* **nbma-address**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of a destination. The protocol address is an IPv4 address.

prefix

Mandatory. The protocol address prefix of the destination. The protocol address prefix is an IPv4 prefix.

nbma-addr

Mandatory. The IPv4 address of the destination in the NBMA network.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            map protocol-addr/prefix {
                nbma-address nbma-addr
            }
        }
    }
}
```

Usage Guidelines

Use this command to statically map a protocol address or protocol address prefix of a destination to an NBMA address in an NBMA network.

Use the **set** form of this command to map a destination protocol address or protocol address prefix to an NBMA address in an NBMA network.

Use the **delete** form of this command to remove a protocol address-to-NBMA address mapping.

Use the **show** form of this command to display a protocol address-to-NBMA address mapping.

interfaces tunnel <tunx> nhrp map <protocol-addr>/<prefix> register

Specifies that an NHRP Registration Request should be sent to the peer when the NHRP process starts.

Syntax

set interfaces tunnel *tunx* **nhrp map** *protocol-addr/prefix* **register**

delete interfaces tunnel *tunx* **nhrp map** *protocol-addr/prefix* **register**

show interfaces tunnel *tunx* **nhrp map** *protocol-addr/prefix*

Command Default

Automatic NHRP registration is disabled.

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of a destination. The protocol address is an IPv4 address.

prefix

Mandatory. The protocol address prefix of the destination. The protocol address prefix is an IPv4 prefix.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            map protocol-addr/prefix {
                register
            }
        }
    }
}
```

Usage Guidelines

Use this command to enable automatic sending of an NHRP Registration request when the NHRP process starts.

Use the **set** form of this command to enable automatic sending of an NHRP Registration request when the NHRP process starts up.

Use the **delete** form of this command to reset automatic NHRP registration to its default configuration.

Use the **show** form of this command to display NHRP address-mapping configuration.

interfaces tunnel <tunx> nhrp map <protocol-addr>/<prefix> register-no-unique

Enables NHRP Registration requests and replies to be sent without the unique flag being set.

Syntax

set interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix* **register-no-unique**

delete interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix* **register-no-unique**

show interfaces tunnel *tunx* **nhrp map** *protocol-addrlprefix*

Command Default

The unique flag is set in NHRP Registration requests and replies.

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of a destination. The protocol address is an IPv4 address.

prefix

Mandatory. The protocol address prefix of the destination. The protocol address prefix is an IPv4 prefix.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            map protocol-addr/prefix {
                register-no-unique
            }
        }
    }
}
```

Usage Guidelines

Use this command to enable NHRP Registration requests and replies to be sent without the unique flag being set. When the unique flag is set in an NHRP Registration request, a next-hop server rejects the request if the IP address to NBMA address mapping has changed and the request is received before the entry in the next-hop server times out. This rejection is typically the case when the client receives a new IP address, for example, in a DHCP environment.

Use the **set** form of this command to enable NHRP Registration requests and replies to be sent without the unique flag being set.

Use the **delete** form of this command to enable the default setting of the unique flag in NHRP Registration requests and replies.

Use the **show** form of this command to display NHRP address-mapping configuration.

interfaces tunnel <tunx> nhrp multicast parameters

Specifies how NHRP should soft switch multicast traffic.

Syntax

set interfaces tunnel *tunx* **nhrp multicast parameters** {**dynamic** | **nhs**}

delete interfaces tunnel *tunx* **nhrp multicast parameters**

show interfaces tunnel *tunx* **nhrp multicast parameters**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

dynamic

Forwards all multicast packets to all directly connected peers.

nhs

Repeats all multicast packets to each statically configured next hop.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            multicast {
                parameters {dynamic | nhs}
            }
        }
    }
}
```

Usage Guidelines

Use this command to specify how NHRP should soft switch multicast traffic. Only one of the two options, **dynamic** or **nhs**, can be specified.

NOTE
Take care to avoid multicast repetition when multiple next-hop servers are present.

Use the **set** form of this command to specify how NHRP should soft switch multicast traffic.

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

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

interfaces tunnel <tunx> nhrp multicast protocol-address <protocol-addr>

Instructs NHRP to forward multicast traffic to a specific protocol address.

Syntax

set interfaces tunnel *tunx* **nhrp multicast protocol-address** *protocol-addr*

delete interfaces tunnel *tunx* **nhrp multicast protocol-address** *protocol-addr*

show interfaces tunnel *tunx* **nhrp multicast protocol-address**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of a destination. The protocol address is an IPv4 address.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            multicast {
                protocol-address protocol-addr
            }
        }
    }
}
```

Usage Guidelines

Use this command to instruct NHRP to forward multicast traffic to the specified protocol address.

NOTE
Take care to avoid multicast repetition when multiple next-hop servers are present.

Use the **set** form of this command to instruct NHRP to forward multicast traffic to a specific protocol address.

Use the **delete** form of this command to remove the multicast protocol-address configuration.

Use the **show** form of this command to display the multicast protocol-address configuration.

interfaces tunnel <tunx> nhrp redirect

Instructs the sender of a forwarding packet to create a direct connection with the destination.

Syntax

set interfaces tunnel *tunx* **nhrp redirect**

delete interfaces tunnel *tunx* **nhrp redirect**

show interfaces tunnel *tunx* **nhrp redirect**

Command Default

Forwarding packets are forwarded normally.

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            redirect
        }
    }
}
```

Usage Guidelines

Use this command to instruct the sender of a forwarding packet to create a direct connection with the destination. This connection is achieved by sending NHRP Traffic Indication packets back to the sender of the forwarding packet.

Use the **set** form of this command to instruct the sender of a forwarding packet to create a direct connection with the destination.

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

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

interfaces tunnel <tunx> nhrp shortcut

Enables the creation of a shortcut route.

Syntax

set interfaces tunnel *tunx* **nhrp shortcut**

delete interfaces tunnel *tunx* **nhrp shortcut**

show interfaces tunnel *tunx* **nhrp shortcut**

Command Default

Shortcut routes are not created.

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            shortcut
        }
    }
}
```

Usage Guidelines

Use this command to enable the creation of a shortcut route; the resolution and establishment of a shortcut route is triggered when an NHRP Traffic Indication packet is received.

NOTE
You still need to use routing protocol or have static routes to the hub node in your NBMA network. NHRP does not advertise routes; creates a shortcut route only for an already routable subnet.

Use the **set** form of this command to enable the creation of a shortcut route.

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

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

interfaces tunnel <tunx> nhrp shortcut-destination

Instructs NHRP to create a shortcut route to a subnet located on the interface.

Syntax

set interfaces tunnel *tunx* **nhrp shortcut-destination**

delete interfaces tunnel *tunx* **nhrp shortcut-destination**

show interfaces tunnel *tunx* **nhrp shortcut-destination**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            shortcut-destination
        }
    }
}
```

Usage Guidelines

Use this command to instruct NHRP to create a shortcut route to a subnet located on the interface.

Use the **set** form of this command to instruct NHRP to create a shortcut route to a subnet located on the interface.

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

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

interfaces tunnel <tunx> nhrp shortcut-target <protocol-addr>/<prefix>

Defines an off-NBMA network prefix for which the Generic Routing Encapsulation (GRE) interface acts as a gateway.

Syntax

set interfaces tunnel *tunx* **nhrp shortcut-target** *protocol-addrlprefix*

delete interfaces tunnel *tunx* **nhrp shortcut-target** *protocol-addrlprefix*

show interfaces tunnel *tunx* **nhrp shortcut-target**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

protocol-addr

Mandatory. The protocol address of a destination. The protocol address is an IPv4 address.

prefix

Mandatory. The protocol address prefix of the destination. The protocol address prefix is an IPv4 prefix.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            shortcut-target protocol-addr/prefix
        }
    }
}
```

Usage Guidelines

Use this command to define an off-NBMA network prefix for which the GRE interface acts as a gateway.

Use the **set** form of this command to define an off-NBMA network prefix for which the GRE interface acts as a gateway.

Use the **delete** form of this command to remove the shortcut-target configuration.

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

interfaces tunnel <tunx> nhrp shortcut-target holding-time <time>

Specifies the hold time for Resolution Requests and Resolution Responses.

Syntax

set interfaces tunnel *tunx* **nhrp shortcut-target holding-time** *time*

delete interfaces tunnel *tunx* **nhrp shortcut-target holding-time**

show interfaces tunnel *tunx* **nhrp shortcut-target holding-time**

Parameters

tunx

Mandatory. The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

time

Mandatory. The hold time in seconds.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    tunnel tunx {
        nhrp {
            shortcut-target holding-time time
        }
    }
}
```

Usage Guidelines

Use this command to specify the hold time for Resolution Requests and Resolution Responses.

Use the **set** form of this command to specify the hold time for Resolution Requests and Resolution Responses.

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

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

reset ip nhrp flush tunnel

Removes all non-permanent entries.

Syntax

reset ip nhrp flush tunnel [*tunx*]

Command Default

Non-permanent entries for all tunnel interfaces are removed.

Parameters

tunx

The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes

Operational mode

Usage Guidelines

Use this command to remove all non-permanent entries for the specified tunnel interface. If no tunnel interface is specified, all non-permanent entries for all tunnel interfaces are removed.

Examples

The following example shows how to remove all non-permanent entries for the tun0 tunnel.

```
vyatta@vyatta:~$ reset ip nhrp flush tunnel tun0
vyatta@vyatta:~$
```

reset ip nhrp purge tunnel

Removes and reregisters all NHRP entries.

Syntax **reset ip nhrp purge tunnel** [*tunx*]

Command Default NHRP entries for all tunnel interfaces are removed.

Parameters *tunx*

The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes Operational mode

Usage Guidelines Use this command to remove and reregister all NHRP entries for a specific tunnel interface. If no tunnel interface is specified, all NHRP entries for all tunnel interfaces are removed and reregistered.

Examples The following example shows how to remove and reregister all NHRP entries for the tun0 tunnel.

```
vyatta@vyatta:~$ reset ip nhrp purge tunnel tun0
vyatta@vyatta:~$
```

show ip nhrp tunnel

Displays NHRP information about a specific tunnel.

Syntax **show ip nhrp tunnel** [*tunx*]

Command Default NHRP information for all tunnel interfaces is displayed.

Parameters *tunx*

The identifier for a tunnel interface. The identifier ranges from tun0 through tun*x*, where *x* is a non-negative integer.

Modes Operational mode

Usage Guidelines Use this command to display NHRP information about a specific tunnel. If no tunnel interface is specified, NHRP information about all tunnel interfaces is displayed.

Examples The following example shows how to display NHRP information about the tun0 tunnel.

```
vyatta@vyatta:~$ show ip nhrp tunnel tun0
Interface: tun0
Type: static
Protocol-Address: 10.0.0.100/24
NBMA-Address: 172.18.0.1
vyatta@vyatta:~$
```

sFlow

sFlow overview

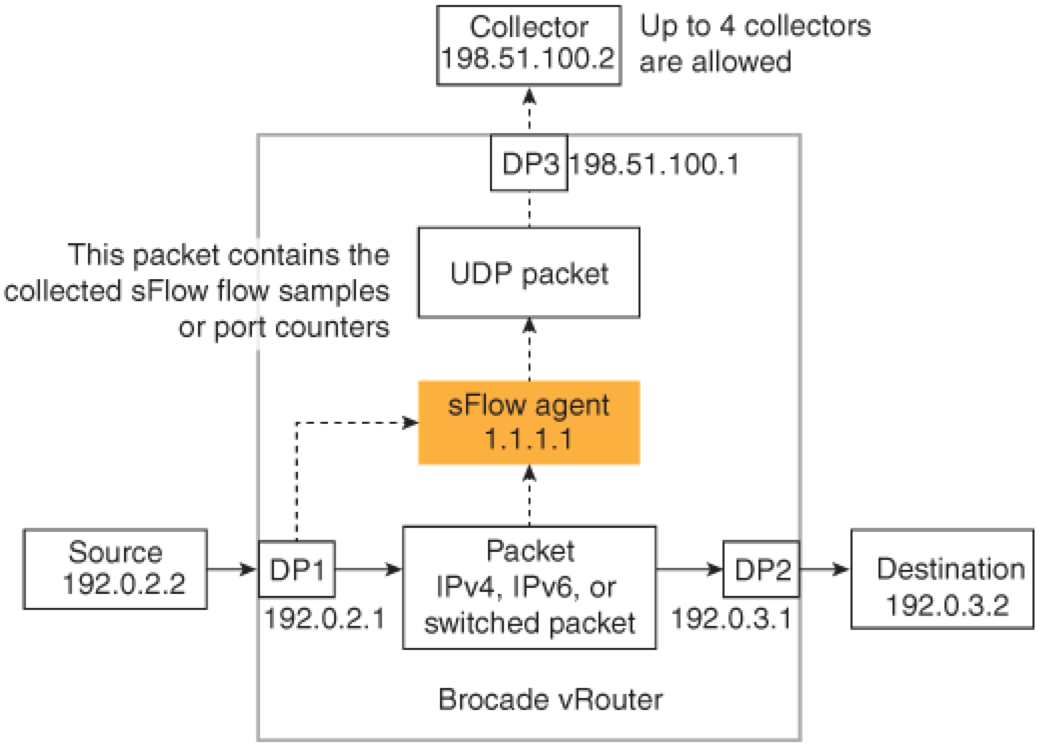
The Brocade vRouter supports sFlow (v5), a high-performance sFlow agent for monitoring traffic in a data network. sFlow, which stands for “sampled flow,” reports packet-flow and port-counter samples from a dataplane interface to receiving collectors on a timely basis.

NOTE
In a Brocade vRouter, sFlow supports enough functionality and flexibility such that packet-forwarding performance does not drop significantly, yet still delivers enough processing power to provide a significant advantage when processing packets through the dataplane pipeline.

You can configure a Brocade vRouter to perform the following tasks:

- Sample packet flows.
- Collect packet headers from sampled packets to gather inbound traffic information on these packets.
- Collect byte and packet port-counter statistics (counter samples).
- Compose flow sample messages from the collected information.
- Relay messages to an external device known as a collector. A Brocade vRouter supports up to four collectors.

FIGURE 13 sFlow configuration example



NOTE
The Brocade vRouter supports sFlow (v5), which replaces the version outlined in RFC 3176.

Physical interfaces

sFlow is physical-interface-based, which means that you can enable and disable sFlow on only a physical interface. However, all other interface types (for example, VIF and tunnel) are indirectly enabled because all are associated with physical interfaces.

Packet-flow and port-counter sampling

The Brocade vRouter acts as an sFlow agent, which collects inbound packet samples from a data-plane interface at the configured sampling rate. The Brocade vRouter also collects sample statistics from ports at the configured polling interval. The Brocade vRouter sends the collected information to the specified collectors.

Packet-flow sampling

The Brocade vRouter, acting as an sFlow agent, samples dropped and nondropped packets. Taking a sample involves extracting information from the packet, including trajectory information (for example, source and destination interface and next hop). For dropped packets, the Brocade vRouter adds the reason for dropping the packet to the extracted information.

Counter-flow sampling

By default, counter-flow sampling is enabled on the Brocade vRouter. The Brocade vRouter, acting as an sFlow agent, sends port statistics with counter sample data in the datagram stream that results from packet-flow sampling.

Communication with collectors

The Brocade vRouter creates one or more samples for each UDP packet. When enough samples are collected to fill a UDP datagram (up to 1,400 bytes), the Brocade vRouter sends the UDP datagram to the collectors.

Supported sFlow types

The Brocade vRouter supports the following sFlow data types:

TABLE 24 sFlow data types

Type	Description
Sampled header	Includes the protocol type, length, and packet header bytes. Packet header information for each packet must be reported using the sampled header format.
Sampled IPv4	Includes the source and destination IPv4 addresses, IPv4 packet length, protocol type, and so on.
Sampled IPv6	Includes the source and destination IPv6 addresses, IPv6 packet length, protocol type, and so on.
Sampled NAT data	Includes the source and destination IP addresses that are translated by NAT. If an address was not translated, it is equal to the address reported for the sampled header.packet length, and protocol type.
Counter data type	The interface port counters. Only the generic counter is supported.

Extended data types provide additional information about the sampled packet.

TABLE 25 Extended data types

Type	Description
Extended router	Includes the nexthop IP address and the source and destination prefixes.

Configuring sFlow

Table 26 shows how to configure sFlow for the sample Brocade vRouter configuration. Figure 13 on page 216 shows this configuration.

TABLE 26 Configuring sFlow

Step	Command
Specify the address of the agent. By default, this collector listens for sFlow data on the 6343 UDP port. **NOTE** A Brocade vRouter supports up to four collectors.	`vyatta@sflow1# set service sflow agent-address 1.1.1.1`
Set the polling interval to three seconds. Every three seconds, the sFlow agent that is running on the Brocade vRouter collects port counter data.	`vyatta@sflow1# set service sflow polling-interval 3`
Configure the sFlow agent to send the UDP datagrams that contain the collected sFlow information to the default port (6343) of the sFlow collector server at 198.51.100.2.	`vyatta@sflow1# set service sflow server-address 198.51.100.2 server-port 6343`
Specify 512 as the number of packets from which a sample is taken by the sFlow agent. In other words, for every 512 packets that flow through the interface, the sFlow agent selects one packet for analysis.	`vyatta@sflow1# set service sflow sampling-rate 512`
Commit the configuration.	`vyatta@R1# commit`
Save the configuration.	`vyatta@sflow1# save`

Display the sFlow configuration.
The output shows that sFlow was configured, but no statistics were collected. This is because sFlow has not yet been enabled on an interface.
	`vyatta@sflow1# run show sflow`
	`{`
	`    "sFlow information": {`
	`        "sFlow version": 5,`
	`        "sFlow services": "enabled",`
	`        "sFlow agent IP address": "1.1.1.1",`
	`        "Collector destinations configured": 1,`
	`        "Collectors": [{`
	`                "IP address": "198.51.100.2",`
	`                "UDP port number": 6343`
	`            }`
	`        ],`
	`        "Polling interval": 3,`
	`        "Configured default sampling rate": 512,`
	`        "Actual default sampling rate": 512,`
	`        "sFlow max-header size": 128,`
	`        "UDP packets sent": 0,`
	`        "Flow samples collected": 0,`
	`        "sFlow interfaces": [],`
	`        "Total sFlow interfaces": 0`
	`    }`
	`}`

Enable sFlow on the dp0p192p1 interface (Figure 13 on page 216 indicates it as DP1).	`vyatta@sflow1# set interfaces dataplane dp0p192p1 sflow`
Commit the configuration.	`vyatta@R1# commit`
Save the configuration. The sFlow agent can now start collecting packet samples and port-counter statistics.	`vyatta@sflow1# save`

TABLE 26 Configuring sFlow (Continued)

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>

Display the sFlow configuration.

The output shows that sFlow is enabled for the dp0p192p1 interface and shows that one sample packet was collected.

</td><td>

```
vyatta@sflow1# run show sflow
{
    "sFlow information": {
        "sFlow version": 5,
        "sFlow services": "enabled",
        "sFlow agent IP address": "1.1.1.1",
        "Collector destinations configured": 1,
        "Collectors": [{
                "IP address": "198.51.100.2",
                "UDP port number": 6343
            }
        ],
        "Polling interval": 3,
        "Configured default sampling rate": 512,
        "Actual default sampling rate": 512,
        "sFlow max-header size": 128,
        "UDP packets sent": 0,
        "Flow samples collected": 0,
        "sFlow interfaces": [{
                "name": "dp0p192p1"
            }
        ],
        "Total sFlow interfaces": 1
    }
}
```

</td></tr>
</table>

sFlow Commands

interfaces dataplane <dp-port> sflow

Specifies an interface for which to record inbound sFlow packet statistics and port counters.

Syntax **set interfaces dataplane** *dp-port* **sflow**

delete interfaces dataplane *dp-port* **sflow**

Parameters *dp-port*

The name of a dataplane interface.

Modes Configuration mode

Configuration Statement

```
interfaces {
        dataplane dp-port {
                sflow
        }
}
```

Usage Guidelines You can enable multiple interfaces by issuing this command multiple times, once for each interface.

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

Use the **delete** form of this command to disable sFlow on an interface.

service sflow agent-address <IPv4-or-IPv6>

Specifies the IPv4 or IPv6 address (of an sFlow agent) to be included in the sFlow packets that are sent to the collectors.

Syntax

set service sflow agent-address *IPv4-or-IPv6*

delete service sflow agent-address *IPv4-or-IPv6*

Command Default

By default, the Brocade vRouter provides an IP address that identifies the agent.

Parameters

IPv4-or-IPv6

The IPv4 or IPv6 address of an sFlow agent.

NOTE
The address does not have to be active.

Modes

Configuration mode

Configuration Statement

```
service {
    sflow {
        agent-address address
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the IP address of an sFlow agent.

Use the **delete** form of this command to delete the IP address of an sFlow agent.

service sflow server-address <IPv4-or-IPv6> server-port <port>

Specifies the IPv4 or IPv6 IP address of an sFlow collector and a port to which to send the UDP datagrams that contain the collected sFlow information.

Syntax

set service sflow server-address *IPv4-or-IPv6* **server-port** *port*

delete service sflow server-address *IPv4-or-IPv6*

Command Default

The default port is 6343.

Parameters

IPv4-or-IPv6

The IP address of the sFlow collector server.

port

A port of an sFlow collector server to which to send the UDP datagrams that contain the collected sFlow information.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    sflow {
        server-address IPv4-or-IPv6 {
            server-port port
        }
    }
}
```

Usage Guidelines

You can specify multiple sFlow collectors by entering this command multiple times. The maximum number of sFlow collectors allowed is four.

Use the **set** form of this command to specify an sFlow collector and a port to which to send the collected sFlow data.

Use the **delete** form of this command to remove an sFlow collector from the list of collectors to which to send sFlow data.

service sflow sampling-rate <sampling-rate>

Specifies the rate at which packets are sampled.

Syntax

set service sflow sampling-rate *sampling-rate*

delete service sflow sampling-rate *sampling-rate*

Command Default

The default sampling rate is 2048 packets.

Parameters

sampling-rate

The number of packets from which to pick one packet for analysis. The number ranges from 512 through 65535. For example, if a number of 512, it means that the sampling rate is 1 in 512. In other words, the Brocade vRouter selects a packet every other 512 packets.

Modes

Configuration mode

Configuration Statement

```
service {
    sflow {
        sampling-rate sampling-rate
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the sampling rate.

Use the **delete** form of this command to restore the default sampling rate, which is 2,048 packets.

service sflow polling-interval <polling-rate>

Specifies how often sFlow port-counter statistics are collected.

Syntax

set service sflow polling-interval *polling-rate*

delete service sflow polling-interval *polling-rate*

Command Default

The default polling rate is 20 seconds.

Parameters

polling-rate

A polling rate in seconds. The rate ranges from 0 through 65535. A rate of 0 means that counter polling is disabled.

Modes

Configuration mode

Configuration Statement

```
service {
    sflow {
        polling-interval polling-rate
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify a polling rate.

Use the **delete** form of this command to restore the default polling rate, which is 20 seconds.

show sflow

Displays sFlow configuration information and sFlow usage statistics.

Syntax **show sflow**

Modes Operational mode

Examples The following example shows how to display sFlow configuration information and usage statistics.

```
vyatta@sflow1# run sh sflow
{
    "sFlow information": {
        "sFlow version": 5,
        "sFlow services": "enabled",
        "sFlow agent IP address": "1.1.1.1",
        "Collector destinations configured": 1,
        "Collectors": [{
                "IP address": "50.1.1.2",
                "UDP port number": 6343
            }
        ],
        "Polling interval": 3,
        "Configured default sampling rate": 512,
        "Actual default sampling rate": 512,
        "sFlow max-header size": 128,
        "UDP packets sent": 0,
        "Flow samples collected": 0,
        "sFlow interfaces": [],
        "Total sFlow interfaces": 0
    }
    }
```

clear sflow

Clears the collected sFlow statistics.

Syntax **clear sflow**

Modes Operational mode

Examples The following example shows how to clear the collected sFlow statistics (the number of UDP packets sent and sFlow flow samples collected).

```
vyatta@vyatta# run clear sflow
[edit]
vyatta@vyatta# run show sflow
{
    "sFlow information": {
        "sFlow version": 5,
        "sFlow services": "enabled",
        "sFlow agent IP address": "1.1.1.1",
        "Collector destinations configured": 1,
        "Collectors": [{
                "IP address": "198.51.100.2",
                "UDP port number": 6343
            }
        ],
        "Polling interval": 3,
        "Configured default sampling rate": 512,
        "Actual default sampling rate": 512,
        "sFlow max-header size": 128,
        "UDP packets sent": 0,
        "Flow samples collected": 0,
        "sFlow interfaces": [{
                "name": "dp0s160"
            }
        ],
        "Total sFlow interfaces": 1
    }
}
```

TWAMP

TWAMP overview

A Two-Way Active Measurement Protocol (TWAMP) server on a Brocade vRouter measures round-trip IP performance between any two devices in a network that supports the standard. The TWAMP server implementation is based on the specifications outlined in RFC 5357.

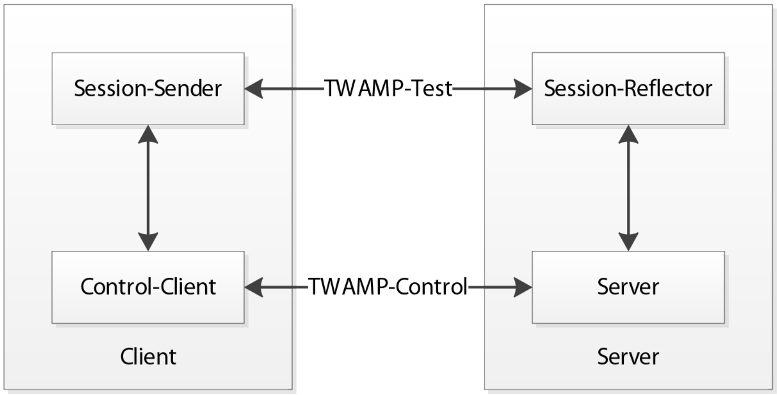
The architecture of the Brocade vRouter TWAMP server solution defines the following logical components, as shown in the following figure.

- Session-Reflector—Creates and sends measurement packets when it receives a TWAMP-test packet.
- Server—Manages multiple TWAMP sessions.

The following components are part of the TWAMP client:

- Session-Sender—Creates and sends TWAMP-test packets to Session-Reflector.
- Control-Client—Sends requests to the TWAMP server to measure IP performance.

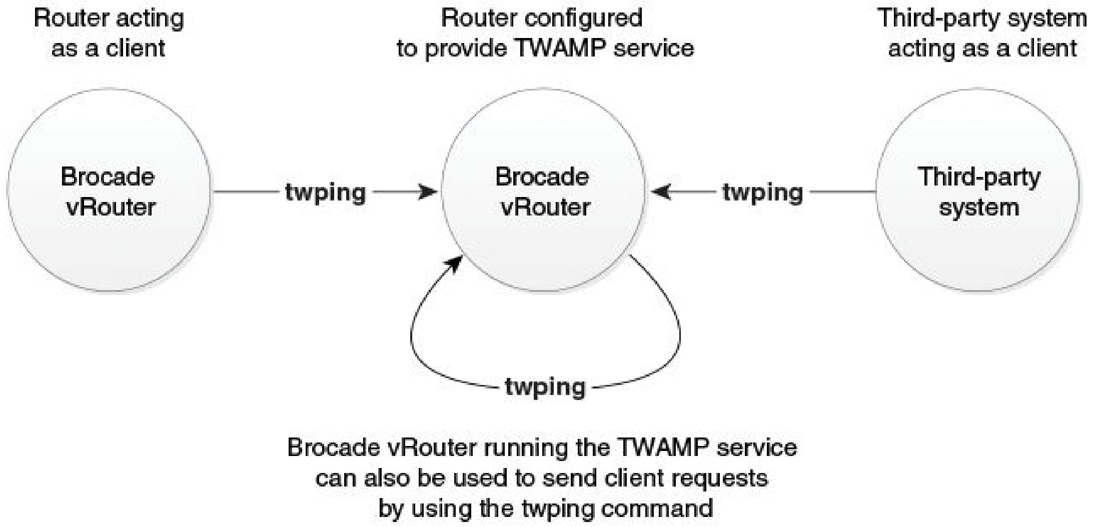
FIGURE 14 TWAMP architecture



A Brocade vRouter includes the **twping** command in the /opt/vyatta/bin/twping directory, which you can use to send client requests to a TWAMP sever.

A Brocade vRouter can be a TWAMP server and client at the same time. However, the client can be another Brocade vRouter or a third-party system, as shown in the following figure.

FIGURE 15 TWAMP server-client interaction


Router acting as a client
Router configured to provide TWAMP service
Third-party system acting as a client
Brocade vRouter
twping
Brocade vRouter
twping
Third-party system
twping
Brocade vRouter running the TWAMP service can also be used to send client requests by using the twping command

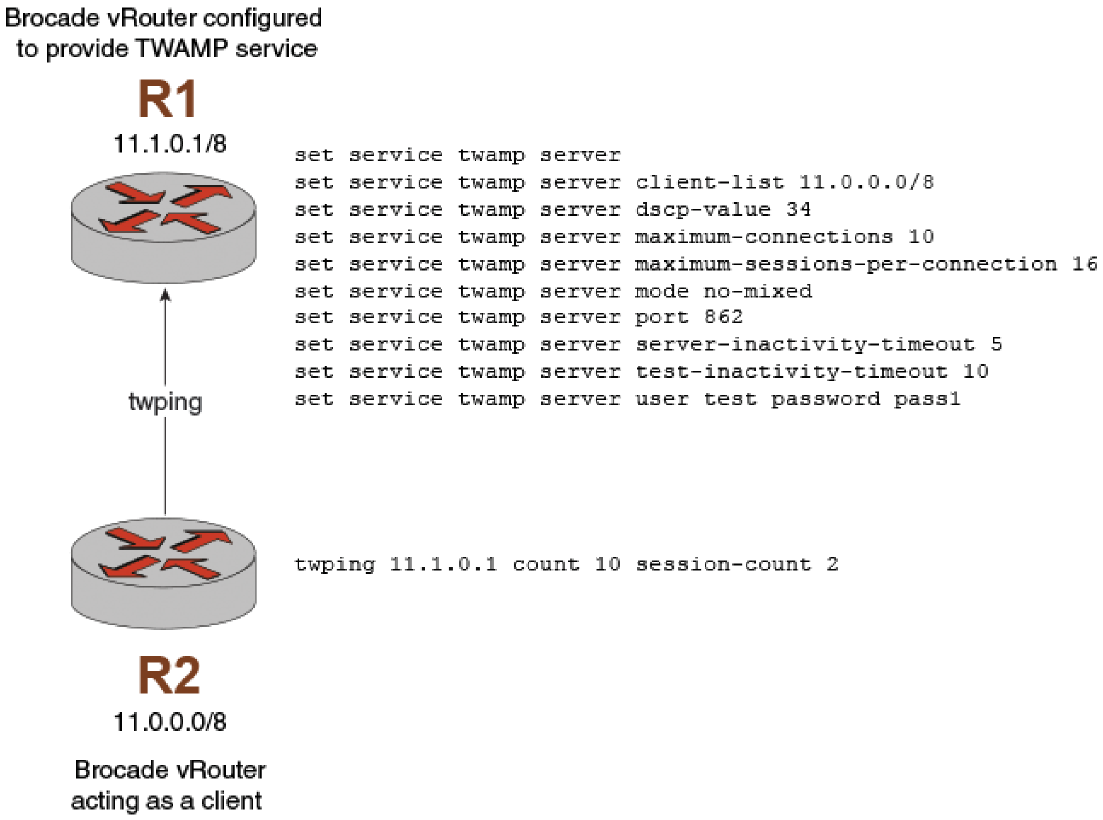
TWAMP configuration

This section includes the following examples:

- Configuring the TWAMP server on page 231
- Using twping to measure IP performance on page 232

Figure 16 shows the TWAMP client/server topology used in the following examples.

FIGURE 16 TWAMP server configuration example


Brocade vRouter configured to provide TWAMP service
R1
11.1.0.1/8
twping
R2
11.0.0.0/8
Brocade vRouter acting as a client

```
set service twamp server
set service twamp server client-list 11.0.0.0/8
set service twamp server dscp-value 34
set service twamp server maximum-connections 10
set service twamp server maximum-sessions-per-connection 16
set service twamp server mode no-mixed
set service twamp server port 862
set service twamp server server-inactivity-timeout 5
set service twamp server test-inactivity-timeout 10
set service twamp server user test password pass1
```

```
twping 11.1.0.1 count 10 session-count 2
```

Configuring the TWAMP server

To use the TWAMP server on a Brocade vRouter, you must first configure the TWAMP service.

To configure the TWAMP service, perform the following steps in configuration mode.

TABLE 27 Configuring the TWAMP service

Step	Command
Start the TWAMP server. By default, the server accepts any connection request from any client.	`vyatta@R1# set service twamp server`
Add the client with the 11.0.0.0/8 IP address to the list of clients that can connect to the TWAMP server in the unauthenticated mode.	`vyatta@R1# set service twamp server client-list 11.0.0.0/8`
Specify 34 as the base-10 value of the DSCP byte in the IP header of control packets sent from the server.	`vyatta@R1# set service twamp server dscp-value 34`
Specify 10 as the maximum number of control sessions for each TWAMP server.	`vyatta@R1# set service twamp server maximum-connections 10`
Limit the number of maximum number of test sessions for each control session to 16.	`vyatta@R1# set service twamp server maximum-sessions-per-connection 16`

TABLE 27 Configuring the TWAMP service (Continued)

Step	Command
Disable the mixed client authentication mode.	`vyatta@R1# set service twamp server mode no-mixed`
Specify 862 as the TCP port used for control sessions.	`vyatta@R1# set service twamp server port 862`
Set the timeout value for control-session inactivity to 5 seconds.	`vyatta@R1# set service twamp server server-inactivity-timeout 5`
Set the timeout value for test-session inactivity to 10 seconds.	`vyatta@R1# set service twamp server test-inactivity-timeout 10`
Create the test user account and assign a password to it.	`vyatta@R1# set service twamp server user test password pass1`
Commit the configuration.	`vyatta@R1# commit`
Exit the configuration mode.	`vyatta@R1# exit`

Show the status of the server.	`vyatta@R1:~$ show service twamp server status`
	`TWAMP Server is configured and is running.`

Show TWAMP server session activity.

```
vyatta@R1:~$ show service twamp server sessions all
Total number of sessions: 16
Total number of active sessions: 16
--> Control Session initiated by [Rtr2-tap0]:39549 in Authenticated mode
SID                              SENDER               REFLECTOR            STATUS  DSCP
0a00020fd863fa0eeecc1444b5000a61 [Rtr2-tap0]:8833    [Rtr1]:56967         ACTIVE   0x0
0a00020fd863fa0eee08638005774d72 [Rtr2-tap0]:8879    [Rtr1]:38676         ACTIVE   0x0
0a00020fd863fa0eecd9f0a68f69a638 [Rtr2-tap0]:8935    [Rtr1]:46080         ACTIVE   0x0
0a00020fd863fa0eeb2dba5175914887 [Rtr2-tap0]:8859    [Rtr1]:60355         ACTIVE   0x0
0a00020fd863fa0ee5a9b06cff9bacf1 [Rtr2-tap0]:8769    [Rtr1]:56089         ACTIVE   0x0
0a00020fd863fa0ee4b11c71bcba56af [Rtr2-tap0]:8778    [Rtr1]:48298         ACTIVE   0x0
0a00020fd863fa0ee3f7c679e973a4f4 [Rtr2-tap0]:8917    [Rtr1]:36039         ACTIVE   0x0
0a00020fd863fa0ee31e6473f8fc1a74 [Rtr2-tap0]:8845    [Rtr1]:45534         ACTIVE   0x0
<--
--> Control Session initiated by [Rtr2-tap0]:39550 in Authenticated mode
SID                              SENDER               REFLECTOR            STATUS  DSCP
0a00020fd863fa1a1f73b8df73463109 [Rtr2-tap0]:8958    [Rtr1]:54254         ACTIVE   0x0
0a00020fd863fa1a1d4dbe795b7d70d7 [Rtr2-tap0]:8949    [Rtr1]:58983         ACTIVE   0x0
0a00020fd863fa1a1bd7fe03660ed58a [Rtr2-tap0]:8817    [Rtr1]:49623         ACTIVE   0x0
0a00020fd863fa1a1a223635b8669238 [Rtr2-tap0]:8871    [Rtr1]:55287         ACTIVE   0x0
0a00020fd863fa1a18d1f250ed2c2c23 [Rtr2-tap0]:8863    [Rtr1]:57902         ACTIVE   0x0
0a00020fd863fa1a17756d14abae00a0 [Rtr2-tap0]:8831    [Rtr1]:37197         ACTIVE   0x0
0a00020fd863fa1a1696dda7c4c0d57b [Rtr2-tap0]:8800    [Rtr1]:54229         ACTIVE   0x0
0a00020fd863fa1a1594751787e08a1f [Rtr2-tap0]:8779    [Rtr1]:53110         ACTIVE   0x0
<--
```

Using twping to measure IP performance

To measure round-trip IP Performance, perform the following steps in operational mode.

TABLE 28 Measuring round-trip IP performance by using twping

Step	Command

Use the **twping** command to send a request to the TWAMP server to measure the round-trip IP performance.	`vyatta@R2:~$ twping 11.1.0.1 count 10 session-count 2`
	`Approximately 4.2 seconds until results available`
	
	`--- twping statistics from [11.1.0.1]:8889 to [Rtr2]:55026 ---`
	`SID:    0b010001d961f8d25daa301413dddec8`
	`first:  2015-07-28T12:49:23.476`
	`last:   2015-07-28T12:49:24.605`
	`10 sent, 0 lost (0.000%), 0 send duplicates, 0 reflect duplicates`
	`round-trip time min/median/max = 0.746/1.5/2.39 ms, (err=1.11 ms)`
	`send time min/median/max = 0.196/0.6/1.54 ms, (err=0.556 ms)`
	`reflect time min/median/max = 0.543/0.9/1.79 ms, (err=0.556 ms)`
	`reflector processing time min/max = 0.0129/0.0277 ms`
	`two-way PDV = 0.9 ms (P95-P50)`
	`send PDV = 1 ms (P95-P50)`
	`reflect PDV = 0.9 ms (P95-P50)`
	`send hops = 0 (consistently)`
	`reflect hops = 0 (consistently)`
	
	`--- twping statistics from [11.1.0.1]:8904 to [Rtr2]:42203 ---`
	`SID:    0b010001d961f8d25eff1b28f68b20f7`
	`first:  2015-07-28T12:49:23.495`
	`last:   2015-07-28T12:49:24.940`
	`10 sent, 0 lost (0.000%), 0 send duplicates, 0 reflect duplicates`
	`round-trip time min/median/max = 1.24/1.6/3.24 ms, (err=1.11 ms)`
	`send time min/median/max = 0.37/0.7/2.52 ms, (err=0.556 ms)`
	`reflect time min/median/max = 0.713/0.9/1.51 ms, (err=0.556 ms)`
	`reflector processing time min/max = 0.0157/0.031 ms`
	`two-way PDV = 1.7 ms (P95-P50)`
	`send PDV = 1.9 ms (P95-P50)`
	`reflect PDV = 0.7 ms (P95-P50)`
	`send hops = 0 (consistently)`
	`reflect hops = 0 (consistently)`

TWAMP Commands

service twamp server

Starts the TWAMP server.

Syntax

set service twamp server

delete service twamp server

show service twamp server

Modes

Configuration mode

Configuration Statement

```
service {
   twamp {
       server
   }
}
```

Usage Guidelines

If other server parameters are not configured, this command configures the TWAMP server with the default values.

Use the **set** form of this command to start the TWAMP server.

Use the **delete** form of this command to stop the TWAMP server.

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

service twamp server client-list <ip-address>

Adds a client in a specified network to the list of clients that can connect to the TWAMP server in the unauthenticated mode.

Syntax

set service twamp server client-list *ip-address*

delete service twamp server client-list *ip-address*

Parameters

ip-address

IPv4 or IPv6 IP address of the system to add to the list of clients that can connect to the server. To specify multiple systems, use a subnet range.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            client-list ip-address
        }
    }
}
```

Usage Guidelines

By default, any client can connect to the server. However, after you configure a client list, only those clients in the list can connect in the unauthenticated mode. Clients that are not in this list can connect to the server in the authenticated, encrypted, or mixed mode.

Use the **set** form of this command to add a client to the list of clients that can connect to the server in the unauthenticated mode. You can use this form multiple times to add multiple clients to the list.

Use the **delete** form of this command to delete a client from the list.

service twamp server dscp-value <value>

Specifies the base-10 value of the DSCP byte in the IP header of control packets sent from the server.

Syntax **set service twamp server dscp-value** *value*

delete service twamp server dscp-value *value*

Command Default 0.

Parameters *value*

Base-10 value of the DSCP byte.

Modes Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            dscp-value value
        }
    }
}
```

Usage Guidelines Use the **set** form of this command to set the DSCP value.

Use the **delete** form of this command to set the DSCP value to the default value.

service twamp server maximum-connections <count>

Specifies the maximum number of control sessions for each TWAMP server.

Syntax

set service twamp server maximum-connections *count*

delete service twamp server maximum-connections *count*

Command Default

16.

Parameters

count

Maximum number of control sessions for each TWAMP server. The count ranges from 1 through 64.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            maximum-connections count
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the maximum number of control sessions for each TWAMP server.

Use the **delete** form of this command to reset the maximum number of control sessions to the default count, which is 16.

service twamp server maximum-sessions-per-connection <count>

Specifies the maximum number of test sessions for each control session.

Syntax

set service twamp server maximum-sessions-per-connection *count*

delete service twamp server maximum-sessions-per-connection *count*

Command Default

8.

Parameters

count

Maximum number of test sessions for each control session. The count ranges from 1 through 64.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            maximum-sessions-per-connection count
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the maximum number of test sessions for each control session.

Use the **delete** form of this command to reset the maximum number of test sessions to the default count, which is 8.

service twamp server mode <authentication-mode>

Disables a client authentication mode.

Syntax

set service twamp server mode *authentication-mode*

delete service twamp server mode *authentication-mode*

Command Default

By default, the following modes are enabled: unauthenticated, authenticated, encrypted, and mixed.

Parameters

authentication-mode

One of the following TWAMP authentication modes:

- **no-authenticated** —Disables support for authenticated sessions.
- **no-encrypted** —Disables support for encrypted sessions.
- **no-mixed** —Disables support for mixed mode sessions.
- **no-unauthenticated** —Disables support for unauthenticated sessions.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            mode authentication-mode
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to disable a client-authentication mode.

Use the **delete** form of this command to enable a client-authentication mode.

service twamp server port <port-number>

Specifies the TCP port for a control session.

Syntax

set service twamp server port *port-number*

delete service twamp server port *port-number*

Command Default

862.

Parameters

port-number

A TCP port number. The port number ranges from 1 through 65535.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            port port-number
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the TCP port for a control session.

Use the **delete** form of this command to reset the TCP port number to the default port number, which is 862.

service twamp server server-inactivity-timeout <seconds>

Specifies the timeout value for control-session inactivity.

Syntax

set service twamp server server-inactivity-timeout *seconds*

delete service twamp server server-inactivity-timeout *seconds*

Command Default

900.

Parameters

seconds

Number of seconds before a control session times out due to inactivity. The number of seconds ranges from 1 through 3600.

NOTE
The inactivity timer starts only when no test sessions are active on the associated control session.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            server-inactivity-timeout seconds
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the control-session timeout value.

Use the **delete** form of this command to reset the timeout value to the default number of seconds, which is 900.

service twamp server test-inactivity-timeout <seconds>

Specifies the timeout value for test-session inactivity.

Syntax

set service twamp server test-inactivity-timeout *seconds*

delete service twamp server test-inactivity-timeout *seconds*

Command Default

900.

Parameters

seconds

Number of seconds before a test session times out due to inactivity. The number of seconds ranges from 1 through 3600.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            test-inactivity-timeout seconds
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the test-session timeout value.

Use the **delete** form of this command to reset the timeout value to the default number of seconds, which is 900.

service twamp server user <username> password <pwd>

Configures a username and password for use with authenticated, encrypted, and mixed mode sessions.

Syntax

set service twamp server user *username* **password** *pwd*

delete service twamp server user *username*

Parameters

username

Specifies the username to create.

pwd

Specifies the user account password.

Modes

Configuration mode

Configuration Statement

```
service {
    twamp {
        server {
            user username {
                password pwd
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to create a user account for accessing the TWAMP server.

NOTE
Pressing Return after entering the **set service twamp server user username password** command, allows you to enter the password without it being shown on the console.

Use the **delete** form of this command to delete a user account.

show service twamp server session [all | client <ip-address> | summary]

Shows TWAMP session information.

Syntax **show service twamp server session [all | client** *ip-address* | **summary**]

Command Default Displays the details of all current TWAMP sessions.

Parameters **all**

Displays the details of all current TWAMP sessions.

client *ip-address*

Displays information for a specific TWAMP session.

summary

Displays a summary of current TWAMP sessions.

Modes Operational mode

Usage Guidelines Use this command to display TWAMP session information.

Examples The following example shows how to display the details of all current TWAMP sessions.

```
vyatta@R1:~$ show service twamp server session all
Total number of sessions: 20
Total number of active sessions: 20
--> Control Session initiated by [11.2.0.2]:40972 in Open mode

SID                               SENDER           REFLECTOR          STATUS   DSCP
0b020001d86c04daae30446b81fd2db7  [11.2.0.2]:8772  [11.2.0.1]:50150   ACTIVE   0x22
0b020001d86c04daa6a044ae89bbc550  [11.2.0.2]:8955  [11.2.0.1]:35927   ACTIVE   0x22
0b020001d86c04da9ed83c6cef1afdcd  [11.2.0.2]:8864  [11.2.0.1]:53680   ACTIVE   0x22
0b020001d86c04da9720535c5eee195b  [11.2.0.2]:8927  [11.2.0.1]:47290   ACTIVE   0x22
0b020001d86c04da8fc0c1fc9ca93c01  [11.2.0.2]:8928  [11.2.0.1]:53642   ACTIVE   0x22
<--
--> Control Session initiated by [11:4::2]:36319 in Open mode
SID                               SENDER           REFLECTOR          STATUS   DSCP
00000001d86c04f64ad1f1cf230de104  [11:4::2]:8778   [11:4::1]:57929    ACTIVE   0x24
00000001d86c04f63dc12f0962f9e295  [11:4::2]:8767   [11:4::1]:59181    ACTIVE   0x24
00000001d86c04f6315b573eb64f175c  [11:4::2]:8788   [11:4::1]:45432    ACTIVE   0x24
00000001d86c04f6242f2f98b2210c93  [11:4::2]:8817   [11:4::1]:41301    ACTIVE   0x24
00000001d86c04f6154ff0023445572a  [11:4::2]:8908   [11:4::1]:60239    ACTIVE   0x24
<--
--> Control Session initiated by [11.1.0.2]:47329 in Open mode
SID                               SENDER           REFLECTOR         STATUS   DSCP
0b010001d86c04c3f56d3b6c3ac57bb6  [11.1.0.2]:8815  [11.1.0.1]:52183  ACTIVE   0x21
0b010001d86c04c3e38b47c75ee48e5b  [11.1.0.2]:8769  [11.1.0.1]:54913  ACTIVE   0x21
0b010001d86c04c3d20663c7f080668a  [11.1.0.2]:8847  [11.1.0.1]:33790  ACTIVE   0x21
0b010001d86c04c3c7b3636fc33e9784  [11.1.0.2]:8960  [11.1.0.1]:40148  ACTIVE   0x21
0b010001d86c04c3bde8b3b3bc4bd31f  [11.1.0.2]:8819  [11.1.0.1]:40670  ACTIVE   0x21
<--
--> Control Session initiated by [11:3::2]:33644 in Open mode
SID                               SENDER           REFLECTOR         STATUS   DSCP
00000001d86c04eb699b6f5c6e85f63f  [11:3::2]:8763   [11:3::1]:57336   ACTIVE   0x23
00000001d86c04eb5c679cc7b99e8083  [11:3::2]:8835   [11:3::1]:53602   ACTIVE   0x23
00000001d86c04eb50058149325653cb  [11:3::2]:8760   [11:3::1]:52560   ACTIVE   0x23
00000001d86c04eb1876d117e72ea6e0  [11:3::2]:8817   [11:3::1]:42781   ACTIVE   0x23
00000001d86c04eb0b238972b68ae13d  [11:3::2]:8935   [11:3::1]:55790   ACTIVE   0x23
<--
```

The following example shows how to display the information for the TWAMP session that is associated with the 11.1.0.2 IP address.

```
vyatta@R1:~$ show service twamp server session client 11.1.0.2
--> Control Session initiated by [11.2.0.2]:40972 in Open mode

SID                               SENDER           REFLECTOR         STATUS   DSCP
0b020001d86c04daae30446b81fd2db7  [11.2.0.2]:8772  [11.2.0.1]:50150  ACTIVE   0x22
0b020001d86c04daa6a044ae89bbc550  [11.2.0.2]:8955  [11.2.0.1]:35927  ACTIVE   0x22
0b020001d86c04da9ed83c6cef1afdcd  [11.2.0.2]:8864  [11.2.0.1]:53680  ACTIVE   0x22
0b020001d86c04da9720535c5eee195b  [11.2.0.2]:8927  [11.2.0.1]:47290  ACTIVE   0x22
0b020001d86c04da8fc0c1fc9ca93c01  [11.2.0.2]:8928  [11.2.0.1]:53642  ACTIVE   0x22
<--
```

The following example shows how to display a summary of current TWAMP sessions.

```
vyatta@R1:~$ show service twamp server session summary
Total connected clients:       4
Total active test sessions:    20
Total inactive test sessions:  0

Client 0: Initiated by [11.2.0.2]:40972 in Open mode
        Active sessions: 5
        Inactive sessions: 0

Client 1: Initiated by [11:4::2]:36319 in Open mode
        Active sessions: 5
        Inactive sessions: 0

Client 2: Initiated by [11.1.0.2]:47329 in Open mode
        Active sessions: 5
        Inactive sessions: 0

Client 3: Initiated by [11:3::2]:33644 in Open mode
        Active sessions: 5
        Inactive sessions: 0
```

twping <host-address>

Measures the round-trip IP performance using the TWAMP server.

Syntax **twping** *host-address* [**auth-mode** { **authenticated** | **encrypted** | **mixed** } **user** *user*] [**control-port** *port*] [**count** *count*] [**interval** *seconds*] [**padding** *size*] [**port-range** *port1* **-** *port2*][**sample** *seconds*][**session-count** *s-count*] [**test-dscp-value** *dscp-value*]

Parameters *host-address*

Host name or IP address (IPv4 or IPv6) of the TWAMP server.

auth-mode { **authenticated** | **encrypted** | **mixed** } **user** *user*

Authentication mode (authenticated, encrypted, or mixed).

control-port *port*

Port for server control connection.

count *count*

Number of test packets to send (default is 100).

interval *seconds*

Number of seconds between test packets.

padding *size*

Padding, in bytes, to add to test packets.

port-range *port1* **-** *port2*

UDP port range to use for test packets.

sample *seconds*

Interval, in seconds, to display statistics during the session.

session-count *s-count*

Number of test sessions to create and use.

test-dscp-value *dscp-value*

Base-10 DSCP value. The value ranges from 0 through 63; the default value is 0.

Modes Operational mode

Usage Guidelines Use this command to measure round-trip IP performance. If the authentication mode is authenticated, encrypted, or mixed, then this command prompts you to enter the required username and password before continuing.

Port Mirroring

Overview

Switch Port Analyzer (SPAN) and Remote SPAN (RSPAN) enable you to monitor and troubleshoot network traffic. SPAN mirrors traffic on one or more source interfaces on a Brocade vRouter to a destination interface on the same router.

RSPAN mirrors traffic to a destination interface on a remote Brocade vRouter. RSPAN mirroring has source and destination Brocade vRouters. On the source router, traffic from source interfaces is mirrored to a VLAN interface (VIF). This RSPAN VLAN is dedicated to transporting mirrored traffic to the remote router. RSPAN on the destination router receives mirrored traffic on this VLAN interface and forwards it to a destination interface on this router.

Source interfaces can be both physical and VLAN interfaces. Both ingress and egress traffic on source interfaces can be mirrored. For SPAN and an RSPAN destination router, the destination interface is a physical port to which a network monitoring tool is connected for capturing and analyzing the traffic.

The following figures show port-mirroring sessions for SPAN, RSPAN-source, and RSPAN-destination session types.

FIGURE 17 SPAN port mirroring session

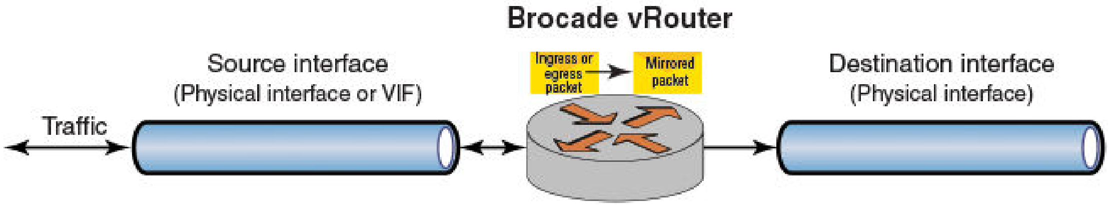
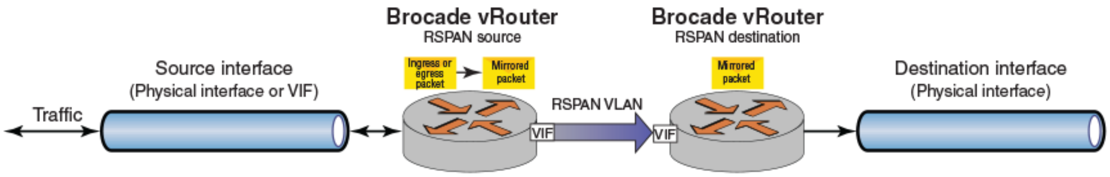


FIGURE 18 RSPAN port mirroring session



Configuration requirements and limits

The Brocade vRouter has the following configuration requirements and limits for SPAN and RSPAN.

- A total of eight sessions is supported.
- By default, a port-monitoring session is enabled. When a source interface is configured, packets from the interface are mirrored only when the session type and the destination interface for the session are also configured.
- When a session is configured for port monitoring, the type value must be specified. The value is mandatory.
- The SPAN type supports the following interfaces:
  - Source interface, physical or VLAN
  - Destination interface, physical only
- The RSPAN-source type supports the following interfaces:
  - Source interface, physical or VLAN
  - Destination interface, VLAN only
- The RSPAN-destination type supports the following interfaces:
  - Source interface, VLAN only
  - Destination interface, physical only
- A source interface and a destination interface cannot be same.
- A total of eight distinct source interfaces is allowed for all sessions.
- The default direction for monitoring on a source interface is both receive (RX) and transmit (TX). The direction can also be configured as RX only or TX only.
- Source interfaces cannot be shared between sessions.
- The source VIF for the RSPAN-destination type must be part of a bridge group.
- A destination interface that is configured for monitoring transmits mirrored packets only. Do not use this interface for other traffic.
- The destination interface should be configured in the system, should not be disabled, and cannot have address, IPv4, and IPv6 attributes.
- A physical destination interface cannot have QoS configured on it.
- Only one destination interface for each session is allowed, and cannot be shared between different sessions.
- The destination interface for any session cannot be part of a bridge group.
- An RSPAN-destination type can have only one source interface for a port-monitor session.
- For port-monitoring sessions, you are not allowed to configure a physical interface and VIF from the same physical interface together. For example, both dp0s7 and dp0s7.700 are not allowed.
- You cannot change the type of the session after you commit the configuration.

Configuring port mirroring

The Brocade vRouter allows you to configure the following types of port mirroring:

- SPAN port mirroring
- RSPAN-source port mirroring
- RSPAN-destination port mirroring

SPAN port mirroring

The following example shows a sample Brocade vRouter configuration for SPAN port mirroring.

TABLE 29 Configuring SPAN port mirroring

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Configure a SPAN session.</td><td><pre>vyatta@vyatta# set service portmonitor session 1 type
span</pre></td></tr>
<tr><td>Configure the source interface for the session. By default, the monitoring of the traffic direction is both ingress and egress.</td><td><pre>vyatta@vyatta# set service portmonitor session 1 source
dp0s3</pre></td></tr>
<tr><td>Configure the destination interface for the session.</td><td><pre>vyatta@vyatta# set service portmonitor session 1
destination dp0s4</pre></td></tr>
<tr><td>Commit the configuration.</td><td><pre>vyatta@vyatta# commit</pre></td></tr>
<tr><td>Save the configuration.
Port monitoring for the session is enabled if the <b>type</b>, <b>source</b> and <b>destination</b> parameters are configured properly.</td><td><pre>vyatta@vyatta# save</pre></td></tr>
<tr><td>Display the port mirroring configuration.</td><td><pre>vyatta@vyatta:~$ show portmonitor session
Session: 1
    Type: span
    State: enabled
    Source interfaces:
      Name: dp0s3
        Direction: both
    Destination interface: dp0s4</pre></td></tr>
</table>

RSPAN-source port mirroring

The following example shows a sample Brocade vRouter configuration for RSPAN-source port mirroring.

TABLE 30 Configuring RSPAN-source port mirroring

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Configure an RSPAN-source session.</td><td><pre>vyatta@vyatta# set service portmonitor session 2 type
rspan-source</pre></td></tr>
<tr><td>Configure the source physical interface for the session. By default, the monitoring of the traffic direction is both ingress and egress.</td><td><pre>vyatta@vyatta# set service portmonitor session 2 source
dp0s5</pre></td></tr>
<tr><td>Configure the destination VLAN interface for the session.</td><td><pre>vyatta@vyatta# set service portmonitor session 2
destination dp0s7 vif 700</pre></td></tr>
<tr><td>Commit the configuration.</td><td><pre>vyatta@vyatta# commit</pre></td></tr>
<tr><td>Save the configuration.
Port monitoring for the session is enabled if the <b>type</b>, <b>source</b> and <b>destination</b> parameters are configured properly.</td><td><pre>vyatta@vyatta# save</pre></td></tr>
<tr><td>Display the port mirroring configuration.</td><td><pre>vyatta@vyatta:~$ show portmonitor session
Session: 2
    Type: rspan-source
    State: enabled
    Source interfaces:
      Name: dp0s5
        Direction: both
    Destination interface: dp0s7.700</pre></td></tr>
</table>

RSPAN-destination port mirroring

The following example shows a sample Brocade vRouter configuration for RSPAN-destination port mirroring.

TABLE 31 Configuring RSPAN-destination port mirroring

Step	Command
Configure an RSPAN-destination session.	`vyatta@vyatta# set service portmonitor session 3 type rspan-destination`
Configure the source VLAN interface for the session. You cannot specify a direction for an RSPAN-destination session. The source VIF for the RSPAN-destination type must be part of a bridge group.	`vyatta@vyatta# set service portmonitor session 3 source dp0s7 vif 700`
Configure the destination physical interface for the session.	`vyatta@vyatta# set service portmonitor session 3 destination dp0s6`
Commit the configuration.	`vyatta@vyatta# commit`
Save the configuration. Port monitoring for the session is enabled if the **type**, **source** and **destination** parameters are configured properly.	`vyatta@vyatta# save`
Display the port mirroring configuration.	see below

```
vyatta@vyatta:~$ show portmonitor session
Session:                      3
    Type:                     rspan-destination
    State:                    enabled
    Source interfaces:
      Name:                   dp0s7.700
        Direction:            both
    Destination interface:    dp0s6
```

Port Monitoring Commands

service portmonitor session <id> description <string>

Specifies a description for the port-monitoring session.

Syntax

set service portmonitor session *id* **description** *string*

delete service portmonitor session *id* **description**

show service portmonitor session *id* **description**

Parameters

id

The number of the port-monitoring session.

string

A brief description of the session. If the description contains spaces, it must be enclosed in double quotation marks.

Modes

Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            description string
        }
    }
}
```

Usage Guidelines

Providing a description for a session can help you to quickly determine its purpose when viewing the configuration.

Use the **set** form of the command to specify a description for the port-monitoring session.

Use the **delete** form of the command to delete the description for the port-monitoring session.

Use the **show** form of the command to show the description for the port-monitoring session.

service portmonitor session <id> destination <interface>

Specifies the destination interface for the port-monitoring session.

Syntax **set service portmonitor session** *id* **destination** *interface-name* [**vif** *vid*]

delete service portmonitor session *id* **destination**

show service portmonitor session *id* **destination** *interface-name*

Parameters *id*

The number of the port-monitoring session.

interface-name

The name of a data plane interface. Following are the supported formats of the interface name:

- **dp***x***p***y***p***z*—The name of a data plane interface, where
  - **dp***x* specifies the data plane identifier (ID). Currently, only dp0 is supported.
  - **p***y* specifies a physical or virtual PCI slot index
  - **p***z* specifies a port index (for example, p1).

  For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- **dp***x***em***y*—The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- **dp***x***s***y*—The name of a data plane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.
- **dp***x***P***n***p***y***p***z*—The name of a data plane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

vif *vid*

For an RSPAN-source session only, specifies the VLAN interface.

Modes Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            destination interface_name {
                vif vid
            }
        }
    }
}
```

Usage Guidelines The destination interface is dedicated for monitoring and transmits only mirrored packets.

The destination interface for a SPAN or an RSPAN-destination session is a physical interface

The destination interface for an RSPAN-source session is a VIF interface.

Only one destination interface can be configured for SPAN, RSPAN-source, and RSPAN-destination in a monitoring session.

An interface cannot be both the source and destination of a monitoring session.

For SPAN and RSPAN, source and destination interfaces cannot be members of more than one monitoring session.

Use the **set** form of the command to specify the destination interface for the port-monitoring session.

Use the **delete** form of the command to remove the destination interface for the port-monitoring session.

Use the **show** form of the command to show the destination interface for the port-monitoring session.

service portmonitor session <id> disable

Disables the port-monitoring session.

Syntax

set service portmonitor session *id* **disable**

delete service portmonitor session *id* **disable**

show service portmonitor session *id*

Parameters

id

The number of the port-monitoring session.

Command Default

By default, the session is enabled if the session type, source interface, and destination interface are configured correctly.

Modes

Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            disable
        }
    }
}
```

Usage Guidelines

Use the **set** form of the command to disable the port-monitoring session.

Use the **delete** form of the command to re-enable the description for the port-monitoring session.

Use the **show** form of the command to show whether the port-monitoring session is disabled.

service portmonitor session <id> source <interface>

Specifies the source interface for the port-monitoring session.

Syntax

set service portmonitor session *id* **source** *interface-name* [**vif** *vid*]

delete service portmonitor session *id* **source** *interface-name*

show service portmonitor session *id* **source**

Parameters

id

The number of the port-monitoring session.

interface-name

The name of a data plane interface. Following are the supported formats of the interface name:

- **dp***x***p***y***p***z*—The name of a data plane interface, where
  - **dp***x* specifies the data plane identifier (ID). Currently, only dp0 is supported.
  - **p***y* specifies a physical or virtual PCI slot index
  - **p***z* specifies a port index (for example, p1).

  For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- **dp***x***em***y*—The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- **dp***x***s***y*—The name of a data plane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.
- **dp***x***P***n***p***y***p***z*—The name of a data plane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

vif *vid*

Specifies a VLAN interface.

Modes

Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            source interface_name {
                vif vid
            }
        }
    }
}
```

Usage Guidelines

For a SPAN or RSPAN-source session, you can configure more than one source interface for which ingress or egress mirroring is enabled. The source interface can be a physical or VLAN interface.

For an RSPAN-destination session, you can configure one source VLAN interface from which to receive the mirrored packets. The source VIF for the RSPAN-destination type must be part of a bridge group.

An interface cannot be both the source and destination of a monitoring session.

For SPAN and RSPAN, source and destination interfaces cannot be members of more than one monitoring session.

Use the **set** form of the command to specify the source interface for the port-monitoring session.

Use the **delete** form of the command to remove the source interface for the port-monitoring session.

Use the **show** form of the command to show the source interface for the port-monitoring session.

service portmonitor session <id> source <interface> direction <direction>

Specifies the direction of the port monitoring for a physical source interface of a SPAN or RSPAN-source session.

Syntax **set service portmonitor session** *id* **source** *interface-name* **direction** { **both** | **rx** | **tx** }

delete service portmonitor session *id* **source** *interface-name* **direction**

show service portmonitor session *id* **source** *interface-name* **direction**

Parameters *id*

The number of the port-monitoring session.

interface-name

The name of a data plane interface. Following are the supported formats of the interface name:

- **dp***x***p***y***p***z*—The name of a data plane interface, where
  - **dp***x* specifies the data plane identifier (ID). Currently, only dp0 is supported.
  - **p***y* specifies a physical or virtual PCI slot index
  - **p***z* specifies a port index (for example, p1).

  For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- **dp***x***em***y*—The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- **dp***x***s***y*—The name of a data plane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.
- **dp***x***P***n***p***y***p***z*—The name of a data plane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

both

Specifies that the interface mirrors and monitors the session in both ingress and egress directions.

rx

Specifies that the interface mirrors packets and monitors the session in the ingress direction.

tx

Specifies that the interface mirrors packets and monitors the session in the egress direction.

Command Default The default is both ingress and egress directions.

Modes Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            source interface_name {
                direction both | rx | tx
            }
        }
    }
}
```

Usage Guidelines

You cannot configure this command for an RSPAN-destination session.

Use the **set** form of the command to specify the direction that the physical source interface monitors the port-monitoring session.

Use the **delete** form of the command to remove the direction that the physical source interface monitors the port-monitoring session.

Use the **show** form of the command to show the direction that the physical source interface monitors the port-monitoring session.

service portmonitor session <id> source <interface> vif <vid> direction <direction>

Specifies the direction of the port monitoring for the VLAN source interface of a port-monitoring session.

Syntax **set service portmonitor session** *id* **source** *interface-name* **vif** *vid* **direction** { **both** | **rx** | **tx** }

delete service portmonitor session *id* **source** *interface-name* **vif** *vid* **direction** { **both** | **rx** | **tx** }

show service portmonitor session *id* **source**

Parameters *id*

The number of the port-monitoring session.

interface-name

The name of a data plane interface. Following are the supported formats of the interface name:

- **dp***x***p***y***p***z*—The name of a data plane interface, where
  - **dp***x* specifies the data plane identifier (ID). Currently, only dp0 is supported.
  - **p***y* specifies a physical or virtual PCI slot index
  - **p***z* specifies a port index (for example, p1).

  For example, dp0p1p2, dp0p160p1, and dp0p192p1.
- **dp***x***em***y*—The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3.
- **dp***x***s***y*—The name of a data plane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. Currently, this format applies only when using the KVM or Hyper-V platforms.
- **dp***x***P***n***p***y***p***z*—The name of a data plane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.

vif *vid*

Specifies a VLAN interface.

both

Specifies that the interface mirrors and monitors the session in both ingress and egress directions.

rx

Specifies that the interface mirrors packets and monitors the session in the ingress direction.

tx

Specifies that the interface mirrors packets and monitors the session in the egress direction.

Command Default

The default is both ingress and egress directions.

Modes

Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            source interface_name {
                vif vid {
                    direction both | rx | tx
                }
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of the command to specify the direction that the VLAN source interface monitors the port-monitoring session.

Use the **delete** form of the command to remove the direction that the VLAN source interface monitors the port-monitoring session.

Use the **show** form of the command to show the direction that the VLAN source interface monitors the port-monitoring session.

service portmonitor session <id> type <type>

Defines the identifier and type for the port-monitoring session.

Syntax

set service portmonitor session *id* **type span** | **rspan-source** | **rspan-destination**

delete service portmonitor session *id* [**type** [**span** | **rspan-source** | **rspan-destination**]]

show service portmonitor session *id* **type**

Parameters

id

The number of the session. The maximum number of sessions on the router is eight. Enter an integer from 1 to 31.

span

Specifies a SPAN session type.

rspan-source

Specifies an RSPAN-source session type.

rspan-destination

Specifies an RSPAN-destination session type.

Modes

Configuration mode

Configuration Statement

```
service {
    portmonitor {
        session id {
            type span|rspan-source|rspan-destination
        }
    }
}
```

Usage Guidelines

Port mirroring is enabled when the session type and the destination interface for the session are configured.

Use the **set** form of the command to define the port-monitoring session.

Use the **delete** form of the command to delete the port-monitoring session.

Use the **show** form of the command to show the identifier and type for the port-monitoring session.

show portmonitor session

Displays the port-monitoring configuration information for the session.

Syntax

show portmonitor session [*id*]

Parameters

id

The number of the port-monitoring session.

Modes

Operational mode

Usage Guidelines

Use this command to display the port-monitoring configuration information.

Examples

The following example shows the output of this command.

```
vyatta@vyatta:~$ show portmonitor session
Session:                      3
    Type:                     rspan-destination
    State:                    enabled
    Source interfaces:
      Name:                   dp0s7.700
        Direction:            both
    Destination interface:    dp0s6
```

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