

## Routing

In this lab, you will setup static routes to route traffic between two networks, one is a small internal network and the other is a common network that provides access to the internet through an internet router. The small private network is between your two machines (we'll call this **NET11**). You will also have one of those machines act as a router, allowing communication between **NET11** and the common network **NET10**, ultimately providing internet access via an internet gateway system on **NET10**. You will work in pairs of systems; the system with only one NIC will only be on the **NET11** network, and the system with two NICs will have connections to both of the **NET10** and **NET11** networks, one network per interface, routing traffic between the **NET11** network and the common **NET10** network. Once I have enabled routing for you on the internet gateway system, your machines will have access to the internet.

Steps that start with **NET11** are to be performed only on the interface connected to the **NET11** network (i.e. both systems), and those that start with **NET10** are to be performed only on the interface connected to the **NET10** network (i.e. only the router system). **NR** refers to steps for the non-routing machine, and **R** is for the routing machine. The last sheet of this handout specifies the net topology of the lab. Turn-in one filled in handout per computer system by the beginning of class next week. Be sure to include your name (and your partner's name, if you have one).

### Configure the System

Bring the system up single-user and perform the initial network configuration.

- Step 1. Boot your system into single user mode. To accomplish this, reboot the system, and interrupt the Red Hat splash screen at the beginning of the boot process with Control-X. Then type **linux single** at the **boot:** prompt.
- Step 2. Configure your system with the IP address 10.0.0.N/24, where N is your system number. Don't forget about the files **/etc/hosts**, **/etc/hostname**, and **/etc/sysconfig/network**. Refer to Lab 1 if necessary – by now, you should be able to reliably configure your network in just a few minutes, and understand the relationship between the various configuration files and commands.
- Step 3. Bring up the loopback interface. You should now be able to ping yourself with **ping localhost**.
- Step 4. Add the entry **10.0.0.200 inet-gw** into the **/etc/hosts** table. This will allow you to refer to this machine more easily.

### Configure and bring-up the NET10 network

- Step 5. **NET10:** Configure your **eth0** interface as shown on the network topology map. Set your **host** part of your IP address to be the number on your computer.
- Step 6. **NET10:** Bring up the network (**ifup eth0**). Ping the Internet Router machine (**ping inet-gw**). This machine will be used as your **default** route later.

### Configure and bring-up the NET11 network

- Step 7. Configure the **NET11** interface as shown on the network topology map (i.e. 11.0.routercpunum.hostnum). Use the computer number of the router machine as the third octet of the IP address (*routercpunum*). By convention, routers are often host 1 on a network. **R:** use 1 as the router's host number; **NR:** use 2 as the other system's host number. Following this allows everyone in the room to know how to determine the IP addresses of any systems.
- Step 8. Bring up the appropriate interface connected to the **NET11** network. (**ifup interface**)
- Step 9. Add entries in the **/etc/hosts** table: the entry **me11** should associate your **NET11** IP address, and **other11** should associate the IP address of the other machine on **NET11**. These are relative host names which allow you to refer to the either the other machine or your own machine, regardless of which machine you are on.
- Step 10. Ping the machine connected on the other end of the **NET11** network to validate the network is operational (**ping other11**).

**Examine and understand the route table**

Step 11. Use the **route -n** or **netstat -rn** command to examine the route table. Enter your configuration below:

Destination	Gateway	Genmask	Flags	Interface

Step 12. Ping some machine *not* on your network (**NR:** ping the Internet Router machine (**inet-gw**); **R:** ping some place on the internet (cisco.com is at 198.133.219.25)). You should receive the error message "*connect: Network is unreachable*" because your system does not yet know how to reach that network. Examine your route table above to be sure you understand why.

**Setup a default route**

Step 13. To reach any other machine outside your network, there must be a route to that network. The easiest way to accomplish this is with a **default route**. The default route is the route taken if there is no other specific route to a particular network. Since one of your interfaces is connected to a network with a router, you need to specify the router's IP address as the default route. All traffic that has no other place to go will go to the routing system at that address for further routing. Set up a default route to the router on your network. **R:** your router is **inet-gw**, IP address 10.0.0.200; **NR:** your router is **other11** at IP address 11.0.routercpunum.1.

```
# route add default gw router-ip-address
```

Step 14. Examine the route table again - you should see the route you just added. Fill in the table below with the new route.

Destination	Gateway	Genmask	Flags	Interface

Step 15. Once again, ping some machine *not* on your network (**NR:** ping the **inet-gw**; **R:** ping some place on the internet (cisco.com is at 198.133.219.25)).

**Q1.** Does the **ping** succeed? Why? \_\_\_\_\_

**Q2.** If not, what is different this time compared with the previous **ping**? \_\_\_\_\_

Step 16. **NR:** Let the **ping** run continuously - if you killed it, re-run it again and let it run. Wait until the router administrator tells you to proceed.

**Setting up the router**

Step 17. **NET10:** You have routes now to both the **NET10** and **NET11** networks. And host **other11** has a default route with your system acting as the router. However, their network traffic is not being forwarded! This is a security measure in Linux. Although the routes may be setup properly, the system will not forward traffic by default. You need to tell the kernel to allow packets received on one interface to be forwarded through another interface. This is called **IP Forwarding**. Enabling **IP Forwarding** requires setting a kernel value. Using the **proc** filesystem interface is the easiest way. First, validate for yourself that your system is not set to forward IP traffic:

```
# cd /proc/sys/net/ipv4
# cat ip_forward
```

**Q3. NET10:** What is the **ip\_forward** value set to? \_\_\_\_\_

Step 18. **NET10:** Now enable IP Forwarding by changing the value to 1.

```
# echo 1 > ip_forward
```

Step 19. **NET10:** You will find that the **ping** from **other11** is still not getting a response. Although traffic *is* being forwarded by your station, the router **inet-gw** does not know about your private **NET11** network. It too needs a route to your **NET11** network to allow the ICMP response (from ping) to be returned. Log into the Internet Router and examine the routing table on this machine (Username: guest; see the board for the password). Fill in the table below (exclude any **NET11** routes not your own).

Destination	Gateway	Genmask	Flags	Interface

**Q4. NET10:** Can the Internet Gateway machine access your **NET11** network? \_\_\_\_\_

**Q5. NET10:** How do you prove this? \_\_\_\_\_

**Q6.** What is the IP address of the gateway used to get your traffic out to the internet? \_\_\_\_\_

Step 20. Bribe the **inet-gw** administrator to add a route to your **NET11** network via your **NET10** interface (have your **NET11** network number and your **NET10** IP address ready). The command will be:

```
# route add -net 11.0.routercpunum.0/24 gw 10.0.0.routercpunum
```

Step 21. **NET10:** Once the route to your **NET11** network is established on **inet-gw**, ask the now-very-impatient administrators of **other11** if their traffic is now reaching **inet-gw**. Tell them it is OK to proceed now!

**Validate that you have internet access**

Step 22. Bring the system up multi-user level 3.

```
# init 3
```

Step 23. Re-enable IP forwarding by placing a 1 in **/proc/sys/net/ipv4/ip\_forward** (it was disabled when the system change init levels).

Step 24. To enable your system to use host and domain names, you need to tell the system the IP address of a domain name server. Edit the file **/etc/resolv.conf** and add the line **nameserver 153.18.8.1** to the file. Remove any other entries in the file. Try pinging some internet host by name rather than by IP address.

Step 25. Start an X session using **startx** and then start a browser to surf the web. Try **telnet** to connect to one of the UNIX systems at foothill (taipei.fhda.edu, losaltos.fhda.edu, kyoto.fhda.edu, etc.).

## Gain experience with the network debugging and statistics tools

Step 26. The **netstat** command has many uses. The **-i** option shows packet statistics and other diagnostics over the various network interfaces. There are statistics for successful transmits and receives (TX-OK, RX-OK), and for some of the other errors that could have occurred during packet delivery (general errors, dropped frames, overruns).

**Q7.** How many packets have been transmitted over each interface? \_\_\_\_\_

**Q8.** How many packets have been received over each interface? \_\_\_\_\_

Step 27. Run **netstat -i** a few more times while there is network activity and look for changes in the packet statistics.

Step 28. The **-s** option of **netstat** shows more detailed statistics.

**Q9.** How many ping messages were received? \_\_\_\_\_

**Q10.** How many were replied to? \_\_\_\_\_

**Q11. NET10:** How many packets were forwarded? \_\_\_\_\_

Step 29. Initiate a telnet session to one of the Foothill UNIX machines (ex. losaltos.fhda.edu, saigon.fhda.edu). In another terminal window run **netstat -s**.

**Q12.** How many active TCP connections exist? \_\_\_\_\_

Step 30. Examine the ARP table with the **arp** command. Be sure you understand each entry.

## Add routes to other NET11 networks

Step 31. **NET10:** Currently, neither machine can reach other machines on the other 11.0.x.0/24 networks. Set up a specific network route to reach hosts on some other 11.0.x.0/24 network. The command below adds a route to the network 11.0.N.0/24 via the router at 10.0.0.N (which is of course on the NET10 network).

```
# route add -net 11.0.N.0/24 gw 10.0.0.N
```

Step 32. Attempt to **ping** the non-router machine on the network for which you just created a route. Use the map to determine the IP address of that machine.

**Q13.** Did the ping succeed? If so, how did the reply get back to you? \_\_\_\_\_

\_\_\_\_\_

## Explore ICMP Redirects

Step 33. Check to see if your system is accepting ICMP redirects. Again, you can use the /proc filesystem interface. Use the command below

```
# grep '[01]' /proc/sys/net/ipv4/conf/*/accept_redirects
```

**Q14.** Which interfaces are accepting redirects? \_\_\_\_\_

Step 34. **R:** You are now going to force an ICMP Redirect route into your route table, by removing the default route to the Internet Router, and adding a default route to another routing system on the **NET10** network. Remove the default network route with the **route** command and use **netstat** to verify the route is gone. Also, ping some place on the internet (cisco.com is at 198.133.219.25) to be sure that you cannot reach that network. Issue the commands shown:

```
# route del default
# netstat -rn
# ping 198.133.219.25
```

Step 35. **R:** Add a new default route to another router on the **NET10** network (be sure they have their default route still).

```
# route add default gw any-NET10-router-address
```

Step 36. Examine your routing table. Now ping the previous IP address again and look carefully at the output of the ping.

**Q15.** What is different with the ping output? \_\_\_\_\_

Step 37. Examine your route table again.

**Q16.** Do you see any new routes added by an ICMP Redirect? \_\_\_\_\_

Step 38. Disable acceptance of ICMP Redirects

```
# cd /proc/sys/net/ipv4/conf
# for i in */accept_redirects ; do echo 0 > $i ; done
```

Step 39. Now ping an internet host again and re-examine your route table. You should find that you can still ping the host. When acting as a gateway, Linux systems do not allow ICMP redirects to update the routing table. This is a security measure preventing your routing system from being hijacked, forcing traffic to flow through undesired networks. On the other hand, hosts may have their tables updated.

**Adding a route to reject traffic**

Step 40. **NET10:** Routing table entries can be added to the routing table that will specifically reject traffic. Add a reject route to your routing table, which will prevent **NET11** traffic.

```
# route add -net 11.0.routercpunum.0/24 reject
```

Step 41. **NET10:** Examine the route table. Fill in the table below with the reject route, specifically noting the Flags:

Destination	Gateway	Genmask	Flags	Interface

Step 42. **NR:** Try to ping the internet gateway machine **inet-gw**. You should find that traffic is blocked by the router machine. Let the ping continue to run.

Step 43. **NET10:** Remove the reject route, using the keyword **del** instead of **add** from the above command. You should see that the pings on the NR machine succeed

Congratulations! You are now a routing expert.

