Workshop Series

Implementing RIP2 Routers using PCs with Linux and Quagga

1 Background

A Cisco router may cost more than a small company wants to spend. It is possible to use an old Pentium computer as a fully-fledged router that implements the routing protocols RIP, OSPF and BGP, for both IPv4 and IPv6. An additional advantage is that the router can be configured with commands that are very similar to those used in Cisco's IOS, so that administrators familiar with Cisco can feel at home. Students may also use it to practice router configuration at home when they do not have access to a Cisco router of their own.

1.1 Setting Up the Router

We will use Red Hat version 8.0, although any Linux distribution would be fine, as long as the Quagga software is not too old. The locations where you can download the software are given below, but you can also buy a boxed set of Red Hat 8.0 from a number of shops in the Golden Shopping Centre.

1.2 Required Hardware

You will need hardware that meets the following requirements:

- A computer, Pentium (any speed), or above, with
- At least 64 MB RAM
- At least 2 GB hard disk
- Two network cards, preferably different brands, but two network cards of the same type will still work (with a small additional amount of testing)

Install both network cards before the installation, if possible, though (with a small additional effort), you can install the network cards later.

1.3 Determine your Network Topology

Draw a diagram like the one above, showing the connections, network interfaces, and decide on your IP addresses, netmasks, and networks.

1.4 Installing Red Hat Linux

I will not repeat details from the excellent book, *The Official Red Hat Linux x86 Installation Guide*, currently (23 January 2002) available at: `http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/install-guide/`, and packaged in convenient form in the boxed set. Please refer to this book for details. You may also ask Nick Urbanik `<nicku@vtc.edu.hk>` for any other questions you may have about installation.

If your hard disk is less than 6GB in size, please do not choose "Everything" when installing, but restrict your installation to include just one of Gnome or KDE, and some other options, proportional to the amount of disk space you have, and in relation to what you intend to use the computer for, in addition to its use as a router.

During the installation process, please create an ordinary account for yourself as well as the root account.

1.5 Installing the Quagga routing software

The software package is called `quagga-0.96.4-0.fc1.i386.rpm`. You will find it in the directory `/RedHat/RPMS/` on the third Fedora Core 1 CDROM. You can install it like this:

1. Open a terminal window with System Tools → Terminal

2. Check if the software package is installed with the command:

   ```
   $ rpm -q quagga
   ```

   If the answer is, "package quagga is not installed", then continue, otherwise it is already installed, and you can skip to the next section.

3. Open a second terminal window

4. Become `root` in that terminal with the command:

   ```
   $ su -
   ```

5. Enter your root password, and you should end up with a prompt like this: `#`. Later in this document, I will say, "at your root prompt," and I will mean, click on this window and type the command. Otherwise, you will click on your first prompt (one that is not owned by root, and has a prompt with a `$`)

6. Mount the CDROM after inserting it like this:

   ```
   # mount /dev/cdrom
   ```

7. Now change to the directory `/mnt/cdrom/RedHat/RPMS` with the command:

   ```
   # cd /mnt/cdrom/RedHat/RPMS
   ```

8. Install the software package using the rpm command:

```
# rpm -Uhv quagga-0.96.4-0.fc1.i386.rpm
```

9. Unmount the CDROM with:

```
# umount /dev/cdrom
```

10. Now remove the CDROM.

1.6 Set up Configuration Files for Quagga

The next step is to create configuration files for quagga, telling it what passwords to use: For this step, you will need to use a text editor. Linux provides a large number of choices, and it will depend on your installation what is available. The editors that may be available on your system include: `emacs`, `gedit`, `nedit`, `pico`, `gvim`, `vi`, `kate`, `xemacs` and others. Use any one of these to create two files: `/etc/quagga/quagga.conf and /etc/quagga/ripd.conf` containing the two lines:

```
password quagga
enable password quagga
```

1. At the root prompt, start an editor on each of these two files in turn, type in the content, and save each file.

1.7 Turn on the Router Services

The router software runs as a service. A service starts automatically when the computer boots, and continues to run till the computer is shut down. Here is how to set it up, at the root prompt The chkconfig program ensures that the service will start the next time the computer boots.

The `service` command can control the immediate starting and stopping of the service.

1. `# chkconfig quagga on`
2. `# chkconfig ripd on`
3. `# service quagga start`
4. `# service ripd start`

1.8 You Have Completed the Setup

All the previous steps only need to be performed once. You will not need to do this today in this workshop, since it has been done already. The remaining steps are for actually configuring the router, and this is what you will do today.

1.9 The Virtual Terminal Interface, and Differences from Cisco IOS

To configure the router, you will connect to two TCP ports on the router using the telnet program. This terminal connection is called a virtual terminal interface (VTI). A major difference from Cisco IOS is that Quagga provides a separate service for `quagga`, that manages the interfaces, and a service for each other service, such as RIP, and each has its own separate VTI. Here, we access the *quagga* interface by the command:

```
$ telnet localhost 2601
```

and give the password "quagga", which you provided in the file `/etc/quagga/quagga.conf`. To access the RIP service, we do:

```
$ telnet localhost 2602
```

You provided the passwords in the file `/etc/quagga/ripd.conf`.

2 An Example Session Setting Up the Computer in the Middle

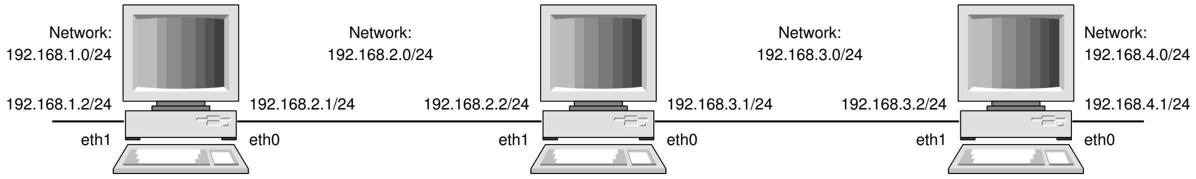

Figure 1: The arrangement with three computers.

Here I provide a session, setting up routing on the computer in the middle of the diagram in figure 1. Note that in this example session, routing has already been set up on the other two computers. I provide this without much comment; the Cisco experts here can explain what is happening better than I can. Perform all the remaining activities using your normal account; do not use the root account. Perform all this activity sitting at the computer you are setting up.

Note that the procedure for setting up the other two computers is very similar, but the IP addresses and networks will differ, according to the diagram.

2.1 Setting up the Network Interfaces with the Quagga Service on Port 2601

```
$ telnet localhost 2601
Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.


Hello, this is quagga (version 0.93a).
```

```
Copyright 1996-2002 Kunihiro Ishiguro.


User Access Verification

Password: quagga
localhost.localdomain> enable
Password: quagga
localhost.localdomain# config terminal
localhost.localdomain(config)# int eth0
! change the 3 that follows to 2 for leftmost, 4 for rightmost computer:
localhost.localdomain(config-if)# ip add 192.168.3.1/24
localhost.localdomain(config-if)# no shutdown
localhost.localdomain(config-if)# exit
localhost.localdomain(config)# int eth1
! change the 2.2 that follows to 1.2 for leftmost, 3.2 for rightmost PC:
localhost.localdomain(config-if)# ip add 192.168.2.2/24
localhost.localdomain(config-if)# no shutdown
localhost.localdomain(config-if)# exit
localhost.localdomain(config)# exit
localhost.localdomain# show running-config

Current configuration:
!
hostname localhost.localdomain
password quagga
enable password quagga
!
interface lo
!
interface eth0
 ip address 192.168.3.1/24
!
interface eth1
 ip address 192.168.2.2/24
!
!
line vty
!
end
localhost.localdomain# exit
Connection closed by foreign host.
```

3 Setting up RIP2 with the ripd Service on Port 2602

```
$ telnet localhost 2602
Trying 127.0.0.1...

Connected to localhost.
```

```
Escape character is '^]'.


Hello, this is quagga (version 0.93a).
Copyright 1996-2002 Kunihiro Ishiguro.


User Access Verification

Password: quagga
localhost.localdomain> enable
Password: quagga
localhost.localdomain# show ip rip
localhost.localdomain# configure terminal
localhost.localdomain(config)# router rip
! Following two networks differ for each computer--see the diagram
localhost.localdomain(config-router)# net 192.168.2.0/24
localhost.localdomain(config-router)# net 192.168.3.0/24
localhost.localdomain(config-rouyester)# exit
localhost.localdomain(config)# exit
localhost.localdomain# show ip rip
Codes: R - RIP, C - connected, O - OSPF, B - BGP
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

     Network            Next Hop         Metric From            Time
R(n) 192.168.1.0/24     192.168.2.1           2 192.168.2.1     02:47
C(i) 192.168.2.0/24     0.0.0.0               1 self
C(i) 192.168.3.0/24     0.0.0.0               1 self
R(n) 192.168.4.0/24     192.168.3.2           2 192.168.3.2     02:49
localhost.localdomain# exit

Connection closed by foreign host.
```

4 Viewing the routing Table using the route command:

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.4.0     192.168.3.2     255.255.255.0   UG    2      0        0 eth0
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
192.168.1.0     192.168.2.1     255.255.255.0   UG    2      0        0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
```

5 Viewing the Network Devices using the ifconfig command:

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:08:02:37:30:79
          inet addr:192.168.3.1  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17055 errors:0 dropped:0 overruns:0 frame:0
          TX packets:227 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2647631 (2.5 Mb)  TX bytes:16638 (16.2 Kb)
          Interrupt:5 Base address:0xc000

eth1      Link encap:Ethernet  HWaddr 00:60:97:08:1F:51
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2800 errors:0 dropped:0 overruns:0 carrier:4
          collisions:0 txqueuelen:100
          RX bytes:385382 (376.3 Kb)  TX bytes:267336 (261.0 Kb)
          Interrupt:10 Base address:0x1040

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:6449 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6449 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:384312 (375.3 Kb)  TX bytes:384312 (375.3 Kb)
```

6 Viewing the Addresses and Routing table using the ip command

```
$ ip address
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:08:02:37:30:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.3.1/24 brd 192.168.3.255 scope global eth0
3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:60:97:08:1f:51 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.2/24 brd 192.168.2.255 scope global eth1
$ ip route
192.168.4.0/24 via 192.168.3.2 dev eth0  proto quagga  metric 2
192.168.3.0/24 dev eth0  proto kernel  scope link  src 192.168.3.1
192.168.2.0/24 dev eth1  proto kernel  scope link  src 192.168.2.2
192.168.1.0/24 via 192.168.2.1 dev eth1  proto quagga  metric 2
```

```
127.0.0.0/8 dev lo  scope link
```