Release 1.0

Vyatta OFR Quick Start Guide

A0-0065-10-01

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

ISSUE DATE: July 24, 2006

DOCUMENT REVISION NO. Rel 1.0 v.01. This document was released with Build 2962.

Contents

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 the Hard Disk

Starting Up

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 your hard disk

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 the Hard Disk" on page 43.

▶ Running from LiveCD

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).

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

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 "root"

After the startup messages complete, the login prompt appears:

```
vyatta login:
```

By default, the system has two user accounts predefined:

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

- 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**).

1. For the purposes of this tutorial, log on as **root**. The system command prompt displays:

   ```
   vyatta login: root
   Password: vyatta
   ~ #
   ```

2. 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
   root@vyatta>
   ```

   You have successfully started the router shell.

▶ Format a floppy disk (optional)

If you are just trying out the system, you may not need to save your configuration changes. In that case, you can skip this step.

However, if you do want to save your configuration and you are running from LiveCD, you can save your changes onto a floppy disk. To do this, you must format the floppy disk to make it accessible to the Vyatta software.

***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.*

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:

```
root@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.

Taking a First Look

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 (“?”).

```
root@vyatta> ?
```

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

```
root@vyatta> ?
Possible completions:
   clear        Clear information in the system
   configure    Manipulate software configuration
      information
   date         Set system date and time
   delete       Delete logs
   exit         Exit the management session
   help         Provide help information
   init-floppy  Format and prepare a floppy
```

```
      to save the config.boot file
   package      Package management functions
   ping         Ping a hostname or IP address
   quit         Exit the management session
   reboot       Reboot the system
   show         Show system information
   traceroute   Trace route to hostname or IP address
root@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:

  ```
  root@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.

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

▶ Display the default configuration

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

```
root@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.

```
[edit]
root@vyatta# show
protocols {
}
policy {
}
interfaces {
   loopback lo {
}
   ethernet eth0 {
}
   ethernet eth1 {
}
firewall {
}
service {
}
system {
   ntp-server “ntp.vyatta.com”
   login {
      user root {
         authentication {
            encrypted-password:
“$1$$Ht7gBYnxI1xCd0/JOnodh.”
         }
      }
}
      user vyatta {
         authentication {
            encrypted-password:
“$1$$Ht7gBYnxI1xCd0/JOnodh.”
         }
      }
   }
   package {
--More--
```

Viewing Long Output ("More")

Most likely, 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.

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

The full default configuration is as shown below.

```
protocols {
}
policy {
}
interfaces {
   loopback lo {
}
   ethernet eth0 {
}
firewall {
}
service {
}
system {
   ntp-server “ntp.vyatta.com”
   login {
      user root {
         authentication {
            encrypted-password:
“$1$$Ht7gBYnxI1xCd0/JOnodh.”
         }
      }
}
      user vyatta {
         authentication {
```

```
                    encrypted-password:
“$1$$Ht7gBYnxI1xCd0/JOnodh.”
                }
            }
        }
        package {
            repository “vyatta/1.0” {
                host-name: “archive.vyatta.com”
                component main {
                    description: “1.0-MainPackages”
                }
                component security {
                    description: “1.0-SecurityUpdates”
                }
            }
        }
    }
    rtrmgr {
        config-directory: “/config”
    }

    [edit]
    --More-- (END)
```

▶ Show version information

If you need 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.

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

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

```
root@vyatta> show version
Revision: 1.0 (2894M)
Image built:Mon Jul 31 14:34:50 PDT 2006
System booted: Tue Aug 1 22:36:46 UTC 2006
Uptime: 23:16:49 up 40 min, load average: 0.00, 0.84,
   0.93
root@vyatta>
```

About Configuration

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 1-1 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 configuratble, the attribute statements are enclosed within the braces.

Example 1-1 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 that doesn't have any configurable attributes.

Example 1-2 shows the **dhcp** configuration node. Within this node, the **start**, **dns-server**, **wins-server**, and **relay** statements are also configuration nodes. You can tell this because those statements have braces. Of the braces, the braces for the **start** configuration node enclose an attribute statement specifying the value for the **stop** attribute. The other configuration nodes (**dns-server**, **wins-server**, and **relay**) don't have any configurable attributes, so their braces are empty.

Example 1-2 Configuration nodes with empty braces

```
dhcp-server {
   name text {
      start ipv4 {
         stop: ipv4
      }
      network-mask: 0–32
      dns-server ipv4 {}
      default-router: ipv4
      wins-server ipv4 {}
      lease: 120–4294967296
      interface: text
      domain-name: text
      authoritative: [enable|disable]
      relay ipv4 {}
   }
}
```

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.

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

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

Notice the [edit] prompt that precedes the command prompt. The [edit] prompt reminds you of your position within the configuration tree. At this point, you are at the root of the tree. At the beginning of this example, you are at the root of the configuration and the prompt displays like this:

```
[edit]
```

By the end of this example, you have navigated down through the configuration treed to the interfaces ethernet eth0 node, and the are at the root of the configuration and the prompt displays like this:

```
[edit interfaces ethernet eth0]
```

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

— Use the **top** command to move to the top of the configuration tree.

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 deletions with a greater-than sign (“>”).

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

[edit interfaces ethernet eth0]
root@vyatta# delete description
Deleting:
   description: “my interface 1”

OK
[edit interfaces ethernet eth0]
root@vyatta# set address 172.16.0.63 prefix-length 24
[edit interfaces ethernet eth0]
root@vyatta# show
-  description: “\“my interface 1\””
   address 172.16.0.65 {
      prefix-length: 24
   }
>  address 172.16.0.63 {
>     prefix-length: 24
>  }

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

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

```
[edit interfaces ethernet eth0]
root@vyatta# commit
OK
[edit interfaces ethernet eth0]
root@vyatta# show
   address 172.16.0.65 {
      prefix-length: 24
   }
   address 172.16.0.63 {
      prefix-length: 24
   }

[edit interfaces ethernet eth0]
root@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.

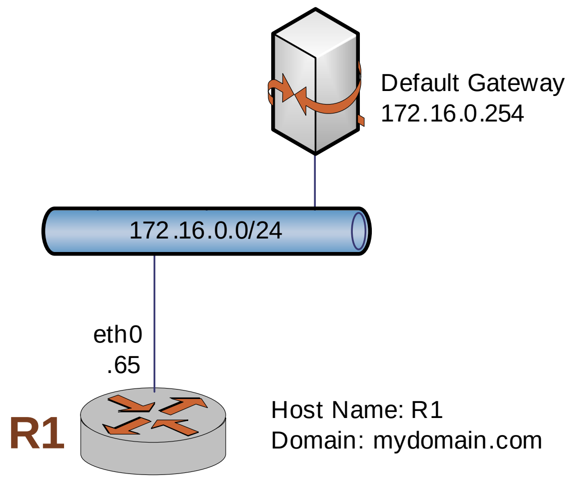
Configuring Basic System Information

In this section, you configure some basic system information. You will:

- Enter host name, domain, and default gateway

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

Figure 1-1 Basic system configuration



▶ Enter host name, domain, and default gateway

This sequence sets the host name to R1, the domain to mydomain.com, and specifies a default gateway for the router at 172.16.0.254.

```
root@vyatta# set system host-name R1
[edit]
root@vyatta# set system domain-name mydomain.com
[edit]
root@vyatta# set system gateway-address 172.16.0.254
[edit]
root@vyatta# commit
OK
[edit]
root@R1#
```

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

Configuring Ethernet Interfaces

In the Vyatta OFR router, most configuration can 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**.

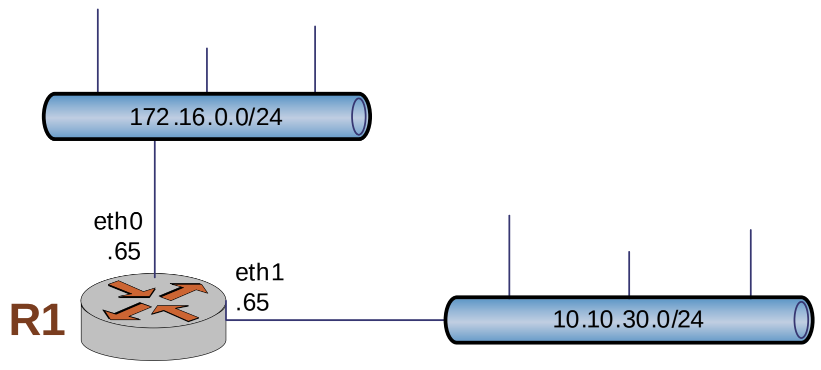
Ethernet vifs are used only when 802.1Q VLANs are to be supported. In a basic Ethernet configuration, such as that for trial or evaluation or for a simple network topology, it will often be simplest and adequate to apply IP addresses directly to the physical interface.

Each physical interface can have multiple 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 interfaces will be configured as in Figure 1-2.

Figure 1-2 Basic interface configuration



▶ Apply IP addresses to Ethernet interfaces

```
root@R1# set interfaces ethernet eth0 address
172.16.0.65 prefix-length 24
[edit]
root@R1# set interfaces ethernet eth1 address
10.10.30.65 prefix-length 24
[edit]
root@R1# commit
OK
[edit]
root@R1# show interfaces
   loopback lo {
   }
   ethernet eth0 {
      address 172.16.0.65 {
         prefix-length: 24
      }
   }
   ethernet eth1 {
      address 10.10.30.65 {
         prefix-length: 24
      }
   }

[edit]
root@R1#
```

Configuring VLANs

In the Vyatta router, most configuration can applied either directly to the interface, or to a *virtual interface* (vif). A vif has its own configuration node, which is subordinate to the interface node.

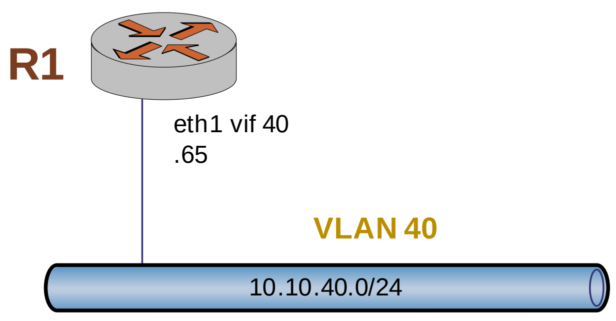
On Ethernet interfaces, a vif is always a VLAN interface, and its identifier becomes the VLAN ID. Vif identifiers are used to specify 802.1Q VLAN tags. Only tagged packets are received on vifs configured on Ethernet interfaces.

Like a physical 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.

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.

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

Figure 1-3 VLAN configuration



```
[edit]
root@R1# set interfaces ethernet eth1 vif 40 address
10.10.40.65 prefix-length 24
[edit]
root@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]
root@R1#
```

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

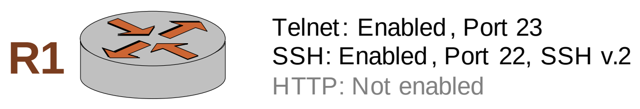
In this section, you configure some basic system information. You will:

- Enable Telnet access
- Enable SSH access

Then you review, commit, and save the configuration.

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

Figure 1-4 Basic system services



▶ Enable Telnet access

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

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

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.*

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

▶ Enable SSH access

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

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.*

Example 1-3 Enabling SSH

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

▶ Save configuration (optional)

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

Tip: *To save to a floppy disk, you must first initialize the floppy disk. (See page 4 for this procedure.)*

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

Configuring Routing Protocols

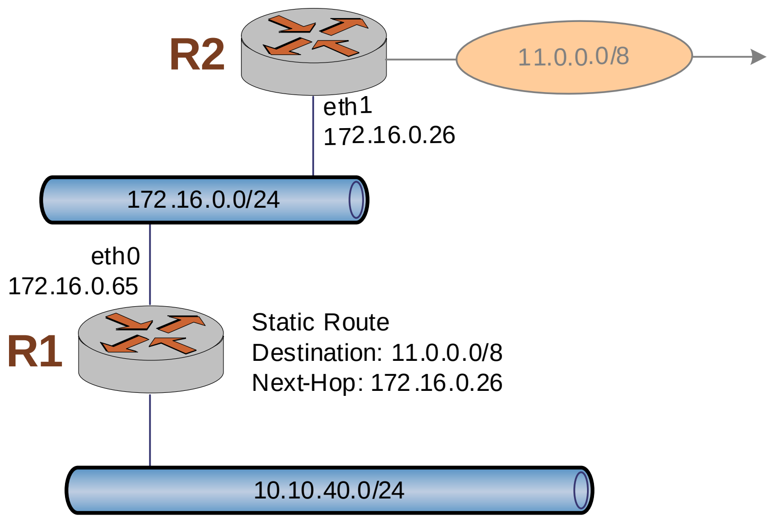
In this section, you create simple routing policies, and then configure the following:

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

Static Routes

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

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

[edit]
root@R1#
```

Simple Routing Policies

Routing policies allow you to manipulate the default behavior of routing protocols to determine how routes are imported into the routing table and exported into other routing protocols. For example, an export policy can be applied to RIP to redistribute routes learned from another protocol, such as OSPF. In this section you define two routing policies for redistributing routes.

When you define a routing policy, you specify the criteria which, if matched, will cause the action specified in the policy. Not every criterion applies to every routing protocol.

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.

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

[edit]
root@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.

```
root@R1# set policy policy-statement EXPORT_CONN term
10 from protocol connected
[edit]
root@R1# set policy policy-statement EXPORT_CONN term
10 then action accept
[edit]
root@R1# commit
OK
[edit]
root@R1# show policy
   policy {
```

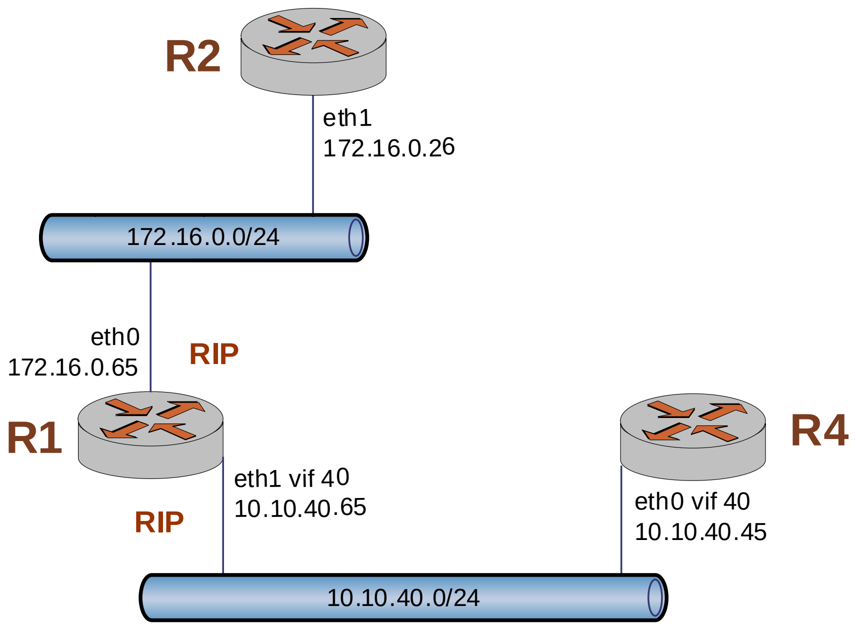
```
            policy-statement “EXPORT_STATIC” {
               term 10 {
                  from {
                     protocol: "static"
                  }
                  then {
                     action: “accept”
                  }
               }
            }
            policy-statement “EXPORT_CONN” {
               term 10 {
                  from {
                     protocol: "connected"
                  }
                  then {
                     action: “accept”
                  }
               }
            }
         }

   [edit]
   root@R1#
```

Now you have two policies defined. You can apply them to individual routing protocols as required.

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__.

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

```
        interface “eth1.40” {
              address 10.10.40.65 {
              }
           }
        }
     }

     [edit]
     root@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 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 22.

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

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

[edit]
root@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.

```
root@R1# quit
root@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
```

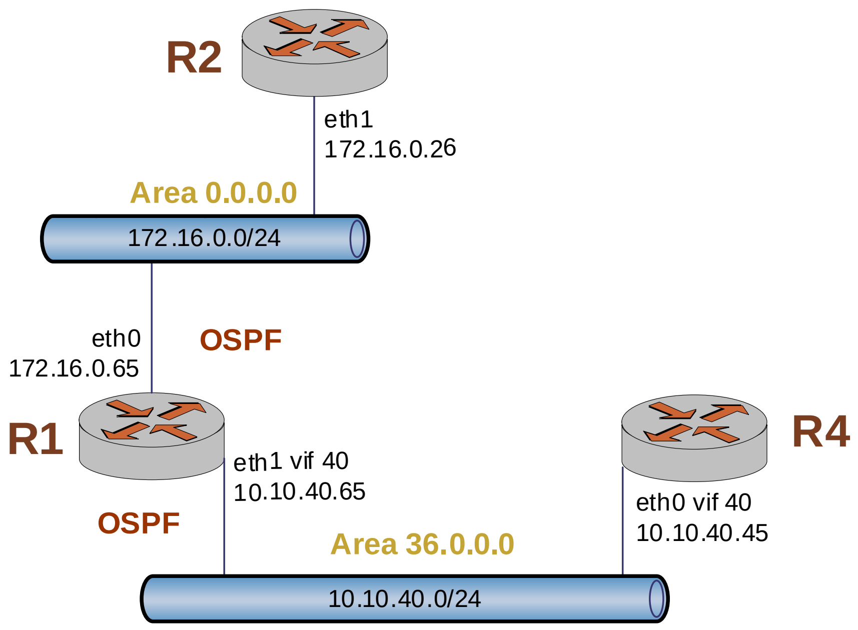
To show RIP routes, use the **show route protocol rip** option in operational mode.

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

OSPF

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, with a router ID using its highest-numbered assigned IP address (172.16.0.65).

***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**.*

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

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

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

```
root@R1# quit
root@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.

```
root@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.

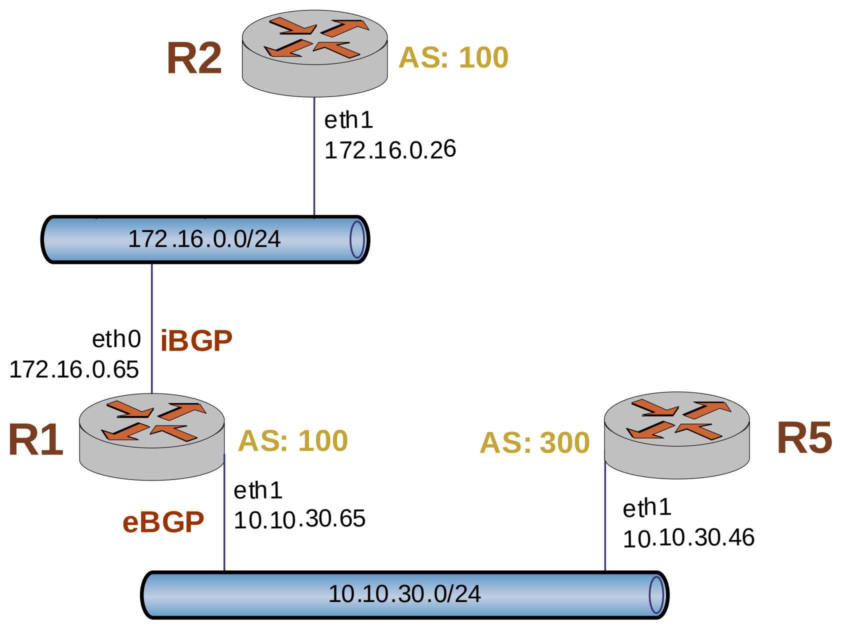
Example 1-4 Showing the OSPF LSA database

```
root@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

This sequence sets up a basic BGP configuration with one iBGP peer and one eBGP peer. When you have finished, BGP will be configured as in Figure 1-8.

Figure 1-8 Basic BGP configuration



▶ Create a basic iBGP configuration

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

```
root@R1# set protocols bgp bgp-id 172.16.0.65
[edit]
root@R1# set protocols bgp local-as 100
[edit]
root@R1# set protocols bgp peer 172.16.0.26 as 100
[edit]
root@R1# set protocols bgp peer 172.16.0.26 local-ip
172.16.0.65
[edit]
root@R1# set protocols bgp peer 172.16.0.26 next-hop
172.16.0.65
[edit]
root@R1#
```

▶ Create a basic eBGP configuration

```
root@R1# set protocols bgp peer 10.10.30.46 as 300
[edit]
root@R1# set protocols bgp peer 10.10.30.46 local-ip
10.10.30.65
[edit]
root@R1# set protocols bgp peer 10.10.30.46 next-hop
10.10.30.65
[edit]
root@R1#
```

▶ View BGP configuration

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

```
root@R1# show protocols bgp
   bgp-id: 172.16.0.65
   local-as: 5
   peer "172.16.0.26" {
      local-ip: "172.16.0.65"
      as: 5
      next-hop: 172.16.0.65
   }
```

```
    peer "10.10.30.46" {
        local-ip: "10.10.30.65"
        as: 1
        next-hop: 10.10.30.65
}
```

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.

```
root@R1# quit
root@R1> show bgp peers detail
Peer 1: local 172.16.0.65/179 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.

```
root@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.

```
root@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
```

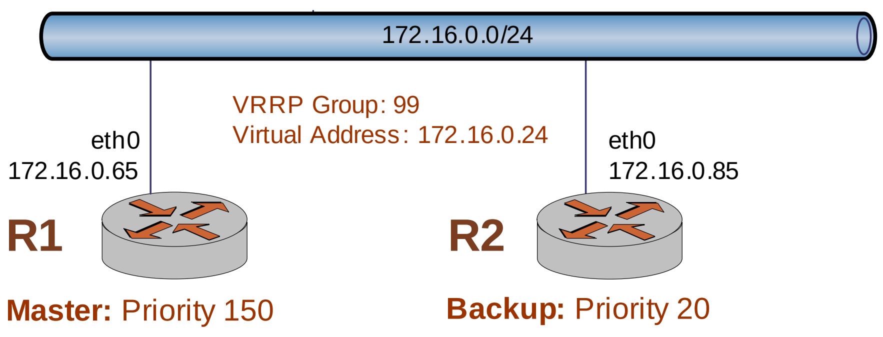
Configuring VRRP

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.

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

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

[edit]
root@R1#
```

▶ Configure the second VRRP router (R2)

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

▶ View VRRP configuration

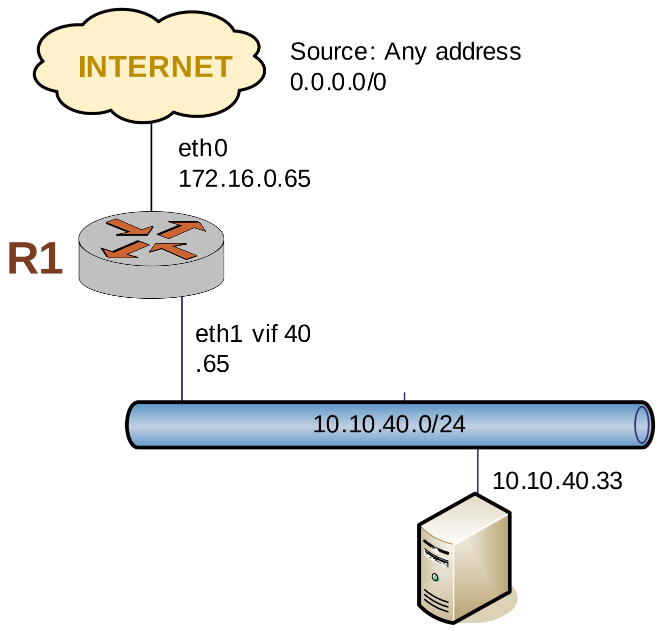
You can use the **show vrrp** command in operational mode to view VRRP configuration. This example shows VRRP configuration for router R1.

```
root@R1# quit
root@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

This sequence sets up a basic NAT configuration on router R1. When you have finished, NAT will be configured as in Figure 1-10.

Figure 1-10 Basic source NAT configuration



▶ Configure a static translation

This section sets up a NAT configuration on router R1 that permits SSH sessions to a single internal host. In effect, this configuration “exports” the private server “outside” the protected network. (Note that this means that you will not be able to access the router from outside using SSH. That is, trying to access address 172.16.0.65 will now access the server rather than the Vyatta OFR.

The external interface used is eth0, using IP address 172.16.0.65. The requests will arrive from the Internet, as defined by source network address 0.0.0.0/0 (any address). The internal host is 10.10.40.33.

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

```
root@R1# create service nat rule 1
[edit]
root@R1# edit service nat rule 1
[edit service nat rule 1]
root@R1# set type destination
[edit service nat rule 1]
root@R1# set translation-type static
[edit service nat rule 1]
root@R1# set inbound-interface eth0
[edit service nat rule 1]
root@R1# set protocols tcp
[edit service nat rule 1]
root@R1# set source network 0.0.0.0/0
[edit service nat rule 1]
root@R1# set destination address 172.16.0.65
[edit service nat rule 1]
root@R1# set destination port-name ssh
[edit service nat rule 1]
root@R1# set inside-address address 10.10.40.33
[edit service nat rule 1]
root@R1# commit
OK
[edit service nat rule 1]
root@R1# top
[edit]
root@R1# show service nat
   rule 1 {
      type: "destination"
      translation-type: "static"
      inbound-interface: "eth0"
      protocols: "tcp"
      source {
         network: 0.0.0.0/0
      }
      destination {
         address: 172.16.0.65
         port-name: ssh
      }
      inside-address {
         address: 10.10.40.33
      }
   }

[edit]
root@R1#
```

Configuring Firewall Features

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.

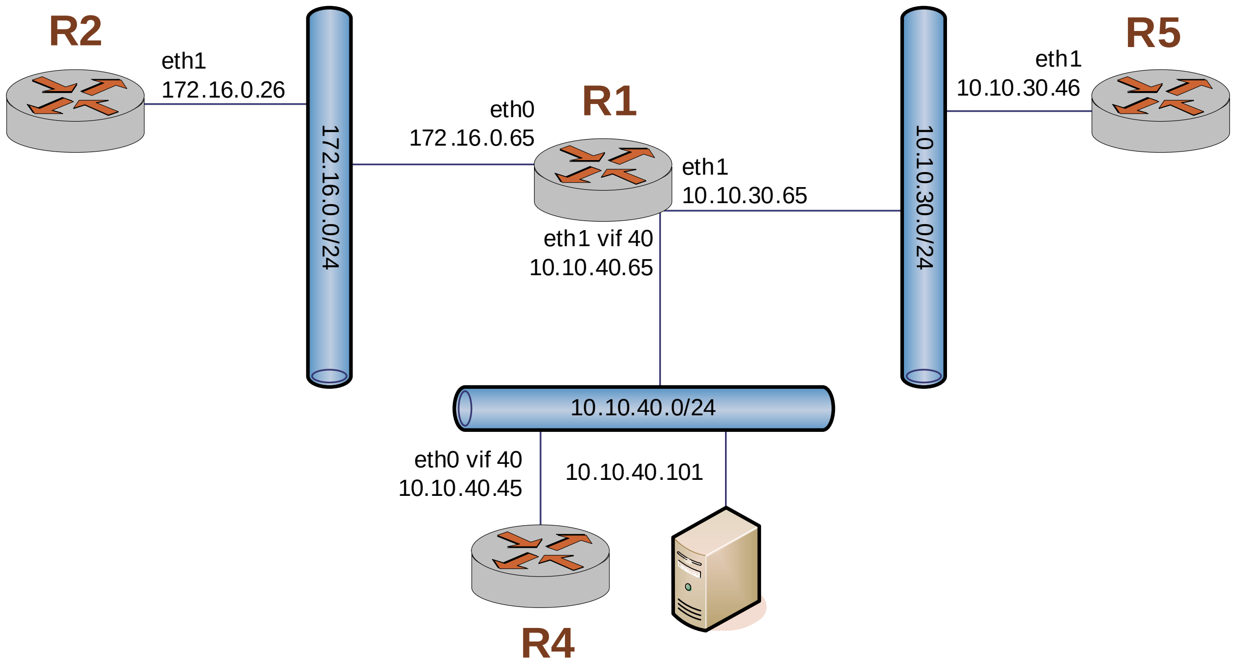
To configure firewall:

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 to 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.

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

[edit]
root@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.

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

[edit]
root@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.

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

[edit]
root@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.

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

Installing to the Hard Disk

The software image on the LiveCD includes an installation script for installing the router software onto a hard drive. This method installs the Vyatta operating system and all the routing and management software onto your hard disk or compact Flash disk.

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 256 MB of free space on your hard disk or Flash for a root partition.

1 Log on to the system as **root**.

2 At the system command prompt (not the **xorpsh** command prompt) enter the following:

```
~ # install-system
```

The installer launches.

Example 1-5 shows a sample install session.

Example 1-5 Sample install session

```
~ # 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 256 mb root
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 paritioned
your drive, you may skip this step.

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

I found the following drives on your system:
hda

Install the image on? [hda]:

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

```
How big of a root partition should I create? [256]mb:
1024

Creating filesystem on /dev/hda1: OK
Mounting /dev/hda1: OK
Copying system image files to /dev/hda1:OK
I found the following configuration files
/mnt/floppy/config/config.boot
Which one should I copy to hda?
[/mnt/floppy/config/config.boot]:

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

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

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

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
```

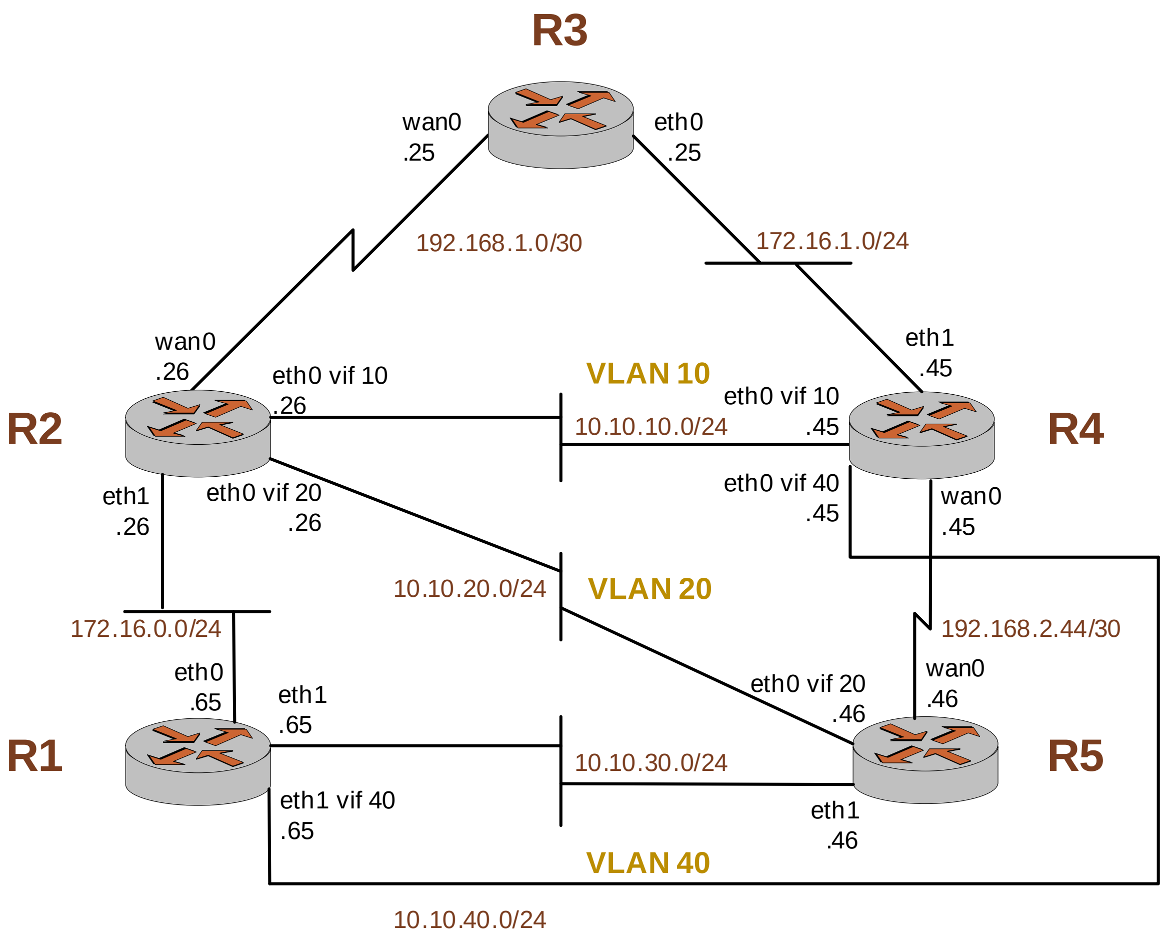
The XORP shell starts and the router command prompt displays:

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

You have successfully started the router shell.

Topology Diagram

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


R3
wan0
.25
eth0
.25
192.168.1.0/30
172.16.1.0/24
wan0
.26
eth0 vif 10
.26
VLAN 10
eth1
.45
eth0 vif 10
.45
R2
10.10.10.0/24
R4
eth1
.26
eth0 vif 20
.26
eth0 vif 40
.45
wan0
.45
10.10.20.0/24
VLAN 20
172.16.0.0/24
192.168.2.44/30
eth0
.65
eth1
.65
eth0 vif 20
.46
wan0
.46
R1
10.10.30.0/24
R5
eth1 vif 40
.65
eth1
.46
VLAN 40
10.10.40.0/24