

VYATTA, INC.

| **Vyatta OFR**

Vyatta OFR Quick Start Guide



Vyatta
Suite 160
One Waters Park Drive
San Mateo, CA 94403
vyatta.com

COPYRIGHT

Copyright © 2005–2007 Vyatta, Inc. All rights reserved.

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

PROPRIETARY NOTICE

The XORP License. © International Computer Science Institute, 2004–2007. © University College London, 2004–2007. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

ISSUE DATE: May 2007

DOCUMENT RELEASE: 2.1.1

DOCUMENT REVISION NO. 2.1 v02.

DOCUMENT PART NO. A0-0083-10-02

Contents

Quick Start Configuration	1
Starting Up	2
Taking a First Look	4
About Configuration	10
Configuring Basic System Information	16
Configuring Ethernet Interfaces	17
Configuring VLANs	19
Configuring System Services	21
Configuring Routing Protocols	23
Static Routes	23
Simple Routing Policies	24
RIP	27
OSPF	29
BGP	33
Configuring VRRP	37
Configuring NAT	40
Configuring Firewall Features	45
Installing to a Persistent Device	50
Saving Configuration	52
Saving Configuration to a Floppy Disk	52
Saving Configuration to a Hard Drive	53
Using the Web GUI	54
Topology Diagram	55

Quick Start Configuration

Thank you for choosing the Vyatta OFR. This document helps you get started and configure your router for the first time.

For more details, please consult the *Vyatta OFR Command Reference*.

The following topics are presented in this document:

- Starting Up
- Taking a First Look
- About Configuration
- Configuring Basic System Information
- Configuring Ethernet Interfaces
- Configuring VLANs
- Configuring System Services
- Configuring Routing Protocols
 - Static Routes
 - Simple Routing Policies

- RIP
- BGP
- OSPF
- Configuring VRRP
- Configuring NAT
- Configuring Firewall Features
- Installing to a Persistent Device

Starting Up

In this section:

- Running from LiveCD
- Begin powered down
- Log on as “vyatta”

You have the option of running the Vyatta OFR software in either of two modes:

- Directly from the CD (in LiveCD mode)
- By installing the software onto a partition on a hard drive. The hard drive can be an internal hard disk, a Flash, or a USB stick.

In this guide, we focus on running from the LiveCD, because that is the fastest and easiest way to get started.

If you would like to install the system to the hard disk, please see the section “Installing to a Persistent Device” on page 50.

▶ Running from LiveCD

Tip: See the documentation for your CD burning utility for information on burning an ISO image.

Your Vyatta representative can supply the OFR software on a CD, or you can download the software image from the Vyatta web site and create a CD. If you are burning a CD from downloaded software, make sure you burn the software onto the CD as an ISO image (not a file).

The LiveCD can run on a machine with an existing operating system without disturbing or changing the previously installed operating system. While you are running the router from a LiveCD, you will not be able to access other applications or programs on your machine.

The LiveCD runs the Vyatta software on a RAM disk on the host machine. The system uses the RAM disk for writeable sections of the file system, and the floppy drive or a TFTP server for configuration files.

► **Begin powered down**

Begin with your system powered down.

- 1 Connect a VGA monitor and keyboard while the system is still powered down.
- 2 Make sure your BIOS is configured to boot from the CD-ROM.
- 3 Insert the Vyatta OFR LiveCD into the CD drive and close the drive.
- 4 Power up the system.

► **Log on as "vyatta"**

After the startup messages complete, the login prompt appears:

```
vyatta login:
```

By default, the system has two user accounts predefined:

- A root user account named **root** with a password of **vyatta**. If you log on as **root**, you are logged on to the Linux shell. From there you can run the router shell (**xorpsh**) or execute Linux commands.
- A non-root user account named **vyatta** with a password of **vyatta**. If you log on as **vyatta**, you are logged directly into the router shell (**xorpsh**).

For the purposes of this tutorial, log on as **vyatta**. This launches the router shell, which provides a standard command-line interface for accessing router functions.

```
vyatta@vyatta's password:
Linux freebird.vyatta.com 2.6.19 #1 Wed Mar 14 00:39:02
PDT 2007 i686
Welcome to the Open Flexible Router.
The programs included with the OFR system are free
software;
the exact distribution terms for each program are
described in the individual files in
/usr/share/doc/*/copyright.
Last login: Tue Apr 24 21:49:03 2007 from
freebird.vyatta.com
Welcome to Vyatta on vyatta
vyatta@vyatta>
```

You have successfully started the router shell.

Tip: To keep your system secure, you should change the default passwords before connection to a production network.

Taking a First Look

In this section:

- Look at the available operational commands
- Enter configuration mode
- Exit configuration mode
- Display the default configuration
- Viewing long output (“More”)
- Show version information

When you first enter the router shell, you are in *operational mode*. In operational mode, you can issue commands—for example, to manually set the date and time. You can also view certain aspects of the router’s configuration, and monitor operation and function using **show** commands.

► **Look at the available operational commands**

The system offers command-line help and command completion.

- At the command prompt type the command completion query operator, which is the question mark (“?”).

```
vyatta@vyatta> ?
```

The system displays all the commands available to you in this context.

```
vyatta@vyatta> ?
vyatta@R1> ?
Possible completions:
  clear          Clear information in the system
  configure      Manipulate software configuration
  information
  date          Set system date and time
  delete        Delete system files
  exit          Exit the management session
  help          Provide help information
  init-floppy   Format and prepare a floppy to save
the config.boot file
  install        Install system features
  mount          Mount a filesystem
  ping          Ping a hostname or IP address
  ping6         Ping an IPv6 hostname or IPv6 address
  quit          Exit the management session
  reboot        Reboot the system
  show          Show system information
  traceroute    Trace route to hostname or IP address
```

```
tracert6          Trace the IPv6 route to a hostname
or IPv6 address
update           Update system features
vpn             VPN utilities
vyatta@R1>
vyatta@vyatta>
```

► Enter configuration mode

In configuration mode, you can view and change configuration for system functionality.

- Enter configuration mode, by typing **configure** at the command prompt:

```
vyatta@vyatta> configure
```

The system enters configuration mode, where you can **set** and **delete** configuration information.

Notice how the command prompt changes to keep you aware that you are in configuration mode. Also, the router lets you know if other users are logged on to configuration mode.

```
vyatta@vyatta> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
vyatta@vyatta#
```

► Exit configuration mode

Exiting configuration mode returns you to operational mode.

```
vyatta@vyatta# exit
[edit]
vyatta@vyatta>
```

If you made configuration changes while in configuration mode, you won't be able to exit configuration mode until you either commit the changes or discard them.

You commit configuration changes using the **commit** command.

```
vyatta@vyatta# exit
ERROR: There are uncommitted changes.
Use "commit" to commit the changes, or "exit discard" to
discard them.
```

```
[edit]
vyatta@vyatta# commit
[edit]
OK
vyatta@vyatta# exit
[edit]
vyatta@vyatta>
```

You discard configuration changes using the **exit discard** command.

```
vyatta@vyatta# exit
ERROR: There are uncommitted changes.
Use "commit" to commit the changes, or "exit discard" to
discard them.
[edit]
vyatta@vyatta# exit discard
[edit]
vyatta@vyatta>
```

► **Display the default configuration**

To display all current configuration, enter **show** in configuration mode:

```
vyatta@vyatta# show
```

Because you haven't configured anything yet, the values shown are the values set by default. These values will include configuration nodes for all the physical interfaces detected on your system. In the example shown, two physical Ethernet interfaces were detected, and the eth0 and eth1 configuration nodes have been created accordingly.

This example starts in operational mode, to remind you how to enter configuration mode. Again, notice how the command prompt changes when you enter configuration mode.

```
vyatta@vyatta> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
vyatta@vyatta# show
system {
  ntp-server "69.59.150.135"
  login {
    user root {
      authentication {
        encrypted-password: "$1$$Ht7gBYnxI1xCd0/JOnodh."
```

```
    }
  }
  user vyatta {
    authentication {
      encrypted-password: "$1$$Ht7gBYnxIlxCdO/JOnodh."
    }
  }
}
}
interfaces {
  loopback lo {
  }
}
service {
}
protocols {
}
policy {
}
firewall {
}
fea {
}
rtrmgr {
  config-directory: "/opt/vyatta/etc/config"
}
interfaces {
  ethernet eth0 {
  }
  ethernet eth1 {
  }
}
}
```

► Viewing long output ("More")

Sometimes, the configuration information will be too long for your screen, and the screen will show the "More" indication where the information breaks.

- To display the next line of configuration information when the "More" indication is showing, press **<Enter>**.
- To page forward one page, press **<Space>**.
- To page backward, press **b**.
- When all the output has been displayed, the "END" flag appears beside the "More" indicator. Press **q** to exit from the "More" display, as shown in Example 1-1.

Example 1-1 Exiting a "More" screen

```
[edit]
--More-- (END) q
vyatta@vyatta#
```

The full default configuration is as shown below.

```
vyatta@vyatta> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
vyatta@vyatta# show
system {
  ntp-server "69.59.150.135"
  login {
    user root {
      authentication {
        encrypted-password: "$1$Ht7gBYnxI1xCdO/JOnodh."
      }
    }
    user vyatta {
      authentication {
        encrypted-password: "$1$Ht7gBYnxI1xCdO/JOnodh."
      }
    }
  }
}
interfaces {
  loopback lo {
  }
}
```

```
service {
}
protocols {
}
policy {
}
firewall {
}
fea {
}
rtrmgr {
    config-directory: "/opt/vyatta/etc/config"
}
interfaces {
    ethernet eth0 {
    }
    ethernet eth1 {
    }
}
}
```

► **Show version information**

If you need to report a bug or request support, you will need to supply version information for your software. You can do this in operational mode, as follows:

- 1 If you are in configuration mode, return to operational mode.
-

```
vyatta@vyatta# exit
vyatta@vyatta>
```

- 2 Use the **show version** command to display version information.
-

```
vyatta@vyatta> show version
Version:      1.1-1
Built by:     autobuild@vyatta.com
Built on:     200612060031 -- Wed Dec  6 00:31:13 UTC 2006
System booted: Fri Dec  8 15:36:39 PST 2006
Uptime: 21:25:45 up 2 days, 5:49, 1 user, load average:
3.00, 1.46, 0.54
vyatta@vyatta>
```

About Configuration

In this section:

- Configuration nodes
- Configuration commands
- Committing configuration changes
- Saving configuration
- Initialize the floppy disk
- Save the configuration file to the floppy drive

The router configuration has a hierarchical tree form similar to the directory structure on a UNIX file system. The configuration tree consists of a series of configuration statements organized into *nodes*. There are three kinds of statements:

- Configuration nodes. These can be either:
 - Single nodes (just one instance can be created; for example, the **rip** protocol node)
 - Multi-nodes (more than one instance can be created; for example, **address** nodes)
- Attribute statements. These set the values or characteristics for parameters within a node.

► Configuration nodes

From the system's point of view, a configuration *node* is different from a simple configuration statement. Example 1-2 shows a configuration node containing attribute statements. In this example, **ssh** is the configuration *node*, and **port** and **protocol** are statements that specify values for attributes or parameters.

Configuration nodes have a pair of braces at the end (“{ }”). If the node has parameters or attributes that are configurable, the attribute statements are enclosed within the braces.

Example 1-2 A configuration node with attribute statements

```
ssh {
  port: 1-65534
  protocol-version: [v1|v2|all]
}
```

Sometimes a configuration node has empty braces at its end. This just means that the configuration node doesn't have any configurable attributes.

Example 1-3 shows the **dhcp-server** configuration node. Within this node, the **name** node is a configuration node, and it encloses all the other statements. The **start**, **exclude**, **static-mapping**, **dns-server**, and **wins-server** statements are also configuration nodes. You can tell this because those statements have braces. Of the configuration nodes with braces, the braces for the **start** configuration node enclose an attribute statement specifying the value for the **stop** attribute. The braces for the **static-mapping** configuration node enclose the attribute statements for **ip-address** and **mac-address**. The other configuration nodes (**exclude**, **dns-server**, and **wins-server**) don't have any configurable attributes, so their braces are empty.

Example 1-3 Configuration nodes with empty braces

```
dhcp-server {
  name text {
    interface: eth0..eth23
    network-mask: 0-32
    start ipv4 {
      stop: ipv4
    }
    exclude: ipv4 {}
    static-mapping: text {
      ip-address: ipv4
      mac-address: macaddr
    }
    dns-server ipv4 {}
    default-router: ipv4
    wins-server ipv4 {}
    lease: 120-4294967296
    domain-name: text
    authoritative: [enable|disable]
  }
}
```

► Configuration commands

When configuring, you will be entering statements prefaced with the following general configuration commands:

- **set.** The configuration tree is nearly empty when you first start up, except for a few automatically configured nodes. You must create a node for any functionality you want to configure on the router. You can do this using the **set** command.

You can also use the **set** command to create or modify the values for any properties or attributes within the configuration node.

One thing you can't do with the **set** command is change the identifier of a multi-node configuration node. (A multi-node is a configuration node where more than one instance can exist—such as multiple IP addresses for a vif or interface.) If a multi-node has an incorrect identifier, you'll need to delete the node and recreate it (using the **set** command) with the correct identifier.

- **delete.** The **delete** command removes the specified configuration node. If you create a node by mistake, you can just delete it and start over.

Remember that when you delete a configuration node, *all sub-nodes* are deleted as well. For example, if you issue a **delete protocols** command, you will delete *all* protocols you have configured so far, including RIP, BGP, OSPF, static routes, and SNMP configuration.

- **edit.** The **edit** command allows you to navigate down the configuration tree to a node you want to configure. This can save you typing if you are configuring a specific portion of the tree.

The following example configures an Ethernet interface by navigating down the configuration tree to the node for the interface, and editing from that location. The resulting commands are much simpler than if they were issued from the top of the configuration tree.

This example begins in operational mode and enters configuration mode.

Example 1-4 Navigating with the "edit" command

```
vyatta@vyatta> configure
Entering configuration mode.
There are no other users in configuration mode.
vyatta@vyatta# edit interfaces ethernet eth0
[edit interfaces/ethernet/eth0]
vyatta@vyatta# set description "my interface 1"
[edit interfaces/ethernet/eth0]
vyatta@vyatta# set address 172.16.0.65 prefix-length 24
[edit interfaces/ethernet/eth0]
vyatta@vyatta# show
> description: "\"my interface 1\""
```

```
> address 172.16.0.65 {
>   prefix-length: 24
> }

[edit interfaces/ethernet/eth0]
vyatta@vyatta# commit
OK
[edit interfaces/ethernet/eth0]
```

Notice the **[edit]** prompt that precedes the command prompt. The **[edit]** prompt provides you with a summary of your location within the configuration tree. This helps when editing deeply hierarchical configuration structures. At the beginning of this example, you are at the root of the configuration tree and the prompt displays like this:

```
[edit]
```

By the end of this example, you have navigated down through the configuration tree to the **interfaces ethernet eth0** node, and the prompt displays like this:

```
[edit interfaces ethernet eth0]
vyatta@R1#
```

— Use the **up** or **exit** command to travel one node up the configuration tree.

```
[edit interfaces ethernet eth0]
vyatta@R1# up
[edit interfaces ethernet]
vyatta@R1# up
[edit interfaces]
vyatta@R1# up
[edit]
vyatta@R1#
```

— Or, use the **top** command to move directly to the top of the configuration tree.

```
[edit interfaces ethernet eth0]
vyatta@R1# top
[edit]
vyatta@R1#
```

Note that to navigate to a node, the node must exist in the configuration tree; that is, it must already be created.

► **Committing configuration changes**

It is important to understand that on the Vyatta OFR, configuration changes do not take effect until you commit them, using the **commit** command.

The following example shows how the system flags uncommitted configuration changes. In this example, the description for interface eth0 is deleted and address 172.16.0.40 is added. Notice how the system flags deletions with a minus sign (“-”) and flags changes and additions with a greater-than sign (“>”).

Example 1-5 Uncommitted Changes

```
[edit interfaces/ethernet/eth0]
vyatta@vyatta# show
  description: "\"my interface 1\""
  address 172.16.0.65 {
    prefix-length: 24
  }

[edit interfaces/ethernet/eth0]
vyatta@vyatta# delete description
Deleting:
  description: "my interface 1"

OK

[edit interfaces ethernet eth0]
vyatta@vyatta# set address 172.16.0.40 prefix-length 24
[edit interfaces/ethernet/eth0]
vyatta@vyatta# show
-  description: "\"my interface 1\""
  address 172.16.0.65 {
    prefix-length: 24
  }
>  address 172.16.0.40 {
>    prefix-length: 24
>  }
```

```
[edit interfaces/ethernet/eth0]
vyatta@vyatta#
```

Commit changes using the **commit** command, as in the following example.

Example 1-6 Committing Changes

```
[edit interfaces/ethernet/eth0]
vyatta@vyatta# commit
OK
[edit interfaces/ethernet/eth0]
vyatta@vyatta# show
  address 172.16.0.65 {
    prefix-length: 24
  }
  address 172.16.0.40 {
    prefix-length: 24
  }

[edit interfaces/ethernet/eth0]
vyatta@vyatta#
```

When you have uncommitted configuration changes, you can only exit from configuration mode by committing or discarding the changes.

- To commit configuration changes, enter **commit** at the command prompt.
- To abandon your changes and exit from configuration mode, enter **exit discard** at the top level of configuration mode.

► Saving configuration

If you are running from LiveCD, you can only save the configuration to floppy disk. You cannot save the configuration to a hard drive or storage device such as Flash or a USB stick unless you have installed the system onto a partition on the device.

To save to floppy disk, you must initialize the floppy and then use the save command to save the configuration. For this procedure, please see “Saving Configuration to a Floppy Disk” on page 52.

If you have installed the system onto a persistent device (such as a hard drive, compact Flash, or USB stick), you can save the configuration onto that device. For this procedure, please see the section “Saving Configuration to a Hard Drive” on page 53.

Configuring Basic System Information

In this section:

- Enter host name and domain
- Specify a default gateway

► Enter host name and domain

This sequence sets the host name to R1 and the domain to mydomain.com.

```
vyatta@vyatta# set system host-name R1
[edit]
vyatta@vyatta# set system domain-name mydomain.com
[edit]
vyatta@vyatta# commit
OK
[edit]
vyatta@R1#
```

Notice how the command prompt changes to reflect the new host name.

► Specify a default gateway

This sequence specifies the default gateway. You do this by setting up a static route to the default router on network 0.0.0.0/0.

```
vyatta@R1# set protocols static route 0.0.0.0/0 next-hop
172.16.0.254
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1#
```

Configuring Ethernet Interfaces

In this section:

- Apply IP addresses to Ethernet interfaces
- Configure the loopback interface

In the Vyatta OFR router, most configuration can be applied either directly to the physical interface, or to a *virtual interface* (vif), which is a logical interface created for the physical interface. When the router starts up, it automatically detects the physical interfaces available on your device and creates configuration nodes for them. For example, on a system with two Ethernet interfaces, the router automatically creates configuration nodes for **eth0** and **eth1**.

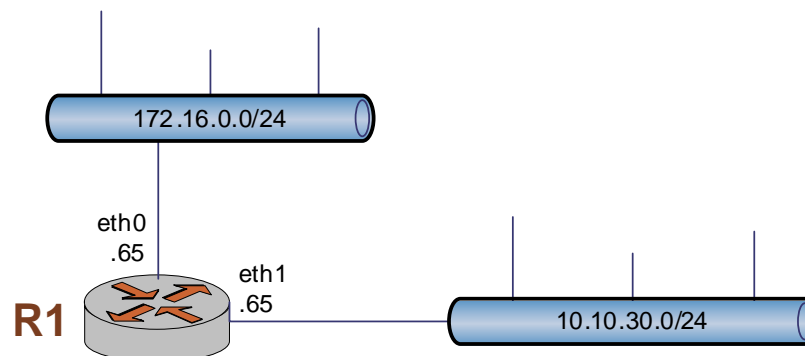
Ethernet vifs are used only when 802.1Q VLANs are to be supported. In a basic Ethernet configuration, such as that for trial or evaluation or for a simple network topology, it will often be simplest and adequate to apply IP addresses directly to the physical interface.

Each physical interface can have multiple IP addresses assigned to it. If you want to have multiple networks on the same physical interface (that is, if you want to use multinetting, but not VLANs), simply create multiple **address** configuration nodes directly under the primary interface.

This sequence applies IP addresses directly to the two Ethernet interfaces already configured for the system—eth0 and eth1. (These interfaces were automatically created by the system on startup.) Each IP address is applied directly to the interface.

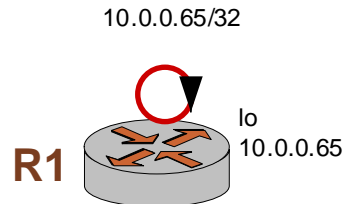
When you have finished, the Ethernet interfaces will be configured as in Figure 1-1.

Figure 1-1 Basic Ethernet interface configuration



The loopback interface will be configured as in Figure 1-2.

Figure 1-2 Loopback interface configuration



► Apply IP addresses to Ethernet interfaces

```
vyatta@R1# set interfaces ethernet eth0 address
172.16.0.65 prefix-length 24
[edit]
vyatta@R1# set interfaces ethernet eth1 address
10.10.30.65 prefix-length 24
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1#
```

► Configure the loopback interface

The loopback interface is automatically created when the system starts up. It is good network practice to assign an IP address to the loopback interface, as the loopback interface is the most reliable interface on the router.

In this example, the loopback interface is assigned an IP address of 10.0.0.65 on network 10.0.0.65/32, which includes just the one IP address.

When configuring the router, you should use the loopback address when reliability is particularly required. This includes using it to map to the router's host name, as the router ID in BGP and OSPF, and as the ID of iBGP peers.

```
vyatta@R1# set interfaces loopback lo address 10.0.0.65
prefix-length 32
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show interfaces
```

```
loopback lo {
  address 10.0.0.65 {
    prefix-length: 32
  }
}
ethernet eth0 {
  address 172.16.0.65 {
    prefix-length: 24
  }
}
ethernet eth1 {
  address 10.10.30.65 {
    prefix-length: 24
  }
}

[edit]
vyatta@R1#
```

Configuring VLANs

In this section:

- Apply VLAN tags to virtual interfaces
- Referring to vifs

VLANs are identified by a 4-byte tag that is inserted in the front of the Layer 2 Ethernet header. Having this additional tag means that interfaces configured for 802.1q are not compatible with standard Ethernet packets.

Like a physical Ethernet interface, each vif can have multiple addresses assigned to it. If you are using 802.1q VLANs, create vif configuration nodes beneath the physical interface and assign the IP address to the vif. If you are not using 802.1q, but you want to have multiple networks on the same physical interface (that is, you want to use multinetting, but not VLANs), simply create multiple **address** configuration nodes directly under the physical interface, without using vifs.

In the Vyatta OFR, an Ethernet interface may be used simultaneously as a standard port and an 802.1q port. To do this, configure a vif for the interface, and assign the VLAN ID for the interface to the vif. On Ethernet interfaces, a vif is always a VLAN interface, and its identifier is the VLAN ID.

This feature may not be compatible with all Ethernet switches: some switches require a physical Ethernet interface to be exclusively either a 802.1q interface or a standard Ethernet interface.

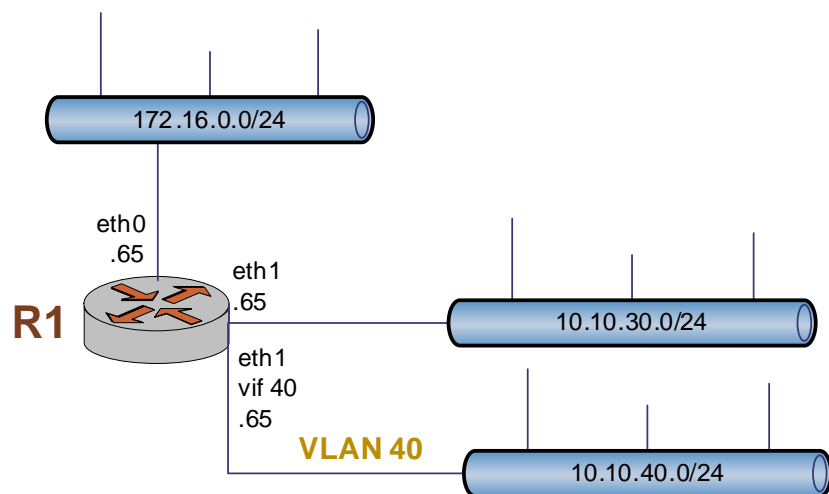
► **Apply VLAN tags to virtual interfaces**

This sequence configures a VLAN interface on router R1—vif 40 on eth1. The vif identifier is the VLAN ID, and this vif connects to VLAN 40. After configuring this VLAN, router R1 will have:

- One interface (eth0) that is configured as only a standard Ethernet interface
- One interface (eth1) that is configured as both a standard interface (IP address 10.10.30.65 applied directly to the interface) and as an 802.1q interface with one logical VLAN interface (IP address 10.10.40.65 applied to vif 40).

When you have finished, the interfaces will be configured as in Figure 1-3.

Figure 1-3 VLAN configuration



```
[edit]
vyatta@R1# set interfaces ethernet eth1 vif 40 address
10.10.40.65 prefix-length 24
[edit]
vyatta@R1# show interfaces ethernet
  ethernet eth0 {
    address 172.16.0.65 {
      prefix-length: 24
    }
  }
  ethernet eth1 {
    address 10.10.30.65 {
      prefix-length: 24
    }
  }
```

```
vif 40 {  
    address 10.10.40.65 {  
        prefix-length: 24  
    }  
}  
}  
  
[edit]  
vyatta@R1#
```

► Referring to vifs

When you refer to a vif within an **interfaces ethernet** command (such as `set interfaces ethernet` or `show interfaces ethernet`) you refer to it as ethernet *int-name* vif *vif-id*, as in the following example:

```
show interfaces ethernet eth1 vif 40
```

When you refer to the same vif within other commands, you refer to it as *int-name.vif-id*, as in the following example:

```
set protocols rip interface eth1.40 address 10.10.40.65
```

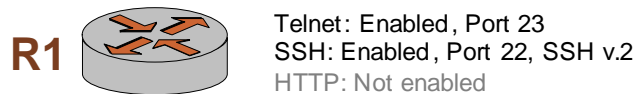
Configuring System Services

In this section:

- Enable Telnet access
- Enable SSH access
- Save configuration (optional)

When you have finished, services will be configured as in Figure 1-4.

Figure 1-4 Basic system services



► Enable Telnet access

Tip: If you wish, you can specify a non-standard port for Telnet. To see the full set of options available for the **service telnet** command, please see the Vyatta OFR Command Reference.

This sequence is optional, but creating the Telnet service will allow you to access the router remotely.

Telnet provides unencrypted communications between the router and another host. If you use SSH, we recommend that you disable Telnet access, which is not secure.

```
vyatta@R1# set service telnet
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1#
```

► Enable SSH access

Tip: If you wish, you can also configure a non-standard port for SSH. For SSH, you can also specify support for SSH version 1, version 2, or both. To see the full set of options available for the **service ssh** command, please see the Vyatta OFR Command Reference.

This sequence is optional, but enabling the SSH service will allow secure remote access to the router using the Secure Shell protocol. By default, SSH v.2 is enabled.

Example 1-7 Enabling SSH

```
vyatta@R1# set service ssh
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1#
```

► Save configuration (optional)

Tip: To save to a floppy disk, you must first initialize the floppy disk. (See page 52 for this procedure.)

If you have a floppy drive and want to save your configuration, you can use the following procedure.

```
vyatta@R1# save /mnt/floppy/config/config.boot
OK
vyatta@R1#
```

Configuring Routing Protocols

In this section:

- Static Routes
- Simple Routing Policies
- RIP
- BGP
- OSPF

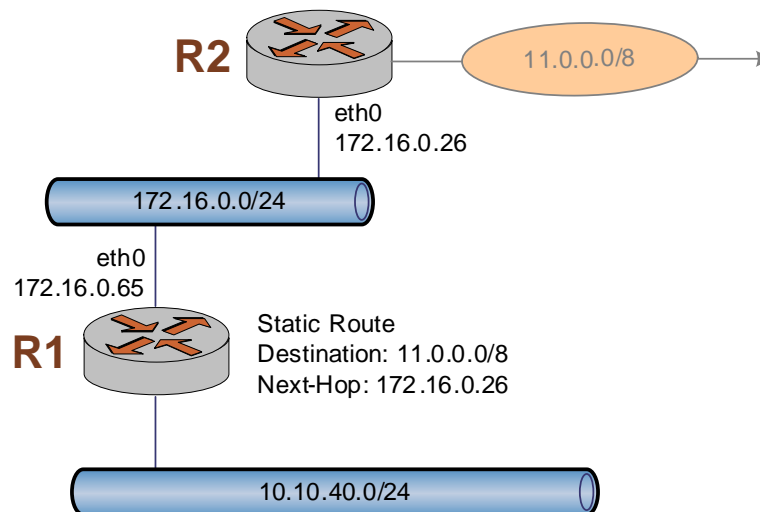
Static Routes

In this section:

- Configure a static route

This sequence sets a static route from R1 to network 11.0.0.0/8 directed towards 172.16.0.26. When you have finished, the static route will be configured as in Figure 1-5.

Figure 1-5 Static route configuration



► Configure a static route

```
vyatta@R1# set protocols static route 11.0.0.0/8 next-hop
172.16.0.26
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show protocols
static {
    route 11.0.0.0/8 {
        next-hop: 172.16.0.26
    }
}

[edit]
vyatta@R1#
```

Simple Routing Policies

In this section:

- Create a policy for exporting static routes
- Create a policy for exporting directly connected routes

A routing policy is a mechanism that allows you to configure criteria for comparing a route against, and for specifying the actions that will be performed on the route if the criteria are met. For example, a routing policy can be used to filter out (block) specific prefixes that are being announced by a BGP neighbor. Routing policies are also used to export routes learned via one protocol (for instance, OSPF) into another protocol (for instance, BGP). This is commonly called route redistribution.

Once a routing policy has been defined, then in order for the policy to take effect it must be applied to a specific routing protocol. Policies can be applied either as an import policy or as an export policy.

Import policies are evaluated for updates that are received via the routing protocol to which the policy has been applied. For example, if you configure an import policy for BGP, all BGP announcements received by the OFR are compared against first, prior to being added to the BGP and routing tables.

Export policies are evaluated for updates that are transmitted via the routing protocol to which the policy has been applied. For example, if you configure an export policy for BGP, all BGP updates originated by the OFR are compared against the policy prior to being sent to any BGP peers.

In this release if you want a routing protocol, such as RIP, to announce connected interfaces (including those with RIP configured) you must define a policy for redistributing connected routes.

Optionally, if you want to redistribute static routes you can also configure a policy to do that. In this release, you must define explicit policies for exporting static routes and directly connected routes. Once defined (as in this section), the policy must be applied to the individual routing protocol using the **import** or **export** directives.

► **Create a policy for exporting static routes**

In this step, you create the policy EXPORT_STATIC. This policy directs the routing protocol to redistribute all static routes.

```
vyatta@R1# set policy policy-statement EXPORT_STATIC term
10 from protocol static
[edit]
vyatta@R1# set policy policy-statement EXPORT_STATIC term
10 then action accept
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show policy
  policy {
    policy-statement "EXPORT_STATIC" {
      term 10 {
        from {
          protocol: "static"
        }
        then {
          action: "accept"
        }
      }
    }
  }

[edit]
vyatta@R1#
```

► **Create a policy for exporting directly connected routes**

In this step, you create the policy EXPORT_CONN. This policy directs the routing protocol to redistribute all directly connected routes.

```
vyatta@R1# set policy policy-statement EXPORT_CONN term 10
from protocol connected
[edit]
vyatta@R1# set policy policy-statement EXPORT_CONN term 10
then action accept
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show policy
policy {
  policy-statement "EXPORT_STATIC" {
    term 10 {
      from {
        protocol: "static"
      }
      then {
        action: "accept"
      }
    }
  }
  policy-statement "EXPORT_CONN" {
    term 10 {
      from {
        protocol: "connected"
      }
      then {
        action: "accept"
      }
    }
  }
}

[edit]
vyatta@R1#
```

Now you have two policies defined. You can apply them to individual routing protocols as required, as shown in subsequent sections.

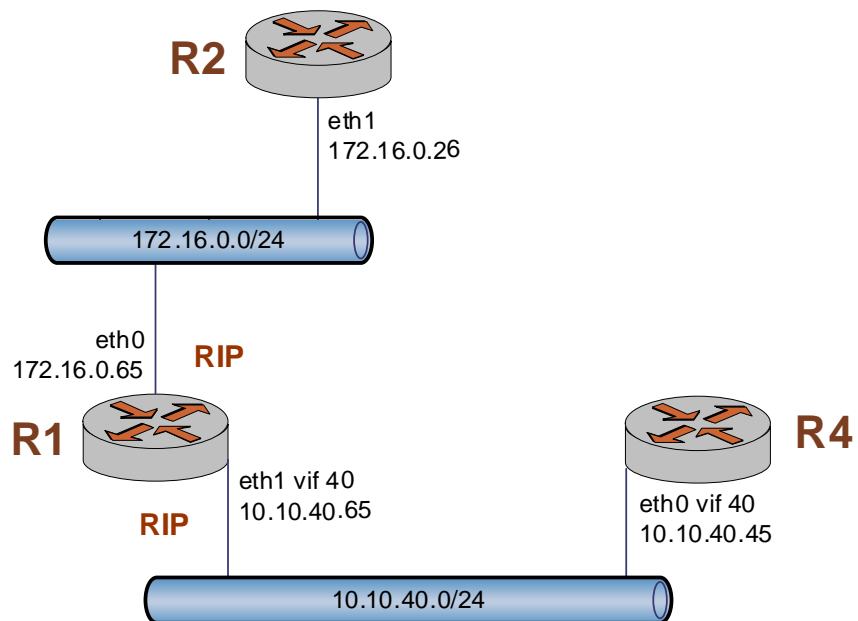
RIP

In this section:

- Configure RIP on eth0 and eth1.40
- Redistribute static and connected routes into RIP

The RIP protocol is enabled on IP addresses. These can be addresses defined directly on interfaces or (if you are using VLAN tagging) on Ethernet vifs. This sequence configures RIP on router R1. It enables RIP on eth0 and on eth1.40 (that is, vif 40 of eth1). When you have finished, RIP will be configured as in Figure 1-6.

Figure 1-6 Basic RIP configuration



► Configure RIP on eth0 and eth1.40

Tip: Note the notation for referring to vif 40 of interface eth1 within the **protocols** statement: **eth1.40**.

```
vyatta@R1# set protocols rip interface eth0 address
172.16.0.65
[edit]
vyatta@R1# set protocols rip interface eth1.40 address
10.10.40.65
[edit]
vyatta@R1# commit
OK
[edit]
```

```

vyatta@R1# show protocols rip
  interface eth0 {
    address 172.16.0.65 {
    }
  }
  interface "eth1.40" {
    address 10.10.40.65 {
    }
  }
}

[edit]
vyatta@R1#

```

► **Redistribute static and connected routes into RIP**

In this release, directly connected routes must be explicitly redistributed by applying a routing policy using the **export** directive within the RIP configuration. You can optionally also redistribute static routes.

This example applies the policy statements defined in the earlier section “Simple Routing Policies” on page 24.

Tip: Notice that there is no space between the “EXPORT_CONN” and the “EXPORT_STATIC” parameters.

```

vyatta@R1# set protocols rip export
EXPORT_CONN,EXPORT_STATIC
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show protocols rip export
  export: "EXPORT_CONN,EXPORT_STATIC"

[edit]
vyatta@R1#

```

To view information about RIP peers, use the **show rip peer** command in operational mode. This example uses the **show rip peer statistics all** option of this command.

```

vyatta@R1# quit
vyatta@R1> show rip peer statistics all
Last Active at Fri Jun  9 12:04:53 2006

Counter                                     Value
-----

```

Total Packets Received	25966
Request Packets Received	167
Update Packets Received	25799
Bad Packets Received	0
Authentication Failures	0
Bad Routes Received	0
Routes Active	2

To show RIP routes, use the **show route protocol rip** option in operational mode.

```
vyatta@R1> show route protocol rip
```

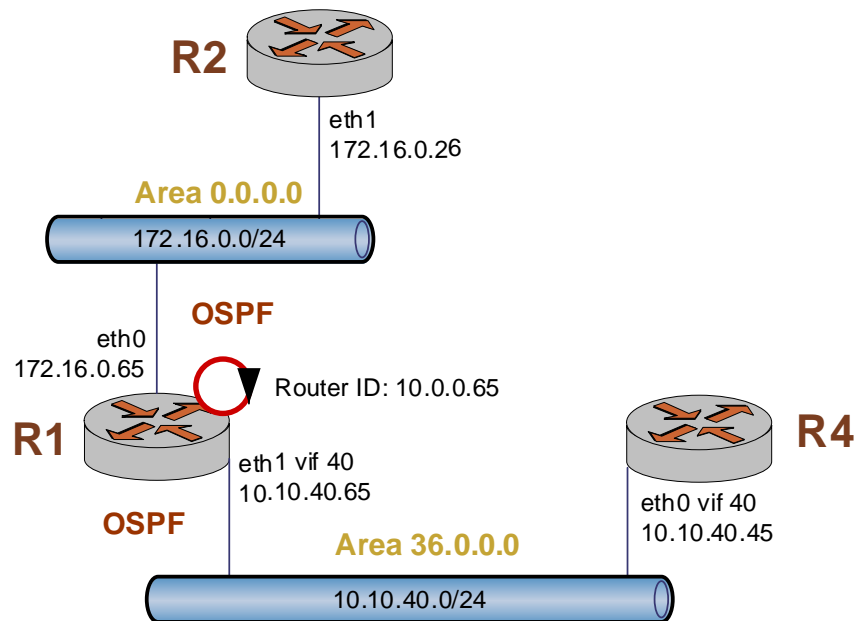
OSPF

In this section:

- Configure OSPF on two interfaces
- Redistribute static and connected routes into OSPF
- View OSPF information

This sequence configures OSPF on router R1. It creates two OSPF areas, where one interface is located in the backbone (area 0.0.0.0) and another interface is located in area 36.0.0.0, making this router an Area Border Router. When you have finished, OSPF will be configured on router R1 as in Figure 1-7.

Figure 1-7 Basic OSPF configuration



► Configure OSPF on two interfaces

This sequence establishes one interface in the backbone area and one interface in area 36.0.0.0, using the loopback interface (10.0.0.65) as the router ID.

Tip: The default hello interval is 10 seconds, the default dead-interval is 40 seconds, and the default priority is 128.

Tip: Again, note the notation for referring to vif 40 of interface eth1 within the **protocols** statement: **eth1.40**.

```
vyatta@R1# set protocols ospf4 router-id 10.0.0.65
[edit]
vyatta@R1# set protocols ospf4 area 0.0.0.0 interface eth0
address 172.16.0.65
[edit]
vyatta@R1# set protocols ospf4 area 36.0.0.0 interface
eth1.40 address 10.10.40.65
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show protocols ospf4
router-id: 10.0.0.65
  area 0.0.0.0 {
    interface eth0 {
      address 172.16.0.65 {
    }
  }
}
  area 36.0.0.0 {
```

```

        interface "eth1.40" {
            address 10.10.40.65 {
            }
        }
    }

[edit]
vyatta@R1#

```

► **Redistribute static and connected routes into OSPF**

In this release, directly connected routes must be explicitly redistributed by applying a routing policy using the **export** directive within the OSPF configuration. You can optionally also redistribute static routes.

This example applies the policy statements defined in the earlier section.

Tip: Notice that there is no space between the "EXPORT_CONN" and the "EXPORT_STATIC" parameters.

```

vyatta@R1# set protocols ospf4 export
EXPORT_CONN,EXPORT_STATIC
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show protocols ospf4 export
    export: "EXPORT_CONN,EXPORT_STATIC"
[edit]
vyatta@R1#

```

► **View OSPF information**

To view information about OSPF neighbors, use the **show ospf4 neighbor** command in operational mode.

```

vyatta@R1# quit
vyatta@R1> show ospf4 neighbor
Address      Interface State      ID           Pri  Dead
10.10.30.46  eth0      Full       192.168.2.44 3   39
10.1.0.49   eth0      Full       10.10.10.49  1   37
172.16.0.26 eth0      TwoWay     172.16.0.26 128  36

```

To show OSPF routes, use the **show route** option in operational mode.

```

vyatta@R1> show route
Total routes: 6, Total paths: 6

```

```

0.0.0.0/0      [static(1)]    > to 10.1.0.1    via
eth0
10.1.0.0/24   [connected(0)] > to 10.1.0.50   via
eth0
10.10.10.49/32 [ospf(2)]      > to 10.1.0.49   via
eth0
24.0.0.0/8    [ospf(1)]      > to 10.1.0.49   via
eth0
172.16.0.0/24 [connected(0)] > to 172.16.0.50 via
eth1
192.168.2.0/24 [ospf(2)]      > to 10.1.0.49   via
eth0

```

To view the OSPF Link State Advertisement (LSA) database, use the **show ospf4 database** command in operational mode.

Example 1-8 Showing the OSPF LSA database

```

vyatta@R1> show ospf4 database
OSPF link state database, Area 0.0.0.0
Type      ID          Adv Rtr      Seq          Age Opt  Cksum Len
Router    *172.16.0.65 172.16.0.65 0x80000002352 0x2    0x6b5836
Network   10.1.0.49    10.10.10.49 0x800002d0359 0x22   0x923440
Router    10.10.30.46 10.10.30.46 0x800002d0510 0x22   0x518f48
Router    172.16.0.26 172.16.0.26 0x80000005485 0x2    0x5e6348
ASExt-2   24.0.0.0     10.1.0.2    0x800001e2839 0x2    0x66d636

```

BGP

In this section:

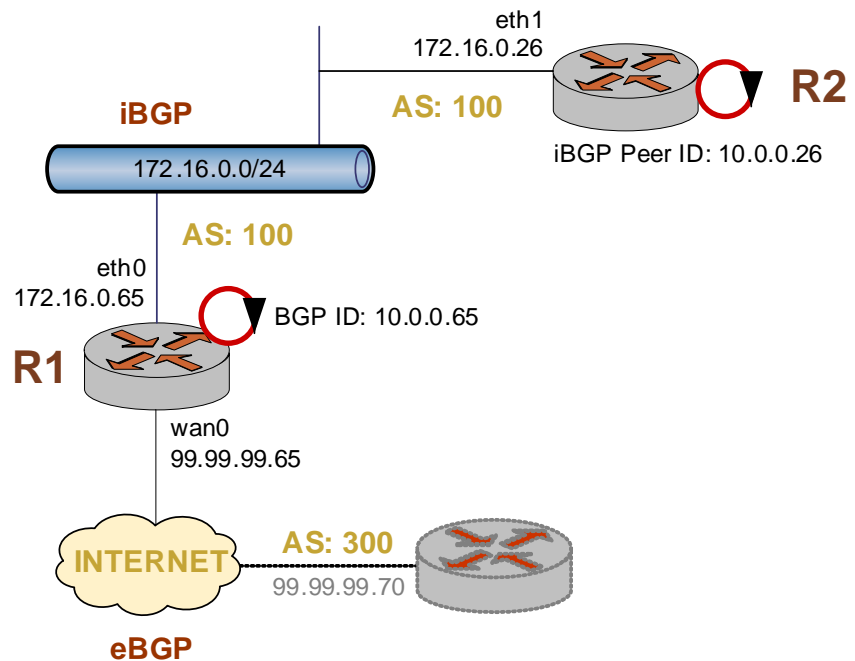
- Enable BGP and set the local AS
- Create the iBGP peer
- Create the eBGP peer
- View BGP configuration

This sequence sets up a basic BGP configuration, then defines one iBGP peer and one eBGP peer.

- For greatest reliability, the BGP router ID is set to the address of the loopback interface.
- The iBGP peer ID is set to the loopback address of the peer.
- The eBGP peer ID is set to the physical interface at the far end of the WAN link.

When you have finished, BGP will be configured as in Figure 1-8.

Figure 1-8 Basic BGP configuration



► **Enable BGP and set the local AS**

If you are in operational mode, re-enter configuration mode now, and then perform the following configuration.

```
vyatta@R1# set protocols bgp bgp-id 10.0.0.65
[edit]
vyatta@R1# set protocols bgp local-as 100
[edit]
vyatta@R1# commit
OK
[edit]
```

► **Create the iBGP peer**

The iBGP peer resides on the same LAN as router R1, and within the same AS. Router 1 advertises its loopback address to the peer for greatest reachability.

```
vyatta@R1# set protocols bgp peer 10.0.0.26 as 100
[edit]
vyatta@R1# set protocols bgp peer 10.0.0.26 local-ip
10.10.0.65
[edit]
vyatta@R1# set protocols bgp peer 10.0.0.26 next-hop
10.10.0.65
[edit]
vyatta@R1# commit
OK
[edit]
```

► **Create the eBGP peer**

The eBGP peer resides across the WAN, and in a different AS.

```
vyatta@R1# set protocols bgp peer 99.99.99.70 as 300
[edit]
vyatta@R1# set protocols bgp peer 99.99.99.70 local-ip
99.99.99.65
[edit]
vyatta@R1# set protocols bgp peer 99.99.99.70 next-hop
99.99.99.65
[edit]
vyatta@R1#
```

► **View BGP configuration**

You can view BGP configuration by using the **show protocols bgp** command in configuration mode.

```
vyatta@R1# show protocols bgp
  bgp-id: 10.0.0.65
  local-as: 100
  peer "10.0.0.26" {
    local-ip: "10.10.0.65"
    as: 100
    next-hop: 10.10.0.65
  }
  peer "99.99.99.70" {
    local-ip: "99.99.99.65"
    as: 300
    next-hop: 99.99.99.65
  }
}
```

► **View operational BGP information**

To view information about BGP peers, use the **show bgp peers** command in operational mode. This example uses the **show bgp peers detail** option of this command.

```
vyatta@R1# quit
vyatta@R1> show bgp peers detail
Peer 1: local 1/179 remote 172.16.0.26/179
  Peer ID: none
  Peer State: ACTIVE
  Admin State: START
  Negotiated BGP Version: n/a
  Peer AS Number: 1
  Updates Received: 0, Updates Sent: 0
  Messages Received: 0, Messages Sent: 0
  Time since last received update: n/a
  Number of transitions to ESTABLISHED: 3
  Time since last in ESTABLISHED state: 112 seconds
  Retry Interval: 120 seconds
  Hold Time: n/a, Keep Alive Time: n/a
  Configured Hold Time: 90 seconds, Configured Keep Alive
    Time: 30 seconds
  Minimum AS Origination Interval: 0 seconds
  Minimum Route Advertisement Interval: 0 seconds

Peer 2: local 10.10.30.65/179 remote 10.10.30.46/179
  Peer ID: none
  Peer State: ACTIVE
  Admin State: START
  Negotiated BGP Version: n/a
```

```
Peer AS Number: 5
Updates Received: 0, Updates Sent: 0
Messages Received: 0, Messages Sent: 0
Time since last received update: n/a
Number of transitions to ESTABLISHED: 0
Time since last in ESTABLISHED state: n/a
Retry Interval: 120 seconds
Hold Time: n/a, Keep Alive Time: n/a
Configured Hold Time: 90 seconds, Configured Keep Alive
Time: 30 seconds
Minimum AS Origination Interval: 0 seconds
Minimum Route Advertisement Interval: 0 seconds
```

To show iBGP routes, use the **show route protocol ibgp** option in operational mode.

```
vyatta@R1> show route protocol ibgp
Total routes: 43534, Total paths: 43534
3.0.0.0/8 [ibgp(0)] > to 192.168.1.26 via eth0
4.0.0.0/8 [ibgp(0)] > to 192.168.1.26 via eth0
4.0.0.0/9 [ibgp(0)] > to 192.168.1.26 via eth0
4.17.250.0/24[ibgp(0)] > to 192.168.1.26 via eth0
```

To show eBGP routes, use the **show route protocol ebgp** option in operational mode.

```
vyatta@R1> show route protocol ebgp
Total routes: 43534, Total paths: 43534

4.21.206.0/24[ebgp(0)] > to 192.168.1.26 via eth0
4.23.84.0/22 [ebgp(0)] > to 192.168.1.26 via eth0
4.23.112.0/24[ebgp(0)] > to 192.168.1.26 via eth0
4.23.113.0/24[ebgp(0)] > to 192.168.1.26 via eth0
4.23.114.0/24[ebgp(0)] > to 192.168.1.26 via eth0
4.36.100.0/23[ebgp(0)] > to 192.168.1.26 via eth0
```

Configuring VRRP

In this section:

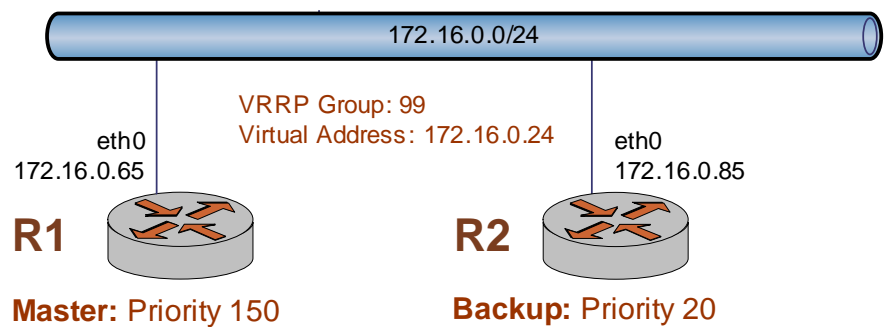
- Configure the first VRRP router (R1)
- Configure the second VRRP router (R2)
- View VRRP configuration

This sequence sets up a basic VRRP configuration between two Vyatta routers, using a virtual address of 172.16.0.99. Remember that in VRRP:

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

When you have finished, VRRP will be configured as in Figure 1-9.

Figure 1-9 Basic VRRP configuration



► **Configure the first VRRP router (R1)**

If you are in operational mode, re-enter configuration mode now, and then perform the following configuration.

```
vyatta@R1# set interfaces ethernet eth0 vrrp vrrp-group 99
[edit]
vyatta@R1# set interfaces ethernet eth0 vrrp
virtual-address 172.16.0.24
[edit]
vyatta@R1# set interfaces ethernet eth0 vrrp preempt true
[edit]
vyatta@R1# set interfaces ethernet eth0 vrrp priority 150
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show interfaces ethernet eth0 vrrp
    vrrp-group: 99
    virtual-address: 172.16.0.24
    priority: 150
    preempt: true

[edit]
vyatta@R1#
```

► **Configure the second VRRP router (R2)**

```
vyatta@R2# set interfaces ethernet eth0 vrrp vrrp-group 99
vyatta@R2# set interfaces ethernet eth0 vrrp
virtual-address 172.16.0.24
[edit]
vyatta@R2# set interfaces ethernet eth0 vrrp preempt true
[edit]
vyatta@R2# set interfaces interface eth0 vrrp priority 20
[edit]
vyatta@R2# commit
OK
[edit]
vyatta@R2# show interfaces interface eth0 vrrp
    vrrp-group: 99
    virtual-address: 172.16.0.24
    priority: 20
    preempt: true
[edit]
vyatta@R2#
```

► **View VRRP configuration**

You can use the **show vrrp** command in operational mode to view VRRP configuration. This example shows VRRP configuration for router R1.

```
vyatta@R1# quit
vyatta@R1> show vrrp
Physical interface: eth0, Address: 172.16.0.24
Interface state: up, Group: 99, State: master
Priority: 150, Advertisement interval: 1s, Authentication
type: none
Preempt: yes, VIP count: 1, VIP: 172.16.0.24
Advertisement timer: 429s, Master router: 172.16.0.65
Virtual MAC: 00:00:5E:00:01:63
```

Configuring NAT

In this section:

- Allow outbound traffic from 172.16.0.0/24
- Allow outbound traffic from 10.10.30.0/24
- Allow outbound traffic from 10.10.40.0/24
- Allow inbound SSH traffic to the public IP address
- View NAT Configuration

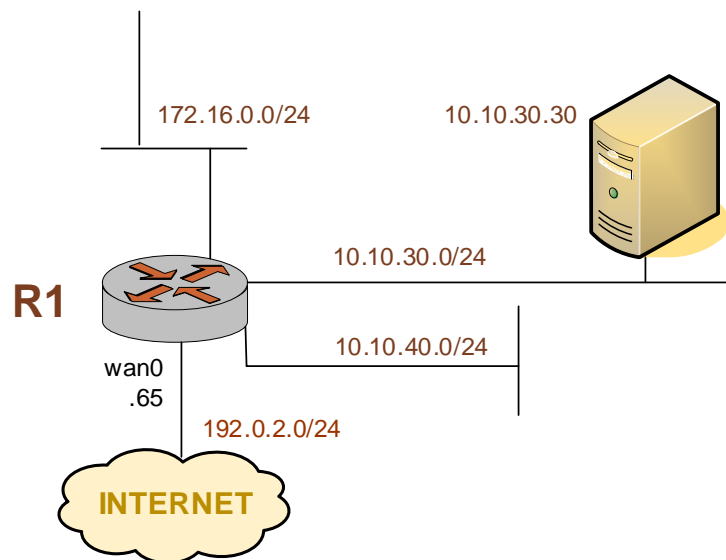
This sequence sets up a basic NAT configuration on router R1.

Rules 1, 2, and 3 of this configuration allow outbound traffic from three private subnets: 172.16.0.0/24, 10.10.30.0/24, and 10.10.40.0/24. Traffic from these subnets exits to the Internet through interface wan0 on R1.

Rule 4 of the configuration allows inbound SSH traffic to pass through wan0 on R1 to the target internal host for SSH, which is 10.10.30.30.

The subnets involved are shown in Figure 1-10.

Figure 1-10 Basic NAT configuration



► **Allow outbound traffic from 172.16.0.0/24**

Rule 1 allows traffic from network 172.16.0.0/24 to exit to the Internet through interface wan0 on R1.

If you are in operational mode, re-enter configuration mode now, and then perform the following configuration.

```
vyatta@R1# create service nat rule 1
[edit]
vyatta@R1# edit service nat rule 1
[edit service nat rule 1]
vyatta@R1# set type source
[edit service nat rule 1]
vyatta@R1# set translation-type masquerade
[edit service nat rule 1]
vyatta@R1# set outbound-interface wan0
[edit service nat rule 1]
vyatta@R1# set protocols all
[edit service nat rule 1]
vyatta@R1# set source network 172.16.0.0/24
[edit service nat rule 1]
vyatta@R1# set destination network 0.0.0.0/0
[edit service nat rule 1]
vyatta@R1# top
[edit]
```

► **Allow outbound traffic from 10.10.30.0/24**

Rule 2 allows traffic from network 10.10.30.0/24 to exit to the Internet through interface wan0 on R1.

```
vyatta@R1# create service nat rule 2
[edit]
vyatta@R1# edit service nat rule 2
[edit service nat rule 2]
vyatta@R1# set type source
[edit service nat rule 2]
vyatta@R1# set translation-type masquerade
[edit service nat rule 2]
vyatta@R1# set outbound-interface wan0
[edit service nat rule 2]
vyatta@R1# set protocols all
[edit service nat rule 2]
vyatta@R1# set source network 10.10.30.0/24
[edit service nat rule 2]
vyatta@R1# set destination network 0.0.0.0/0
[edit service nat rule 2]
```

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

► **Allow outbound traffic from 10.10.40.0/24**

Rule 3 allows traffic from network 10.10.40.0/24 to exit to the Internet through interface wan0 on R1.

```
vyatta@R1# create service nat rule 3
[edit]
vyatta@R1# edit service nat rule 3
[edit service nat rule 3]
vyatta@R1# set type source
[edit service nat rule 3]
vyatta@R1# set translation-type masquerade
[edit service nat rule 3]
vyatta@R1# set outbound-interface wan0
[edit service nat rule 3]
vyatta@R1# set protocols all
[edit service nat rule 3]
vyatta@R1# set source network 10.10.40.0/24
[edit service nat rule 3]
vyatta@R1# set destination network 0.0.0.0/0
[edit service nat rule 3]
vyatta@R1# top
[edit]
```

► **Allow inbound SSH traffic to the public IP address**

Rule 4 accepts SSH traffic directed at the public IP address of wan0 on R1 (99.99.99.65) to pass through to a single internal host at the private IP address 10.10.30.30.

Note that, in effect, this configuration “exports” the private server “outside” the protected network. This means that you will not be able to access the router from outside using SSH. That is, trying to access address 99.99.99.65 will now access the SSH server rather than the Vyatta OFR.

```
vyatta@R1# create service nat rule 4
[edit]
vyatta@R1# edit service nat rule 4
[edit service nat rule 4]
vyatta@R1# set type destination
[edit service nat rule 4]
vyatta@R1# set translation-type static
[edit service nat rule 4]
vyatta@R1# set inbound-interface wan0
```

```
[edit service nat rule 4]
vyatta@R1# set protocols tcp
[edit service nat rule 4]
vyatta@R1# set source network 0.0.0.0/0
[edit service nat rule 4]
vyatta@R1# set destination address 192.0.2.65
[edit service nat rule 4]
vyatta@R1# set destination port-name ssh
[edit service nat rule 4]
vyatta@R1# set inside-address address 10.10.30.30
[edit service nat rule 4]
vyatta@R1# top
[edit service nat rule 4]
```

► **View NAT Configuration**

To view NAT configuration, use the **show service nat command** in configuration mode.

```
vyatta@R1# show service nat
rule 1 {
  type: "source"
  translation-type: "masquerade"
  inbound-interface: "wan0"
  protocols: "all"
  source {
    network: 172.16.0.0/24
  }
  destination {
    network: 0.0.0.0/0
  }
}
rule 2 {
  type: "source"
  translation-type: "masquerade"
  inbound-interface: "wan0"
  protocols: "all"
  source {
    network: 10.10.30.0/24
  }
  destination {
    network: 0.0.0.0/0
  }
}
rule 3 {
  type: "source"
  translation-type: "masquerade"
```

```
inbound-interface: "wan0"
protocols: "all"
source {
    network: 10.10.30.0/24
}
destination {
    network: 0.0.0.0/0
}
}
rule 4 {
    type: "destination"
    translation-type: "static"
    inbound-interface: "wan0"
    protocols: "tcp"
    source {
        network: 0.0.0.0/0
    }
    destination {
        address: 99.99.99.65
        port-name: "ssh"
    }
    inside-address {
        address: 10.10.30.30
    }
}
}
```

Configuring Firewall Features

In this section:

- Firewall Example 1: Filter on source IP
- Firewall Example 2: Filter on source and destination IP
- Firewall Example 3: Filter on source IP and destination protocol
- Firewall Example 4: Define a network-to-network filter

This section sets up a basic firewall configuration. Essentially, this sequence defines a number of firewall rule sets allowing certain kinds of packets. All other packets are implicitly denied because of the final implicit **deny all** firewall rule.

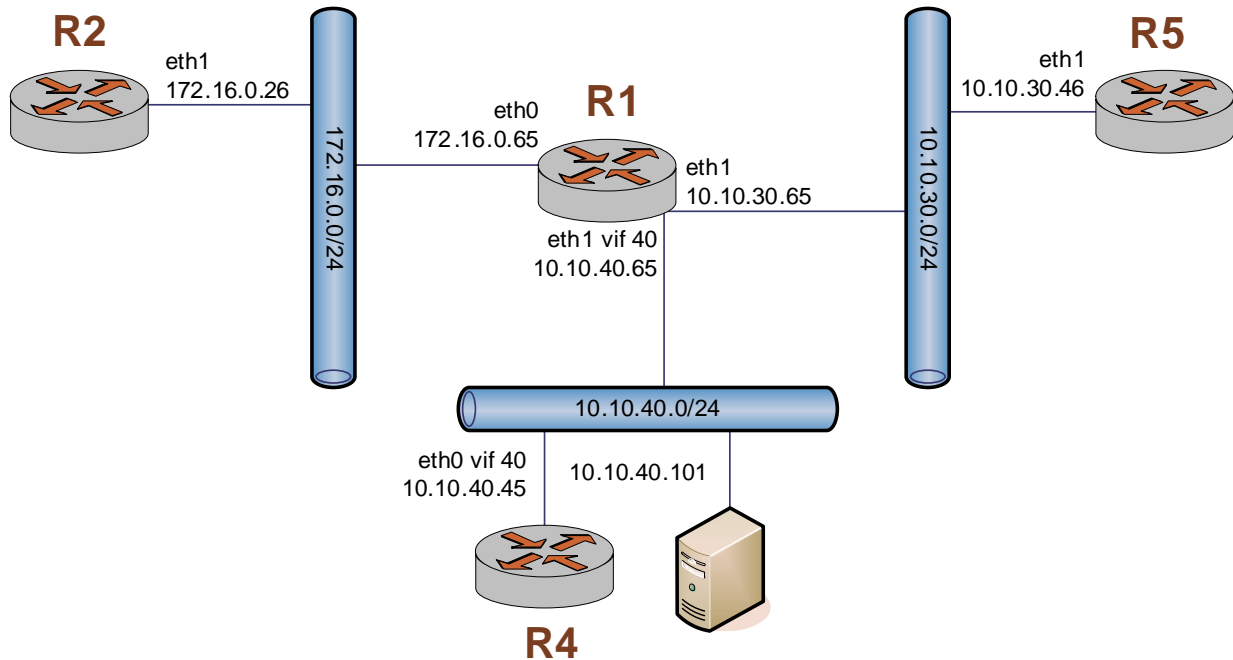
To configure firewall:

- 1 You define a number of named firewall rule sets, containing one or more firewall rules.
- 2 You apply the named rule sets to an interface or vif as packet filters. You can apply one named rule set as each of the following:
 - **in.** If you apply the rule set as **in**, the firewall will filter packets entering the interface or vif.
 - **out.** If you apply the rule set as **out**, the firewall will filter packets leaving the interface or vif.
 - **local.** If you apply the rule set as **local**, the firewall will filter packets destined for the router directly connected to this interface or vif.

When applying a firewall rule set, keep in mind that after the final user-defined rule in a rule set is executed, an implicit rule of **deny all** takes effect.

When you have finished, the firewall on router R1 will be configured as in Figure 1-11.

Figure 1-11 Basic firewall configuration



► **Firewall Example 1: Filter on source IP**

This sequence defines a firewall rule set for router R1 that contains one rule, which filters on source IP address only. This rule will accept packets coming from router R2. It then applies the firewall rule set to packets inbound on interface eth0.

```
vyatta@R1# set firewall name FWTEST-1 rule 1 action accept
[edit]
vyatta@R1# set firewall name FWTEST-1 rule 1 source
address 172.16.0.26
[edit]
vyatta@R1# set interfaces ethernet eth0 firewall in name
FWTEST-1
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show firewall name FWTEST-1
rule 1 {
    action: accept
    source {
        address: 172.16.0.26
    }
}
```

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

► **Firewall Example
2: Filter on source
and destination IP**

This sequence defines another firewall rule set for router R1. It contains one rule, which filters on both source and destination IP address. This rule accepts packets leaving R5 through eth1 using 10.10.30.46, and destined for 10.10.40.101. It then applies the firewall rule set to packets outbound from vif 40 on interface eth1.

```
vyatta@R1# set firewall name FWTEST-2 rule 1 action accept  
[edit]  
vyatta@R1# set firewall name FWTEST-2 rule 1 source  
address 10.10.30.46  
[edit]  
vyatta@R1# set firewall name FWTEST-2 rule 1 destination  
address 10.10.40.101  
[edit]  
vyatta@R1# set interfaces ethernet eth1 vif 40 firewall  
out name FWTEST-2  
[edit]  
vyatta@R1# commit  
OK  
[edit]  
vyatta@R1# show firewall name FWTEST-2  
rule 1 {  
    action: accept  
    source {  
        address: 10.10.30.46  
    }  
    destination {  
        address: 10.10.40.101  
    }  
}  
  
[edit]  
vyatta@R1#
```

► **Firewall Example 3: Filter on source IP and destination protocol**

This sequence defines a firewall rule for router R1 that filters on source IP address and destination protocol. This rule allows TCP packets originating from address 10.10.30.46 (that is, R5 eth1), and destined for the Telnet port of R1. The rule set is applied to local packets (that is, packets destined for this router, R1) through vif 30 on eth1.

```
vyatta@R1# set firewall name FWTEST-3 rule 1 action accept
[edit]
vyatta@R1# set firewall name FWTEST-3 rule 1 source
address 10.10.30.46
[edit]
vyatta@R1# set firewall name FWTEST-3 rule 1 protocol tcp
[edit]
vyatta@R1# set firewall name FWTEST-3 rule 1 destination
port-name telnet
[edit]
vyatta@R1# set interfaces ethernet eth1 firewall local
name FWTEST-3
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show firewall name FWTEST-3
rule 1 {
    action: accept
    source {
        address: 10.10.30.46
    }
    protocol: tcp
    destination {
        port-name: telnet
    }
}

[edit]
vyatta@R1#
```

► **Firewall Example 4: Define a network-to-network filter**

This sequence creates a network-to-network packet filter for router R1, allowing packets originating from 10.10.40.0/24 and destined for 172.16.0.0/24. It then applies the firewall rule set to packets inbound through vif 40 on interface eth1.

```
vyatta@R1# set firewall name FWTEST-4 rule 1 action accept
[edit]
vyatta@R1# set firewall name FWTEST-4 rule 1 source
network 10.10.40.0/24
[edit]
vyatta@R1# set firewall name FWTEST-4 rule 1 destination
network 172.16.0.0/24
[edit]
vyatta@R1# set interfaces ethernet eth1 vif 40 firewall in
name FWTEST-4
[edit]
vyatta@R1# commit
OK
[edit]
vyatta@R1# show firewall name FWTEST-4
rule 1 {
    action: accept
    source {
        network: 10.10.40.0/24
    }
    destination {
        network: 172.16.0.0/24
    }
}

[edit]
vyatta@R1#
```

Installing to a Persistent Device

The software image on the LiveCD includes an installation script for installing the router software onto a persistent device, such as a hard drive, compact Flash, or USB stick. This method installs the Vyatta operating system and all the routing and management software onto the persistent device.

The installer is an interactive install script that prompts you for some basic information and confirmation during the install.

To install the operating system and router software you need a minimum of 512 MB of free space on your hard drive for a root partition plus 10 MB for a configuration partition.

- 1 Log on to the system as **root**.
- 2 At the system command prompt (not the **xorps** command prompt) enter the following:

```
~ # install-system
```

The installer launches.

Example 1-9 shows a sample install session.

Example 1-9 Sample install session

```
~ # install-system  
Last login: Mon Dec 11 11:44:57 2006 from 192.168.247.175  
Linux vyatta 2.6.16 #1 Tue Dec 5 15:56:41 PST 2006 i686  
Welcome to the Open Flexible Router.
```

```
The programs included with the OFR system are free  
software;
```

```
the exact distribution terms for each program are  
described in the  
individual files in /usr/share/doc/*/copyright.
```

```
vyatta:~# install-system  
Welcome to the Vyatta install program. This script  
will walk you through the process of installing the  
Vyatta image to a local hard drive.
```

```
Would you like to continue? [Yes]:
```

```
Probing drives: OK
```

```
The Vyatta image will require a minimum 512 MB root  
partition and a minimum 10 MB configuration partition.  
Would you like me to try and partition a drive  
automatically
```

or would you rather partition it manually with parted? If you have already partitioned a drive, you may skip this step.

```
Partition (Auto/Parted/Skip) [Auto]:
```

```
I found the following drives on your system:
sda    1074MB
```

```
Install the image on? [sda]:
```

```
This will destroy all data on /dev/sda.
Continue? (Yes/No) [No]: Yes
```

```
How big of a root partition should I create? (512MB -
1064MB) [512]MB: 1000
```

```
Creating filesystem on /dev/sda1: OK
How big of a config partition should I create? (10MB -
74MB) [10]MB: 74
```

```
Creating config filesystem on /dev/sda2: OK
Mounting /dev/sda1: OK
Copying system image files to /dev/sda1: OK
OK
```

```
I need to install the GRUB bootloader.
I found the following drives on your system:
sda1074MB
```

```
Which drive should GRUB modify the boot partition on?
[sda]:
```

```
Setting up grub: OK
Done!
vyatta:~#
```

-
- 3 Remove the CD and reboot. When the system starts, it will be running from the local install.
 - 4 Start the XORP shell, which provides a standard command-line interface for accessing router functions. Enter **xorpsh** at the command prompt:

```
~ # xorpsh
```

The XORP shell starts and the router command prompt displays:

```
Welcome to Vyatta on vyatta
vyatta@vyatta>
```

You have successfully started the router shell.

Saving Configuration

In this section:

- Saving Configuration to a Floppy Disk
- Saving Configuration to a Hard Drive

Saving Configuration to a Floppy Disk

In this section:

- Initialize the floppy disk
- Save the configuration file to the floppy drive

This is the only way you can save configuration if you are running from LiveCD.

► Initialize the floppy disk

Tip: You should only format the floppy disk during your initial installation. If you format the floppy subsequently, you will lose all your saved configuration changes.

Before you can save the configuration from LiveCD to a floppy disk, you must initialize the floppy with a file system.

The floppy drive is referred to as **/dev/fd0**. It will be mounted in the directory **/mnt/floppy**.

- 1 Insert a blank floppy disk into the floppy disk drive.
- 2 At the router command prompt, enter the following:

```
vyatta@vyatta> init-floppy
```

The system prepares the floppy to receive configuration files. It also saves a copy of the current configuration to **/mnt/floppy/config/config.boot**. Within the **config.boot** file is the pointer to **/mnt/floppy/config** as the default configuration directory.

► Save the configuration file to the floppy drive

The floppy drive is referred to as **/dev/fd0**. It will be mounted in the directory **/mnt/floppy**, and the default configuration directory is **/mnt/floppy/config**.

Use the **save** command in configuration mode to save the configuration.

- If you want to be able to boot from the saved configuration, save the file to the default configuration directory under the file name **config.boot**. On a floppy disk, the default configuration directory is **/mnt/floppy/config**.

```
vyatta@vyatta# save /mnt/floppy/config/config.boot
```

- If you just want to store the configuration (for example, because you want the option to load it another time), you can use any file name and save it to any directory.

```
vyatta@vyatta# save /mnt/floppy/my_config1
```

Saving Configuration to a Hard Drive

You can only save to a hard drive if you have installed the system onto a partition on the drive. A hard drive may be a hard disk, a Flash, or a USB stick.

Use the **save** command to save the configuration. The system automatically saves the file into the default configuration directory. By default, this is the **/opt/vyatta/etc/config** directory.

- If you want to be able to boot from the saved configuration, save the file under the file name **config.boot**. This is the default. The following command:

```
vyatta@vyatta# save
```

saves the running configuration to **/opt/vyatta/etc/config/config.boot**.

- If you just want to store the configuration (for example, because you want the option to load it another time), you can use any file name and save to any directory.

```
vyatta@vyatta# save my_config1
```

This command saves the running configuration to **/opt/vyatta/etc/config/my_config1**.

Using the Web GUI

The OFR provides a web GUI based on the same XORP API as the command-line interface (CLI). To access the GUI:

- 1 Enable the HTTP service on the router.

```
vyatta@R1# set service http
[edit]
vyatta@R1# commit
[edit]
OK
vyatta@R1#
```

- 2 Configure an IP address for the router.
- 3 Point your web browser at the router's IP address.

Any non-root user may log on to the router through the GUI. Root login through the GUI is not supported at this time.

The default user ID/password for the initial non-root user is **vyatta/vyatta**. When this password is changed (using either the CLI or the GUI), it is changed for both the CLI and the GUI. If more users are added, the GUI will authenticate them as well.

Any operation that can be performed through the CLI can also be performed through the GUI, with the exception of the initial enabling of the HTTP service. This can only be performed through the CLI or the root console.

Topology Diagram

This section shows the topology used in examples in Vyatta documentation.

