LAB THREE

Static Routing

OBJECTIVES

- How to turn a computer with multiple interfaces into a router
- How to set up static routing on Linux PC routers and Cisco routers
- How ICMP messages update routing table entries
- How Proxy ARP helps to connect different networks without reconfiguring the hosts
- How to work with different network masks

CONTENTS

You only need to do Part 1 to Part 3

QUESTION SHEET FOR PRELAB 3

Optional: Not Graded

Answer the questions in the space provided below each one. Use extra sheets of paper if needed and attach them to this document. Submit the answers to the question sheet with your lab report.

Name (please print):____

1. What is the IOS command to change the MTU (Maximum Transmission Unit) for an interface on a Cisco router?

2. How does a router determine whether datagrams to a particular host can be directly delivered through one of its interfaces?

3. Which systems generate ICMP route redirect messages—routers, hosts, or both?

4. What is the default maximum TTL value used by `traceroute` when sending UDP datagrams?

5. Describe the role of a *default gateway* in a routing table.

6. What is the network prefix of IP address 192.110.50.3/24?

7. Explain the difference between a network IP address and a network prefix.

8. An organization has been assigned the network number 140.25.0.0/16 and it needs to create networks that support up to 60 hosts on each IP network. What is the maximum number of networks that can be set up? Explain your answer.

LAB 3

In this lab you work with four different network topologies. The topology for Parts 1–4 is shown in Figure 3.1. These parts address router configuration on a Linux PC and a Cisco router. The topology for Part 5 is shown in Figure 3.2. This topology is used to study the role of *ICMP route redirect* message. For Part 6 we add one more router to the topology of Part 5 and examine the effect of routing loops. The topology for Part 7 is shown in Figure 3.4. There, you explore the relationship between network prefixes and IP forwarding.

RECALL:

- Before you get started, please reboot the Linux PCs.
- During the lab, you need to save data to files. Save all files in the directory */labdata*.
- Save your files to a floppy disk before the end of the lab. You will need the files when you prepare your lab report.

PART 1. CONFIGURING A LINUX PC AS AN IP ROUTER

Every Linux PC with at least two network interfaces can be set up as an IP router. Configuring a Linux PC as an IP router involves two steps: (1) modifying the configuration of Linux, so that IP forwarding is enabled, and (2) configuring the routing table. Figure 3.1 shows the network topology used in Parts 1–4 of this lab. PC1 and PC4 are used as hosts, and PC2 and Router1 are set up as IP routers. The PCs and the Cisco router are connected by three Ethernet hubs. In Lab 3, all routing table entries are manually configured, which is known as static routing.

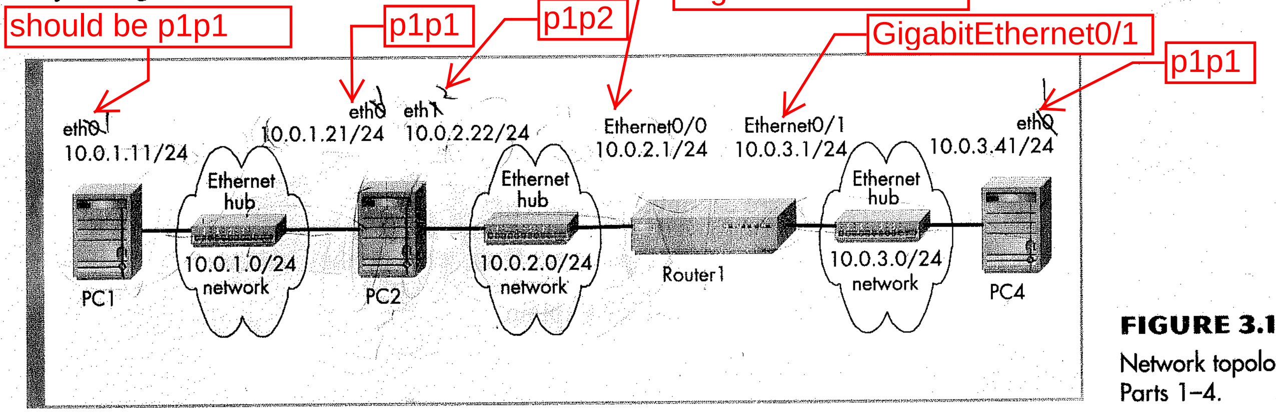

FIGURE 3.1. Network topology for Parts 1–4.

You need to save this and compress all outputs to one file, when you finish your lab. You need to submit it by uploading the file to gateway. It's optional and will not be graded, but it proves you are innocent when we detect your lab report is similar with others To upload, run:
scp filename submit@gateway:~/

TABLE 3.1. IP addresses for Parts 1–4.

p1p1

p1p2

Linux PC	Ethernet Interface *eth0* 1	Ethernet Interface *eth1* 2
PC1	`10.0.1.11/24`	Disabled
PC2	`10.0.1.21/24`	`10.0.2.22/24`
PC4	`10.0.3.41/24`	Disabled
Cisco Router	**Ethernet Interface *Ethernet0/0***	**Ethernet Interface *Ethernet0/1***
Router1	`10.0.2.1/24`	`10.0.3.1/24`

GigabitEthernet0/1

GigabitEthernet0/1

Note: Table 3.1 assumes that there is an Ethernet interface card in slot 0 of the Cisco router. If the Ethernet is in a different slot, say slot 1, the name has to be changed (e.g., *Ethernet1/0* and *Ethernet1/1*). If the Ethernet interface card, assumed to be in slot 1, is a 100 Mbps interface card, the names are *FastEthernet1/0* and *FastEthernet1/1*. On a router that does not have slotted interfaces (e.g., Cisco 2500 class routers), the names of the Ethernet interfaces are *Ethernet0* and *Ethernet1*.

EXERCISE 1(A).
Network setup.

1. Connect the Ethernet interfaces of the Linux PCs and the Cisco router as shown in Figure 3.1. Configure the IP addresses of the interfaces as given in Table 3.1.

Please note that you dont need to config the IP address of the router at this step

2. Start to capture traffic on PC1 with *ethereal*.
3. Issue a `ping` command from PC1 to PC2, Router1, and PC4. Save the output of each `ping` command.

```
PC1% ping -c 5 10.0.1.21
PC1% ping -c 5 10.0.2.1
PC1% ping -c 5 10.0.3.41
```

some IPs are not reachable

4. Save the captured *ethereal* output.

Lab Report Use the saved data to answer the following questions:

- What is the output on PC1 when the `ping` commands are issued?
- Which packets, if any, are captured by *ethereal*?
- Do you observe any ARP or ICMP packets? If so, what do they indicate?
- Which destinations are not reachable? Explain.

You need to answer the questions in lab report. You lab score will be graded base on the quality of lab report.

EXERCISE 1(B).
Configuring a Linux PC as an IP router.

On a Linux system, IP forwarding is enabled when the file */proc/sys/net/ipv4/ip_forward* contains a 1 and disabled when it contains a 0. You can enable IP forwarding by writing a 1 in the file, with the command

```
PC1% echo "1" > /proc/sys/net/ipv4/ip_forward
```

The command `echo` writes the given argument, here, the string `"1"`, to the standard output. Using the redirect operator (>) and a filename, the output of the command is written to a file. IP forwarding is disabled with the command

```
PC1% echo "0" > /proc/sys/net/ipv4/ip_forward
```

The command has an immediate effect; however, changes are not permanent and are lost when the system is rebooted. Modifying the IP forwarding state permanently requires changes to the configuration file */etc/sysctl.conf.* IP forwarding is enabled if the file contains a line *net.ipv4.ip_forward = 1*, and IP forwarding is disabled when the line does not exist or the file contains the line *net.ipv4.ip_forward = 0.*[1] Changes to the configuration file */etc/sysctl.conf* take effect the next time Linux is rebooted.

- Enable PC2 as an IP router using the command

  ```
  PC2% echo "1" > /proc/sys/net/ipv4/ip_forward
  ```

EXERCISE 1(C).
Setting static routing table entries for a Linux PC.

Next, you must set up the routing tables of the Linux PCs. PC1 and PC4 are hosts, and PC2 is an IP router. The routing tables are configured so that they conform to the network topology shown in Figure 3.1 and Table 3.1. The routes are configured manually, which is also referred to as *static routing*.

Configuring static routes in Linux is done with the command `route`, which has numerous options for viewing, adding, deleting or modifying routing entries. The various uses of the `route` command are summarized in the list.

```
route add –net netaddress netmask mask gw gw_address
route add –net netaddress netmask mask dev iface
```

Adds a routing table entry for the network prefix identified by IP address *`netaddress`* and netmask *`mask`*. The next-hop is identified by IP address *`gw_address`* or by interface *`iface`*.

```
route add –host hostaddress gw gw_address
route add –host hostaddress dev iface
```

Adds a host route entry for IP address *`hostaddress`* with the next-hop identified by IP address *`gw_address`* or by interface *`iface`*.

```
route add default gw gw_address
```

Sets the default route to IP address *`gw_address`*.

```
route del –net netaddress netmask mask gw gw_address
route del –host hostaddress gw gw_address
route del default gw gw_address
```

Deletes an existing route from the routing table. It is not necessary to type all arguments. If enough arguments are provided that it can be matched with an existing routing entry, the first entry that matches the given arguments is deleted.

[1] Yet another method to configure a Linux as a router is to add a line *FORWARD_IPV4=true* to the file */etc/sysconfig/network.* IP forwarding is disabled by setting *FORWARD_IPV4=false.*

`route –e`

Displays the current routing table with extended fields. The command is identical to the `netstat –r` command.

`route –C`

Displays the routing table cache.

The command for adding a route for the network prefix 10.21.0.0/16 with next-hop address 10.11.1.4 is

```
PC1%route add -net 10.21.0.0 netmask 255.255.0.0 gw 10.11.1.4
```

The command to add a host route to IP address 10.0.2.31 with the next-hop set to 10.0.1.21 is

```
PC1%route add -host 10.0.2.31 gw 10.0.1.21
```

The command to add the IP address 10.0.4.4 as the default gateway is done with the command

```
PC1%route add default gw 10.0.4.4
```

The commands to delete the entries created with the previous commands are

```
PC1%route del -net 10.21.0.0 netmask 255.255.0.0
PC1%route del -host 10.0.2.31
PC1%route del default
```

In Linux, there is no simple way to delete all entries in the routing table. When the commands are issued interactively in a Linux shell, the added entries are valid until Linux is rebooted. To make static routes permanent, the routes need to be entered in the configuration file */etc/sysconfig/static-routes*, which is read each time Linux is started.

The listed commands are helpful to get information on routing and to find mistakes in the routing setup.

`ping` *`IPaddress`*

Tests whether *`IPaddress`* can be reached

`traceroute` *`IPaddress`*

Displays the route to the interface *`IPaddress`*

1. Configure the routing table entries of PC1 and PC4. You can either specify a default route or insert separate routing entries for each remote network. For this exercise, add a route for each individual remote network. As a hint, here is the configuration information for PC4:

```
PC4%route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.3.1
PC4%route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.0.3.1
```

2. Configure the routing table entries of the IP router PC2. (The correctness of the routing entries will be tested after Router1 has been set up.)

3. Display the routing table of PC1, PC2, and PC4 with `netstat -rn` and save the output.

Lab Report Include the saved output of the routing table. Explain the entries in the routing table and discuss the values of the fields for each entry.

PART 2. CONFIGURING A CISCO ROUTER

The setup of a Cisco router is more involved. The first step is to establish a physical connection to the router, so that configuration commands can be entered. There are different ways to connect to a Cisco router. In the Internet Lab, you establish a serial connection to the router. This is done with a serial cable that connects the serial port of a Linux PC to the console port of a Cisco router. The next step is to run a terminal emulation program on the Linux PC. In the Internet Lab, you use the *kermit* communication software to access the router. Lastly, you have to type IOS commands to configure the Cisco router. Refer to Section 4 in the Introduction for a detailed discussion on how to navigate and work with the IOS.

The network setup for this part is as shown in Figure 3.1 and Table 3.1.

EXERCISE 2(A).
Accessing a Cisco router via the console port with *kermit*.

To access a Cisco router from one of the Linux PCs, connect one of the serial ports of the PC to the console port of the Cisco router via a serial cable. Then, you can use the `kermit` command to establish a remote terminal connection to the router. You will use Router1 as the IP router and PC1 as the console.

The following steps access the console port of Router1 from PC1:

You can skip step 1 to 5. We already set those connection parameters. Just type "minicom" and hit enter twice

1. Use a serial cable to connect the serial port of PC1 to the console port of Router1. The serial port is labeled *ttyS0* or *ttyS1*.
2. Start *kermit* by typing

   ```
   PC1% kermit
   ```

   This brings up the prompt:

   ```
   [/root]C-kermit>
   ```

3. Use the `set line` command to select the `ttyS0` serial port.

   ```
   [/root]C-kermit> set line /dev/ttyS0 (or /dev/ttyS1)
   ```

4. Use the `set carrier-watch` command to disable the requirement for a carrier detect signal

   ```
   [/root]C-kermit> set carrier-watch off
   ```

5. Connect to the router by issuing the following command:

   ```
   [/root]C-kermit> connect
   ```

If the connection is successful, you see a command prompt (user EXEC prompt) from Router1:

```
Router1>
```

When you see this prompt, you can type Cisco IOS commands. If the prompt does not appear, then press the Enter key several times.

NOTE:

To terminate a *kermit* session, type Ctrl-\ (control-backslash) and then press the C key or type `c`. *kermit* takes a few seconds (maybe 10) to exit.

EXERCISE 2(B).
Switching Cisco IOS command modes.

This exercise demonstrates how to log into a router and how to navigate the different Cisco IOS command modes. It is important to understand the different modes so you know where you are and what commands are accepted at any time.

1. Make sure that PC1 is connected to Router1 via a serial cable and that a *kermit* session is started.

2. When PC1 is connected to the router, you see the prompt of the User EXEC mode (`Router>`). To see which commands are available in this mode, type a question mark (`?`):

```
Router1> ?
```

3. To view and change system parameters of a Cisco router, you must enter the Privileged EXEC mode, by typing

```
Router1> enable
Password : <enable secret>
Router1#
```

   You need a password, the *`enable secret`*, to enter the Privileged EXEC mode.

4. To modify systemwide configuration parameters, you must enter the global configuration mode. This mode is entered by typing

```
Router1# configure terminal
Router1(config)#
```

5. To make changes to a network interface, enter the interface configuration mode, with the command

```
Router1(config)# interface Ethernet0/0
Router1(config-if)#
```

GigabitEthernet0/0

   The name of the interface is provided as an argument. Here, the network interface that is configured is *Ethernet0/0*.

GigabitEthernet0/0

6. To return from the interface configuration to the global configuration mode or from the global configuration mode to the Privileged EXEC mode, use the `exit` command:

```
Router1(config-if)# exit
Router1(config)# exit
Router1#
```

The `exit` command takes you one step up in the command hierarchy. To directly return to the Privileged EXEC mode from any *configuration* mode, use the `end` command:

```
Router1(config-if)# end
Router1#
```

7. To return from the Privileged EXEC mode to the User EXEC mode, type

```
Router1# disable
Router1>
```

8. To terminate the console session from the User EXEC mode, type

```
Router1> logout
Router1 con0 is now available
Press RETURN to get started.
```

Or type `logout` or `exit` from the Privileged EXEC mode.

EXERCISE 2(C).
Configuring IP interfaces on a Cisco router.

The following exercises use basic commands from IOS that are needed to configure a Cisco router. Refer to Section 4 of the Introduction for detailed explanations.

1. Connect PC1 to Router1 via the serial cable and start a *kermit* session.

2. Configure Router1 with the IP addresses given in Table 3.1.

```
Router1> enable
Password: <enable secret>
Router1# configure terminal
Router1(config)# no ip routing
Router1(config)# ip routing
Router1(config)# interface Ethernet0/0      GigabitEthernet0/0
Router1(config-if)# ip address 10.0.2.1 255.255.255.0
Router1(config-if)# no shutdown
Router1(config-if)# interface Ethernet0/1      GigabitEthernet0/1
Router1(config-if)# ip address 10.0.3.1 255.255.255.0
Router1(config-if)# no shutdown
Router1(config-if)# end
```

3. When you are done, use the following command to check the changes you made to the router configuration and save the output:

```
Router1# show interfaces
Router1# show running-config
```

4. Analyze the output to ensure that you have configured the router correctly.

Lab Report Include the output from Step 3 in your lab report.

EXERCISE 2(D).
Setting static routing table entries on a Cisco router.

Next you must add static routes to the routing table of Router1. The routing table must be configured so that it conforms to the network topology shown in Figure 3.1 and Table 3.1.

The IOS command to configure static routing is `ip route`. The command can be used to show, clear, add, or delete entries in the routing table. The commands are summarized in the list.

IOS MODE: PRIVILEGED EXEC

```
show ip route
```

Displays the contents of the routing table

```
clear ip route *
```

Deletes all routing table entries

```
show ip cache
```

Displays the routing cache

IOS MODE: GLOBAL CONFIGURATION

```
ip route-cache
```

Enables route caching. By default, route caching is enabled on a router.

```
no ip route-cache
```

Disables route caching.

```
ip route destination mask gw_address
```

Adds a static routing table entry to *`destination`* with `netmask` *`mask`*. The argument *`gw_address`* is the IP address of the next-hop router.

```
ip route destination mask Iface
```

Adds a static routing table entry to *`destination`* with netmask *`mask`*. Here, the next-hop information is the name of a network interface (e.g., *FastEthernet0/0*).

```
no ip route destination mask gw_address
no ip route destination mask Iface
```

Deletes the route table entry with *`destination`*, *`mask`*, and *`gw_address`* or *`Iface`* from the routing table.

We next show some examples for adding and deleting routing table entries in IOS. Compare these commands to the corresponding Linux commands in Part 1, Exercise

1(C). As in Linux, whenever an IP address is configured for a network interface, routing table entries for the directly connected network are added automatically.

The command for adding a route for the network prefix 10.21.0.0/16 with 10.11.1.4 as the next-hop address is

```
Router1(config)#ip route 10.21.0.0 255.255.0.0 10.11.1.4
```

The command to add a host route to IP address 10.0.2.31 with the next-hop set to 10.0.1.21 is

```
Router1(config)#ip route 10.0.2.31 255.255.255.255 10.0.1.21
```

In IOS, a host route is identified by a 32 bit prefix.

The command to add the IP address 10.0.4.4 as the default gateway is done with the command

```
Router1(config) #ip route 0.0.0.0 0.0.0.0 10.0.4.4
```

Finally, commands to delete the previous entries use the `no ip route` command.

```
Router1(config)# no ip route 10.21.0.0 255.255.0.0 10.11.1.4
Router1(config)# no ip route 10.0.2.31 255.255.255.255 10.0.1.21
Router1(config)# no ip route 0.0.0.0 0.0.0.0 10.0.4.4
```

1. Display the content of the routing table with `show ip route`. Note the routing entries that are already present. Save the output.
2. Add routing entries to Router1 so that the router forwards datagrams for the configuration shown in Figure 3.1. Routing entries should exist for the following networks:
    - 10.0.1.0/24
    - 10.0.2.0/24
    - 10.0.3.0/24
3. Display the routing table again with `show ip route` and save the output.

Lab Report Include the saved output of the routing table from Steps 1 and 2. Explain the fields of the routing table entries of the Cisco router. Explain how the routing table has changed from Step 1 to Step 3.

PART 3. FINALIZING AND EXPLORING THE ROUTER CONFIGURATION

If the configuration of PC2 and Router1 was done correctly, it is now possible to send IP datagrams between any two machines in the network shown in Figure 3.1. However, if the network is not configured properly, you need to debug and test your setup. Table 3.2 illustrates several common problems that may arise. Since it is impossible to cover all scenarios, network debugging is a crucial skill that you need to obtain for your lab experiments to work well.

The network setup for this part is as shown in Figure 3.1 and Table 3.1.

TABLE 3.2. Troubleshooting network configurations.

Problem	Possible Causes	Debugging
Traffic does not reach destinations on local network.	Network interface not configured correctly.	Verify the interface configuration with `show protocols` (in IOS) or `ifconfig` (in Linux).
	Incorrectly connected, faulty or loose cables.	Most interface cards and Ethernet hubs have green LED status lights. Check whether the status lights are on. Verify the connection of the cables. Verify that no crossover cables are used.
Traffic reaches router but is not forwarded to remote networks.	IP forwarding is not enabled.	Use `show protocols` (in IOS) or look into */proc/sys/net/ipv4/ip_forward* (in Linux) to display the forwarding status.
	Routing tables are not configured correctly.	Display routing tables with `show ip route` (in IOS) or `netstat -rn` (in Linux). Run `traceroute` between all hosts and routers.
ICMP Request message reaches destination, but ICMP Reply does not reach source.	Routing tables are not correctly configured for the reverse path.	Display routing tables with `show ip route` (in IOS) or `netstat -rn` (in Linux). Run `ping` and `traceroute` in both directions.
A change in the routing table has no effect on the flow of traffic.	The ARP cache has old entries.	Delete the ARP cache with `clear arp` (in IOS) or delete entries with `arp -d` (in Linux).

EXERCISE 3(A).
Finalizing the network setup.

Continue with the network configuration from Part 2.

Test the network configuration by issuing `ping` commands from each host and router to every other host and router. If some `ping` commands do not work, you need to modify the configuration of routers and hosts. If all `ping` commands are successful, the network configuration is correct, and you can proceed to the next step.

EXERCISE 3(B).
Testing routes with traceroute.

1. Start an *ethereal* session on PC1.
2. Execute a `traceroute` command from PC1 to PC4 and save the output.

   ```
   PC1% traceroute 10.0.3.41
   ```

   Observe how `traceroute` gathers information on the route.
3. Stop the traffic capture of *ethereal* and save the traffic generated by the `traceroute` command.
4. Save the routing table of PC1, PC4, PC2, and Router1.

Lab Report Use the *ethereal* output and the previously saved routing table to explain the operation of `traceroute`.

EXERCISE 3(C).
Observe MAC addresses at a router.

When a router forwards an IP datagram from one Ethernet segment to another, it does not modify the IP destination address. However, the destination Ethernet address in the Ethernet header is modified at a router.

This exercise requires manipulations to the ARP cache. The `arp` command in Linux was covered in Lab 2. The list shows corresponding IOS commands for Cisco routers.

IOS MODE: PRIVILEGED EXEC

`show ip arp`

Displays the contents of the ARP cache

`clear arp`

Deletes the entire ARP cache

IOS MODE: GLOBAL CONFIGURATION

`arp` *`IPaddress`*

Adds an entry for *`IPaddress`* to the ARP cache

`no arp` *`IPaddress`*

Deletes the ARP entry for *`IPaddress`* from the ARP cache

1. Erase all ARP entries on PC1, PC2, PC4, and Router1.
2. Run *ethereal* on both PC1 (interface *eth0*) and PC4 (interface *eth0*).
3. Issue a `ping` command on PC1 to PC4.

   ```
   PC1% ping -c 5 10.0.3.41
   ```

4. Save the packet transmissions triggered by the `ping` command, including ARP requests, ARP Reply, ICMP Echo Request, and ICMP Echo Reply on both PC1 and PC4.

Lab Report

- Determine the source and destination addresses in the Ethernet and IP headers for the ICMP Echo Request messages that were captured at PC1.
- Determine the source and destination addresses in the Ethernet and IP headers for the ICMP Echo Request messages that were captured at PC4.
- Use your previous answers to explain how the source and destination Ethernet and IP addresses are changed when a datagram is forwarded by a router.

EXERCISE 3(D).
Multiple matches in the routing table.

A router or host uses a routing table to determine the next hop of the path of an IP datagram. In Linux, routing table entries are sorted in the order of decreasing prefix length and are read from top to bottom. In this exercise, you determine how an IP router or Linux PC resolves multiple matching entries in a routing table.

1. Add the following routes to the routing table of PC1:

   ```
   PC1% route add -net 10.0.0.0 netmask 255.255.0.0 gw 10.0.1.71
   PC1% route add -host 10.0.3.9 gw 10.0.1.81
   ```

   From Exercise 1(C) there should be a network route for the network prefix 10.0.3.0/24. If there is no such route, then add the following entry:

   ```
   PC1% route add -net 10.0.3.0 netmask 255.255.255.0
   gw 10.0.1.61
   ```

2. Referring to the routing table, determine how many matches exist for the following IP addresses:

   ```
   10.0.3.9
   10.0.3.14
   10.0.4.1
   ```

3. Start an *ethereal* session on PC1 and issue the following `ping` commands from PC1:

   ```
   PC1% ping -c 1 10.0.3.9
   PC1% ping -c 1 10.0.3.14
   PC1% ping -c 1 10.0.4.1
   ```

Note that gateways with IP addresses 10.0.1.61, 10.0.1.71, and 10.0.1.81 do not exist. However, PC1 still sends ARP Request packets for these IP addresses.

4. Save the output of *ethereal* and PC1's routing table.

Lab Report Use the saved output to indicate the number of matches for each of the preceding IP addresses. Explain how PC1 resolves multiple matches in the routing table. Include only relevant output data in your report to support your analysis of the data.

EXERCISE 3(E). Default Routes.

1. Delete the routing table entries added in Step 1 of Exercise 3(D). (Otherwise, the entries interfere with the remaining exercises in this lab.)
2. Add default routes on PC1 and PC2.
   a. On PC1, add a default route with interface *eth0* of PC2 as the default gateway.
   b. On PC2, add a default route with interface *Ethernet0/0* of Router1 as the default gateway.
3. Start to capture traffic on PC1 (on *eth0*) and PC2 (on both *eth0* and *eth1*) with *ethereal.*
4. Issue a `ping` command from PC1 to a host on a network that does not exist.

```
PC1% ping -c 5 10.0.10.110
```

5. Save the *ethereal* output.

Lab Report Use your saved data to answer the following questions:

- What is the output on PC1 when the `ping` command is issued?
- Determine how far the ICMP Echo Request message travels?
- Which, if any, ICMP Echo Reply message returns to PC1?

PART 4. PROXY ARP

~~Proxy Address Resolution Protocol (Proxy ARP) is a method by which a router can forward traffic without using its routing table. Proxy ARP is a configuration option when an IP router responds to ARP Requests that arrive from one of its connected networks for a host that is on another of its connected networks. Without Proxy ARP enabled, an ARP Request for a host on a different network would be unsuccessful, since routers do not forward ARP packets to another network.~~

~~In this part you explore how Proxy ARP enables routers to forward an IP datagram even though the sender of the datagram is not aware that the IP datagram should be forwarded to a router. Continue with the network configuration from Figure 3.1, and with IP addresses as shown in Table 3.1.~~

~~The commands to enable and disable Proxy ARP in IOS are listed.~~