Based on
Mastering Networks - An Internet Lab Manual
by Jörg Liebeherr and Magda Al Zarki

Adapted for
'Labo Computernetwerken'
by Johan Bergs, Nicolas Letor, Michael Voorhaen and Kurt Smolderen

Completed by
Josse Coen Armin Halilovic Jonas Vanden Branden Group 2

March 11, 2016

Lab 3

Static Routing

What you will learn in this lab:

- How to turn a computer with multiple interfaces into a router
- How to set up static routing on Linux PC-routers and Cisco commercial 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 prefixes

3.1 Prelab 3

Network Commands in Linux

Read the manual pages of the following commands at http://manpages.ubuntu.com/ for the operating system version “trusty 14.04 LTS”:

- `route`
- `traceroute`
- `minicom`: This lab uses the `minicom` utility program to establish a serial connection between a Linux PC and a Cisco router.

Proxy ARP

Go to the website of Cisco at http://goo.gl/ixuktT and read about Proxy ARP.

Cisco IOS

The Cisco routers in the Lab are running a recent version of the Cisco Internet Operating System (IOS). Read about the IOS at http://goo.gl/UD23vX. Note that this is reference material that you can use. You are not expected to go through all of the manuals listed here!


The most useful manuals for this course are the “IP Application Services Configuration Guide” and “Cisco IOS IP Switching Configuration Guide”.

Prelab Questions

Question 1)

What is the IOS command to change the MTU (maximum transmission unit) for an interface on a Cisco router?
ip mtu <bytes> With a minimum of 128 bytes.

Question 2)

How does a router determine whether a datagrams to particular host can be directly delivered through one of its interfaces?
The datagrams are ip packets with the destination ip address in its header. The router determines this destination address and checks it with entries in its routing table. If its present, the corresponding interface can be read from this table.

Question 3)

Which systems generate ICMP Route Redirect messages? Routers, hosts, or both?
ICMP redirect messages are used by routers to notify the hosts on the data link that a better route is available for a particular destination. So we can state that these messages are sent only by routers.

Question 4)

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

When running 'traceroute -U 8.8.8.8' we see that there are '30 hops max'. From the way that traceroute works, by incrementing the TTL so that the datagram 'dies' at every next hop, we can deduct that the default max ttl is equal to 30 hops. The manual confirms this.

Question 5)

Describe the role of a default gateway in a routing table?
A default gateway is the node that is assumed to know how to forward packets on to other networks. In a IP network, nodes have this default route setting where packets are sent for which they can not determine a specific route.

Question 6)

What is the network prefix of IP address 192.110.50.3/24?
This ip-address has a subnet mask equal to 255.255.255.0 (/24).
Binary this can be written as:
11111111.11111111.11111111.00000000
combined with the ip address we can calculate the network prefix easily here:
192.110.50.0
With the host part = 0.0.0.3

Question 7)

Explain the difference between an IP address and a network prefix.
The network prefix is part of the IP addresses of all hosts in the same network.
An IP address is unique in this network, while the prefix can be seen as the 'range' in which all IP addresses can be found.
For example the prefix 192.0.1.0 with subnet mask 24 (255.255.255.0) corrsponds to all IP addresses 192.0.1.x with x ranging from 1-254, and broadcast on 255.

An IP address can be seen as a prefix with length (=subnet mask) equal to 32. This leaves but one specific address in the range.

Question 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.

This mask can be notated as 255.255.0.0. This gives 65534 different hosts ranging from : 140.25.0.1 to 140.25.255.254. (140.25.255.255= broadcast).

When creating up to 60 hosts per network, we can work with 64 addresses/network (including broadcasting).

This gives us an amount of maximum 1024 different networks to be set-up on this subnet.

3.2 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 3.4. 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 as displayed in Figure 3.5. The topology for Part 7 is shown in Figure 3.6. There, you explore the relationship between network prefixes and IP forwarding.

Part 1. Configuring a Linux PC as a Router

Any 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, a procedure known as static routing.

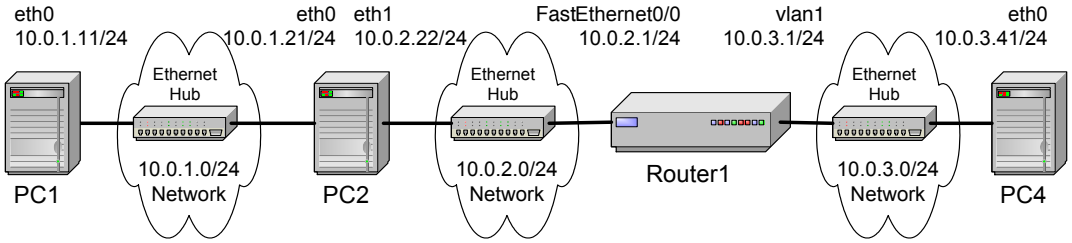

Figure 3.1: Network configuration for Parts 1-4

Linux PC	eth0	eth1
PC1	10.0.1.11/24	Disabled
PC2	10.0.1.21/24	10.0.2.22/24
PC3	10.0.3.41/24	Disabled
Cisco Router	**FastEthernet0/0**	**vlan1**
Router1	10.0.2.1/24	10.0.3.1/24

Table 3.1: IP addresses for Parts 1-4

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

```
C1% ping -c 5 10.0.1.21
C1% ping -c 5 10.0.2.1
C1% ping -c 5 10.0.3.41
```

4. Save the captured wireshark output.

Use the saved data to answer the following questions:

Question 1.A.1)

What is the output on PC1 when the `ping` commands are issued?
For 10.0.1.21, we see the following output:

```
PING 10.0.1.21 (10.0.1.21) 56(84) bytes of data.
64 bytes from 10.0.1.21: icmp_seq=1 ttl=64 time=0.596 ms
64 bytes from 10.0.1.21: icmp_seq=2 ttl=64 time=0.527 ms
64 bytes from 10.0.1.21: icmp_seq=3 ttl=64 time=0.477 ms
64 bytes from 10.0.1.21: icmp_seq=4 ttl=64 time=0.529 ms
```

For both 10.0.2.1 and 10.0.3.41, we see:

connect: Network is unreachable

Question 1.A.2)

Which packets, if any, are captured by Wireshark?
The ICMP ping packets sent to and from 10.0.1.21 (PC2) are the only packets to be captured along with a few ARP packets from PC1 to PC2.

Question 1.A.3)

Do you observe any ARP or ICMP packets? If so, what do they indicate?
Yes, we do see both ARP and ICMP packets. The ICMP packets are the ping packets from 10.0.1.11 (PC1) pinging 10.0.1.21 (PC2). The ARP packets are a request from PC1 asking who has IP address 10.0.1.21.

Question 1.A.4)

Which destinations are not reachable? Explain.
Router1 and PC4 are not reachable. Packets sent to these nodes have to travel through PC2, while we haven't configured PC2 to do any IP forwarding yet.

Exercise 1-b. Configuring a Linux PC as a 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. Hence, enabling IP forwarding is done 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 file name, 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`. Changes to the configuration file `/etc/sysctl.conf` take effect the next time when 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 below.

- Add 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 -net netaddress netmask mask gw gw_address
route add -net netaddress netmask mask dev iface
```

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

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

- Set the default route to IP address gw_address.

```
route add default gw gw_address
```

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

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

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

```
route -e
netstat -r
```

- Display the routing table cache.

```
route -C
```

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 gateways is done with the command

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

The commands to delete the entries created with the above 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
```

There is no simple way to delete all entries in the routing table. One method to flush the routing table is to disable the interface and then enable the interface, as in

```
PC1% ifconfig eth0 down up
```

The following commands are helpful to get information on routing and to find mistakes in the routing setup:

`ping IPaddress`
Tests if IPaddress can be reached.

`traceroute IPaddress`
Displays the route to the interface IPaddress.

When the commands are issued interactively in a Linux Shell, the added entries are valid until Linux is rebooted. To make static routes permanent on Debian-based Linux distributions, the routes need to be entered in the configuration file `/etc/network/interfaces` as post-up commands.

1. Configure the routing table entries of PC1 and PC4. You can either specify a default route or you 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 setup.)

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

Question 1.C.1)
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.

PC 1:

```
root@lab2pc1:/home/student# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.1.21       0.0.0.0         UG    0      0        0 eth0
10.0.1.0        *               255.255.255.0   U     0      0        0 eth0
```

PC2:

```
root@lab2pc1:/home/student# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface1
10.0.1.0        10.0.1.11       255.255.255.0   UG        0 0          0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
10.0.2.0        10.0.2.1        255.255.255.0   UG        0 0          0 eth1
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.3.0        10.0.2.1        255.255.255.0   UG        0 0          0 eth1

VRAGEN WAAROM PC1 NIET KAN PINGEN VERDER DAN 10.0.2.1
```

PC4:

```
root@lab2pc1:~# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.0.1.0        10.0.3.1        255.255.255.0   UG        0 0          0 eth0
10.0.2.0        10.0.3.1        255.255.255.0   UG        0 0          0 eth0
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

The Destination field specifies the address to which outgoing packets will be matched in order to select the right gateway.
The gateway field specifies the gateway address a packet should be sent to if a match was made.
The Genmask field specifies which bits of the Destination field must match the IP address of the outgoing packet; every bit (in the binary representation of the numbers) that is 1 in the Genmask must match, every bit that equals 0 must not. In this case the first 24 bits of the destination IP address of an outgoing packet must be equal to the address in the Destination field in order to match.
For example, any outgoing packet on PC2 with a destination IP address in the range 10.0.3.0-255 would match the fifth entry of PC2's routing table, thus sending the packet to gateway 10.0.2.1.
For the flags, U means that the route is üp¨; G means üse gateway¨. The G flag signifies whether the destination is directly connected. If it is, the G flag won't be present. Other flags not seen here exist, such as H (target is a host), or C (cache entry).
MSS is the default maximum segment size for TCP connections over the route. Since the value of this field is always 0, we assume that there is no limit on the segment size, although it might also be that no TCP whatsoever is allowed.
Window specifies the default window size for TCP connections over the route. Again, every value is 0 in this case. We assume a default value will be selected from somewhere else.
irtt is the initial roundtrip time. This field is used to guess about the best TCP parameters initially, before any TCP packets have been sent over the network. Once again, this value is not configured to anything useful here.
Metric is a measure of the distance to the target, usually represented by the number of hops.

Ref is the number of references to this route.
Use is the count of lookups to this route.
Iface specifies the interface the outgoing datagram will have to be sent out of in order to reach the gateway.

Part 2. Configuring a Cisco Router

The setup of the 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 will 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 `minicom` software to access the router. Lastly, you have to type IOS (Internet Operating System) commands using the command line interface of 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 Minicom

Each lab is equipped with 4 cisco 1760 routers and each PC is connected through a serial cable to one of the routers, i.e., PC1 is connected to Router1, PC2 is connected to Router2, etc. You can use the `minicom` command to establish a remote terminal connection to the router. You will use Router1 and PC1 as the console.

Access the console port of Router1 from PC1 using `minicom` by typing:

```
PC1% minicom
```

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 hit Enter key several times.

To terminate a `minicom` session, type `Ctrl-A`, then `Z` which will show a menu. Exit by typing `Q` and following the instructions.

Exercise 2-b. Switching Cisco IOS command modes

This exercise demonstrates how to log into a router and how to operate through 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. Start a `minicom` session on PC1 which is connected to Router1 with a serial cable.
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 system wide 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 FastEthernet0/0
Router1(config-if)#
```

The name of the interface is provided as an argument. Here, the network interface that is configured is *FastEthernet0/0*.

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:

```
Router1# exit
Router1 con0 is now available Press RETURN to get started.
```

Exercise 2-c. Configuring IP interfaces on a Cisco router

For this course we will be working with the Cisco 1760 Router, which is shown in Figure 3.2.

The Cisco 1760 router has the following interfaces.

- 1 Ethernet Port: FastEthernet0/0
- 1 Ethernet Switch Module: vlan1.

The 4 ports of the switch module have the following names *FastEthernet0/1*, *FastEthernet0/2*, *FastEthernet0/3*, *FastEthernet0/4*. Note that you can use the shorthand *FA0/X* instead of writing *FastEthernet0/X*.

The easiest way to configure the router is to:

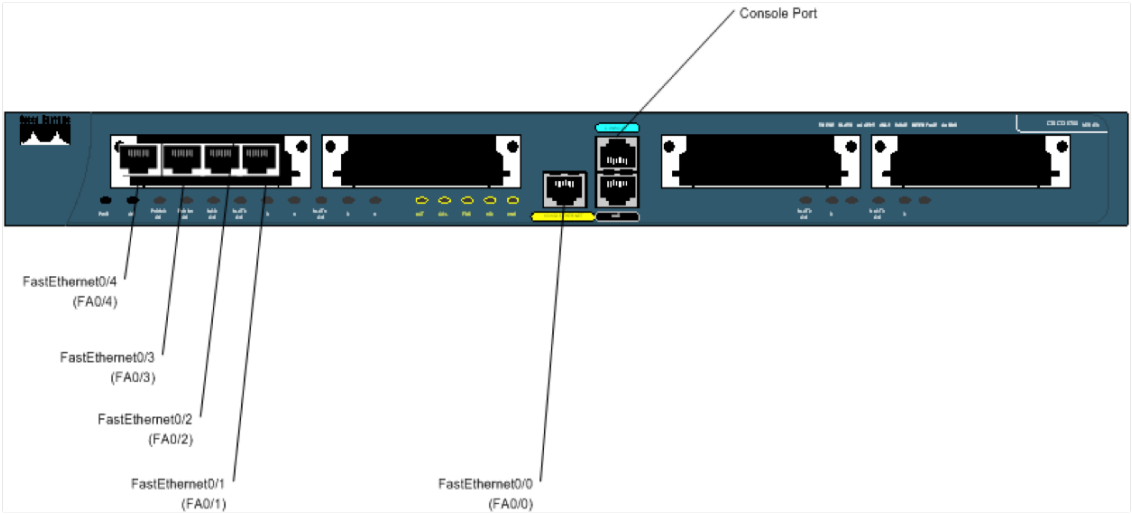

Figure 3.2: Cisco 1760

- Enable the onboard interface *FA0/0* and give it an IP address.
- Turn on one of the ports of the switch module, we recommend you to always use *FA0/1*.
- Enable the *Vlan1* interface and assign it an IP address.
- We also recommend not changing any of the VLAN settings on the switch module.

In IOS this becomes:

```
Router1(config)# interface FastEthernet0/1
Router1(config-if)# no shutdown
Router1(config-if)# interface vlan1
Router1(config-if)# ip address 10.0.2.1 255.255.255.0
Router1(config-if)# no shutdown
```

Figure 3.3 shows a logical representation of the internal operation of the Cisco1760, and how the virtual interface *Vlan1* can be configured with an IP address.

The following exercises use basic commands from the Cisco IOS that are needed to configure a Cisco router.

1. Start a minicom session on PC1 which is connected to Router1 with a serial cable.
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 FastEthernet0/0
Router1(config-if)# ip address 10.0.2.1 255.255.255.0
```

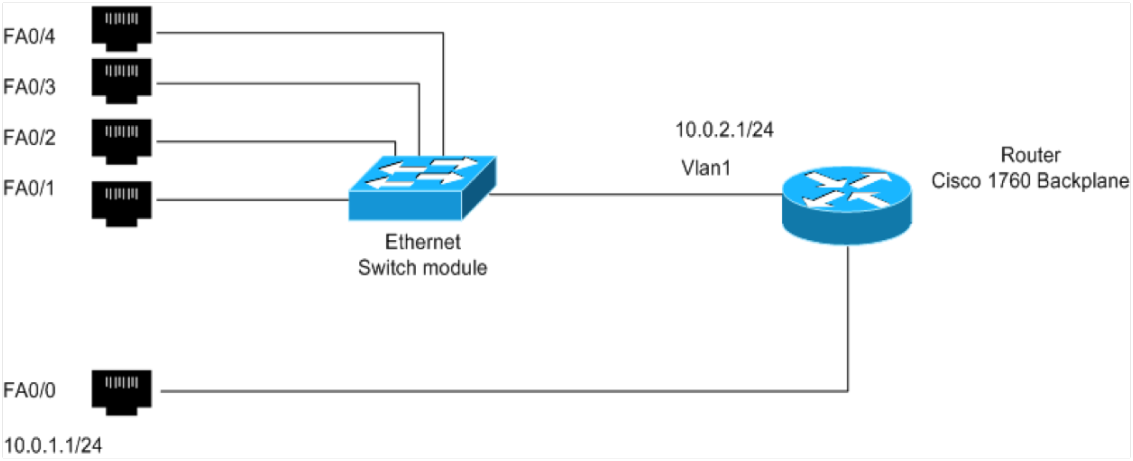

Figure 3.3: Cisco 1760 Switch Module

```
Router1(config-if)# no shutdown
Router1(config-if)# interface FastEthernet0/1
Router1(config-if)# no shutdown
Router1(config-if)# interface vlan1
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.

Question 2.C.1)

Include the output from Step 3 in your lab report.

Router1# show interfaces

```
FastEthernet0/0 is up, line protocol is up
  Hardware is PQUICC_FEC, address is 000e.83f5.ff4c (bia 000e.83f5.ff4c)
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
```

```
     0 input packets with dribble condition detected
     28 packets output, 1680 bytes, 0 underruns
     2 output errors, 0 collisions, 1 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     2 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
FastEthernet0/1 is up, line protocol is up
  Hardware is Fast Ethernet, address is 0012.00d4.bf30 (bia 0012.00d4.bf30)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     35 packets input, 5663 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     104 packets output, 6656 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
FastEthernet0/2 is administratively down, line protocol is down
  Hardware is Fast Ethernet, address is 0012.00d4.bf31 (bia 0012.00d4.bf31)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
FastEthernet0/3 is administratively down, line protocol is down
  Hardware is Fast Ethernet, address is 0012.00d4.bf32 (bia 0012.00d4.bf32)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
```

```
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
FastEthernet0/4 is administratively down, line protocol is down
  Hardware is Fast Ethernet, address is 0012.00d4.bf33 (bia 0012.00d4.bf33)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Vlan1 is up, line protocol is up
  Hardware is EtherSVI, address is 000e.83f5.ff4c (bia 000e.83f5.ff4c)
  Internet address is 10.0.3.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 1000000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:01:25, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     3 packets input, 321 bytes, 0 no buffer
     Received 3 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     2 packets output, 120 bytes, 0 underruns
     0 output errors, 3 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

Router1# show running-config

```
Building configuration...
```

```
Current configuration : 842 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot system flash:c1700-advipservicesk9-mz.124-25d.bin
boot-end-marker
!
enable password mvkbj1n
!
no aaa new-model
ip cef
!
!
!
!
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
 ip address 10.0.2.1 255.255.255.0
 speed auto
!
interface FastEthernet0/1
!
interface FastEthernet0/2
 shutdown
!
interface FastEthernet0/3
 shutdown
!
interface FastEthernet0/4
 shutdown
!
interface Vlan1
 ip address 10.0.3.1 255.255.255.0
!
ip forward-protocol nd
!
!
no ip http server
```

```
70 no ip http secure-server
   !
72 no cdp run
   !
74 !
   !
76 control-plane
   !
78 !
   !
80 !
   !
82 !
   !
84 !
   !
86 line con 0
   line aux 0
88 line vty 0 4
    login
90 !
   end
```

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. Below is a summary of the commands.

The following can be executed in the privileged EXEC mode.:

`show ip route`
Display the contents of the routing table.

`clear ip route *`
Delete all routing table entries.

`show ip cache`
Display the routing cache.

The following can can be executed in the Global Configuration mode.

`ip route-cache`
Enable route caching. By default, route caching is enabled on a router.

`no ip route-cache`
Disable route caching.

`ip route destination mask gw_address`
Add 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`
Add 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`
Delete 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 2, 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 gateways 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 above 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.

Question 2.D.1)

Include the saved output of the routing table from Step 1 and Step 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.

Before:

```
Router1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 2 subnets
C       10.0.2.0 is directly connected, FastEthernet0/0
C       10.0.3.0 is directly connected, Vlan1
```

After:

```
Router1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.0.2.0 is directly connected, FastEthernet0/0
C       10.0.3.0 is directly connected, Vlan1
S       10.0.1.0 [1/0] via 10.0.2.22
```

The first column displays the flags that apply to each entry. The flags are explained above. For the entries in our tables, we see C and S. C means that the destination subnet is directly connected to the router. S means that the entry is static. We did indeed add the static entry ourselves.

How the routing table changes: An entry was added for subnet 10.0.1.0/24, for packets to be forwarded to 10.0.2.22 (PC2 eth1)

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. The table below 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.

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 if the status lights are on. Verify the connection of the cables. Verify that no cross-over 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 messages 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

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 Wireshark 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 Wireshark and save the traffic generated by the `traceroute` command.
4. Save the routing table of PC1, PC4, PC2 and Router1.

Question 3.B.1)

Use the Wireshark output and the previously saved routing table to explain the operation of `traceroute`.

traceroute sends packets with different time-to-live (TTL) to the destination, starting from TTL=1 and incrementing on each following packet.
The packet with TTL=1 will be decremented to 0 at the first hop it encounters (PC2 in this case), and it will return a "Time-to-live exceeded" ICMP message.
Each packet with initial TTL=n will reach one hop further than a packet with TTL=n-1. traceroute saves the source of every TTL exceeded message and considers it a hop. The hops are ordered based on the time they arrived: the longer it takes for a message to return, the further along the route it is assumed to be.
A packet which reaches the destination will (hopefully) return a "Destination unreachable" message. This is because traceroute sends packets to "unlikely ports". These are ports which are unlikely to be in use, causing the host to send such an error message upon arrival if the port was indeed not in use.

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. Below are the corresponding IOS commands for Cisco routers.

The following can can be executed in the privileged EXEC mode:

`ip arp`
: *Display the contents of the ARP cache*

`clear arp`
: *Delete the entire ARP cache*

The following can can be executed in the Global Configuration mode:

`arp IPaddress`
Add an entry for `IPaddress` to the ARP cache

`no arp IPaddress`
Delete the ARP entry for `IPaddress` from the ARP cache

1. Erase all ARP entries on PC1, PC2, PC4 and Router1.
2. Run Wireshark 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, ICMP echo reply on both PC1 and PC4.

Question 3.C.1)
Determine the source and destination addresses in the Ethernet and IP headers, for the ICMP Echo Request messages that were captured at PC1.
Ethernet:
src: 68:05:ca:1a:7c:70
dst: 68:05:ca:1a:7c:77

IP:
src: 10.0.1.11
dst: 10.0.3.41

Question 3.C.2)
Determine the source and destination addresses in the Ethernet and IP headers, for the ICMP Echo Request message that were captured at PC4.
Ethernet:
src: 00:0e:83:f5:ff:4c
dst: 68:05:ca:1a:7c:6f

IP:
src: 10.0.1.11
dst: 10.0.3.41

Question 3.C.3)
Use your answers above to explain how the source and destination Ethernet and IP addresses are changed when a datagram is forwarded by a router.
The IP addresses remain the same: the source address is the IP from the host that sent the message, and the destination address is the IP of the host that should receive the message. Every IP address is assumed to uniquely identify a host or router throughout the whole network.

The Ethernet addresses change at each hop, because those addresses are only used link-local.
The destination Ethernet address of the packet captured at PC1 is the hardware address of PC2, and the source Ethernet address of the packet captured at PC4 is the hardware address of Router1.

In the link between PC2 and Router1, the source address will be the hardware address of PC2 and the destination address will be the hardware address of Router1. So every time a router forwards a packet, it will use its own Ethernet address as the source and the Ethernet address of the next hop as the destination.

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 Wireshark 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 Wireshark and PC1âĂŹs routing table.

Question 3.D)

Use the saved output to indicate the number of matches for each of the IP addresses above. Explain how PC1 resolves multiple matches in the routing table. Only include relevant output data in your report to support your analysis of the data.
PC1's routing table:

```
default         10.0.1.21       0.0.0.0         UG    0 0        0 eth0
10.0.0.0        10.0.1.71       255.255.0.0     UG    0 0        0 eth0
10.0.1.0        *               255.255.255.0   U     0 0        0 eth0
10.0.3.0        10.0.1.61       255.255.255.0   UG    0 0        0 eth0
10.0.3.9        10.0.1.81       255.255.255.255 UGH   0 0        0 eth0
```

Number of matches:
10.0.3.9: 4
10.0.3.14: 3
10.0.4.1: 2

We see ARP messages asking for 10.0.1.81, followed by messages asking for 10.0.1.61, followed by messages asking for 10.0.1.71.

Longer network prefixes are always preferred.
That's why, for a message sent to 10.0.3.9, PC1 will match the table entry 10.0.3.9/32, thus explaining the ARP message asking for 10.0.1.81.
10.0.3.14 doesn't match any table entries of network prefix length 32, but it does match 10.0.3.0/24, thus explaining the ARP message asking for 10.0.1.61.
10.0.4.1 doesn't match any table entries of network prefix length 24, but it does match 10.0.0.0/16, thus explaining the ARP message asking for 10.0.1.71.

Exercise 3-E. Default Routes

1. Delete the routing table entries added in Step 1 of Exercise 3-D above. (Otherwise, the entries interfere with the remaining exercises in this lab.)
2. Add default routes on PC1 and PC2.
   - On PC1, add a default route with PC2 as the default gateway.
   - On PC2, add a default route with Router1 as the default gateway.
3. Start to capture traffic on PC1 (on *eth0*) and PC2 (on both *eth0* and *eth1*) with Wireshark.
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 Wireshark output.

Question 3.E.1)
What is the output on PC1, when the ping command is issued?

```
No.     Time           Source                Destination           Protocol Length
    Info
      1 0.000000000    10.0.1.11             10.0.10.110           ICMP     98
        Echo (ping) request  id=0x0cbc, seq=1/256, ttl=64
      2 0.001736000    68:05:ca:1a:7c:77     ff:ff:ff:ff:ff:ff     ARP      60
        Who has 10.0.1.11?  Tell 10.0.1.21
      3 0.001760000    68:05:ca:1a:7c:70     68:05:ca:1a:7c:77     ARP      42
        10.0.1.11 is at 68:05:ca:1a:7c:70
      4 0.002264000    10.0.2.1              10.0.1.11             ICMP     70
        Destination unreachable (Host unreachable)
      5 1.001440000    10.0.1.11             10.0.10.110           ICMP     98
        Echo (ping) request  id=0x0cbc, seq=2/512, ttl=64
      6 1.003132000    10.0.2.1              10.0.1.11             ICMP     70
        Destination unreachable (Host unreachable)
      7 2.003310000    10.0.1.11             10.0.10.110           ICMP     98
        Echo (ping) request  id=0x0cbc, seq=3/768, ttl=64
      8 2.005025000    10.0.2.1              10.0.1.11             ICMP     70
        Destination unreachable (Host unreachable)
```

```
     9 3.005216000    10.0.1.11             10.0.10.110           ICMP     98
       Echo (ping) request  id=0x0cbc, seq=4/1024, ttl=64
11   10 3.006920000    10.0.2.1              10.0.1.11             ICMP     70
       Destination unreachable (Host unreachable)
     11 4.007100000    10.0.1.11             10.0.10.110           ICMP     98
       Echo (ping) request  id=0x0cbc, seq=5/1280, ttl=64
13   12 4.008788000    10.0.2.1              10.0.1.11             ICMP     70
       Destination unreachable (Host unreachable)
     13 6.002800000    68:05:ca:1a:7c:70     68:05:ca:1a:7c:77     ARP      42
       Who has 10.0.1.21?  Tell 10.0.1.11
15   14 6.003358000    68:05:ca:1a:7c:77     68:05:ca:1a:7c:70     ARP      60
       10.0.1.21 is at 68:05:ca:1a:7c:77
```

We see ICMP and ARP packets.

Question 3.E.2)

Determine how far the ICMP Echo Request message travels?

The ICMP Echo Request message travels up to Router1.

This happens because PC1 has set PC2 as the default gateway, and PC2 has set Router1 as its default gateway.

In the router, there is no default gateway and no other match for 10.0.10.110/X. Thus, a "Destination unreachable" ICMP packet is returned by the router.

Question 3.E.3)

Which ICMP Echo Reply message returns to PC1?

Destination unreachable (Host unreachable)

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, where 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 is 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. Proxy ARP is enabled and disabled separately on each interface. In IOS, proxy ARP is enabled by default.

The commands to enable and disable Proxy ARP in the IOS Interface configuration mode are:

```
ip proxy-arp
no ip proxy-arp
```

Exercise 4.

1. Erase the ARP table and the routing table of PC4.
2. Set the netmask of PC4 to 255.0.0.0, so that PC4 assumes it belongs to network 10.0.0.0/8, instead of belonging to the network 10.0.3.0/24.
3. Run Wireshark on PC4 (*eth0*), PC2 (*eth1*), and PC1 (*eth0*). Set a display or capture filter to only display ICMP and ARP packets.
4. Issue a `ping` from PC4 to PC1:

   ```
   PC4% ping -c 2 10.0.1.11
   ```

   Explore the captured data and interpret the outcome. Even though PC4 had no default routing entry in its table for Router1, it was still able to connect to PC1, i.e., you should not observe a “network unreachable” error message.
5. Save the ARP table of PC4 and the packets captured by Wireshark on the hosts.
6. Explore the captured data and interpret the outcome.
7. Now, disable Proxy ARP on both interfaces of Router1. Is it still feasible to issue a ping from PC4 to PC1?
8. Reset the network mask of PC4 to its original value of 255.255.255.0. Then, re-enable Proxy ARP on Router1.

Question 4.1)

Use the captured data to explain the outcome of the exercise. Use the data to explain how Proxy ARP allowed PC4 to communicate with PC1. Include only relevant data from your saved output.

*** icmp en arp van pc4 ****

PC4 sends an ARP request to the router. If Proxy ARP is enabled in the router, the router will forward the ARP request to the PCs on the other subnet, and the router will respond to PC4 with its own MAC address, causing PC4 to map the destination IP address to the

router's Ethernet address. Then, PC4 will send the pings desinted for PC1 to the router's Ethernet address, not being aware that isn't actually PC1. The router will then forward the pings to the other subnet, and they will successfully reach PC1. If Proxy ARP is not enabled in the router, no packets or ARP requests will be forwarded to the other subnet and the result of pings will be D̈estination host unreachable.

Part 5. ICMP Route Redirect

ICMP route redirect messages are sent from a router to a host, when a datagram should have been forwarded to a different router or interface. In Linux, an ICMP Route Redirect message updates the routing cache, but not the routing table.

Both the routing cache and the routing table contain information for forwarding traffic. When a Linux system performs a routing table lookup, it first inspects the routing cache. If no matching entry is found in the cache, Linux performs a lookup in the routing table. After each routing table lookup, an entry is added to the routing cache. The routing cache does not aggregate table entries, and there is a separate entry for each destination IP address. As a consequence, a lookup in the routing cache does not require a longest prefix match. An entry in the routing cache is deleted if it has not been used for some time, usually after 10 minutes. When an ICMP redirect message arrives, an entry is added to the routing cache, but no update is performed to the routing table.

The following are the commands to display the contents of the routing cache:

`route -C`
In Linux

`show ip cache`
In IOS

In this part of the lab, you use three Cisco routers. Figure 3.4 and Table 3.2 describe the network configuration for the exercises below.

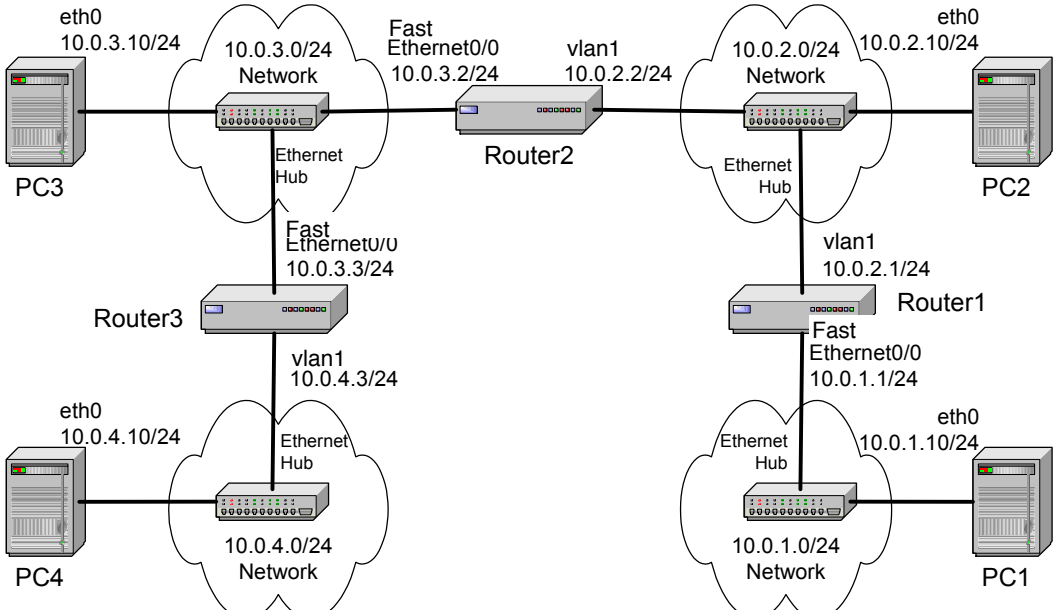

Figure 3.4: Network configuration for Part 5

Cisco Router	FastEthernet0/0	vlan1
Router1	10.0.1.1/24	10.0.2.1/24
Router2	10.0.3.2/24	10.0.2.2/24
Router3	10.0.3.3/24	10.0.4.3/24
Linux PC	**eth0**	**eth1**
PC1	10.0.1.10/24	Disabled
PC2	10.0.2.10/24	Disabled
PC3	10.0.3.10/24	Disabled
PC4	10.0.4.10/24	Disabled

Table 3.2: IP addresses for Part 5

Exercise 5.

In the network shown in Figure 3.4, when PC2 sends datagrams with destination 10.0.3.10 (PC3) to 10.0.2.1 (Router1), as opposed to 10.0.2.2 (Router2), then Router1 sends an ICMP Route Redirect message to PC2. The ICMP Route Redirect informs PC2 that it should send datagrams with destination 10.0.3.10 to Router2 instead.

In this exercise, you create the above scenario. First, you will trigger the transmission of an ICMP Route Redirect message and subsequently observe a change to the routing cache.

1. Connect the Ethernet interfaces of the routers and the hosts to the hubs as shown in Figure 3.4.
2. Delete all routing table entries and all ARP cache entries on all PCs and on Router 1.
   - Delete the routing cache on PC1 with the command:
     ```
     PC1% echo "1" > /proc/sys/net/ipv4/route/flush
     ```
   - Delete all static routes on Router 1 with the following commands:
     ```
     Router1(config)# no ip routing
     Router1(config)# ip routing
     ```
   - Build a new static routing entry on Router1 for network prefix 10.0.3.0/24 as follows:
     ```
     Router1(config)# ip route 10.0.3.0 255.255.255.0 10.0.2.2
     ```
3. Setup the routing table of PC2 in such a way that it provokes the transmission of an ICMP Route Redirect message as discussed above.
4. Save the contents of the routing table and the routing cache of Router1, Router2, and PC2.
5. Use Wireshark to capture the ICMP messages being sent, and issue a ping from PC2 to PC3:
   ```
   PC2% ping -c 5 10.0.3.10
   ```
6. Save the network traffic and the contents of the routing table and the routing cache after the ICMP Route Redirect messages.
7. Wait a few minutes and check the contents of the routing cache again. Save the output.

Question 5.1)
Is there a difference between the contents of the routing table and the routing cache immediately after the ICMP Route Redirect message?
The routing cache has been deprecated since linux 3.6 and is therefore empty throughout this exercise.

Question 5.2)

When you viewed the cache a few minutes later, what did you observe?
The routing cache has been deprecated since linux 3.6 and is therefore empty throughout this exercise.

Question 5.3)

Describe how the ICMP Route Redirect works using the output you saved. Include only relevant data from your saved output to support your explanations.
PC2 sends ping requests for PC3 to Router1 instead of Router2.
Router1 notices the routing table entry for the subnet 10.0.3.0 to Router2 via interface vlan1.
This is the same interface where the ping request arrived in, so Router1 responds with an ICMP redirect message to PC1 and forwards the ping request to Router2.

We can confirm this by looking at messages with no. 8, 9, 10 and 11 in 5.pcap.
Message 8 is sent by PC2 and goes to Router1.
Message 9 is the redirect message from Router1 to PC2.
Message 10 is the forwarded ping request by Router1 to Router2.
Message 11 is the ping reply from Router2 to PC2.

Question 5.4)

Explain how Router1, in the above example, knows that datagrams destined to network 10.0.3.10 should be forwarded to 10.0.2.2?
Router1 knows this because of the entry we added to the routing table with:

```
1  Router1(config)# ip route 10.0.3.0 255.255.255.0 10.0.2.2
```

Part 6. Routing Loops

A potential problem when setting routing tables manually is that routing loops may occur. In this part of the lab, you intentionally configure a routing loop in the configuration of the routing table and observe what happens to network traffic in such a situation.

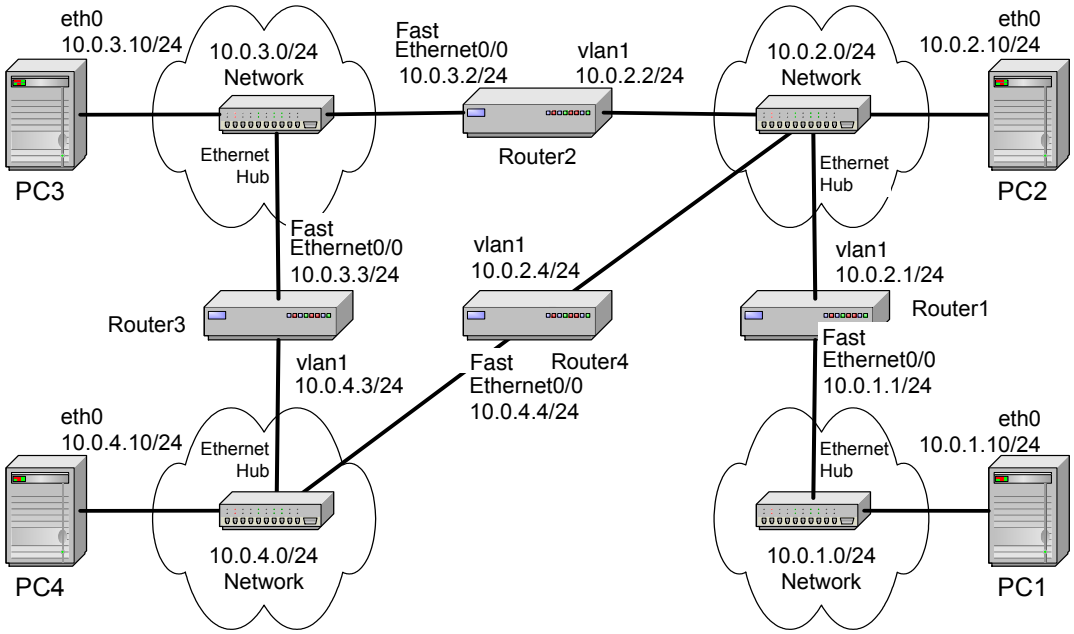

Figure 3.5: Network configuration for Part 6

Cisco Router	FastEthernet0/0	vlan1
Router4	10.0.4.4/24	10.0.2.4/24

Table 3.3: IP addresses for Part 6

Exercise 6.

1. Add Router4 to the network topology of Part 5 and configure the interfaces as shown in Figure 3.5 and Table 3.3 above.
2. Configure the routing tables of Router2, Router3 and Router4, so that an ICMP Echo Request message generated by a ping from PC4 to PC1 creates an infinite loop. Issue a `traceroute` to verify that a loop exists:

   ```
   PC4% traceroute 10.0.1.10
   ```

   You should observe that the traced path is a loop.
3. Start Wireshark sessions on PC2, PC3, and PC4.
4. Issue a ping from PC4 to

   ```
   PC4% ping -c 1 10.0.1.10
   ```

Observe in Wireshark that the same ICMP Echo Request message is looping.

5. Save the routing tables of Router2, Router3 and Router4. Count the number of times you see the ICMP Echo Request message, as captured by Wireshark on PC4. Save at least two of these ICMP Echo Request messages for the lab report.

Question 6.1)

Are the two ICMP packets that you saved identical? If not, what is different? Include the packet data in your lab report to substantiate your claims.
No, they have different values for TTL. We the echo request message 22 times. Also, the very first message has a different source hardware address, namely PC4's hardware address, where the message originated from. The other times the source hardware address was the address corresponding with Router4's FastEthernet0/0 interface.

Question 6.2)

Why does the ICMP Echo Request packet not loop forever in the network?
Because the TTL value has dropped to 1. After that it is discarded

Part 7. Network Prefixes and Routing

In this exercise you study the role that network prefixes (netmasks) play when hosts determine if a datagram can be directly delivered or if it must be sent to a router.

This part uses the network setup shown in Figure 3.6. The network includes one router, four hosts and two hubs. The IP addresses of all devices are given in Table 3.4. Here, each host has only a default route. In other words, the routing table at a host only knows about the directly connected networks and the default gateway.

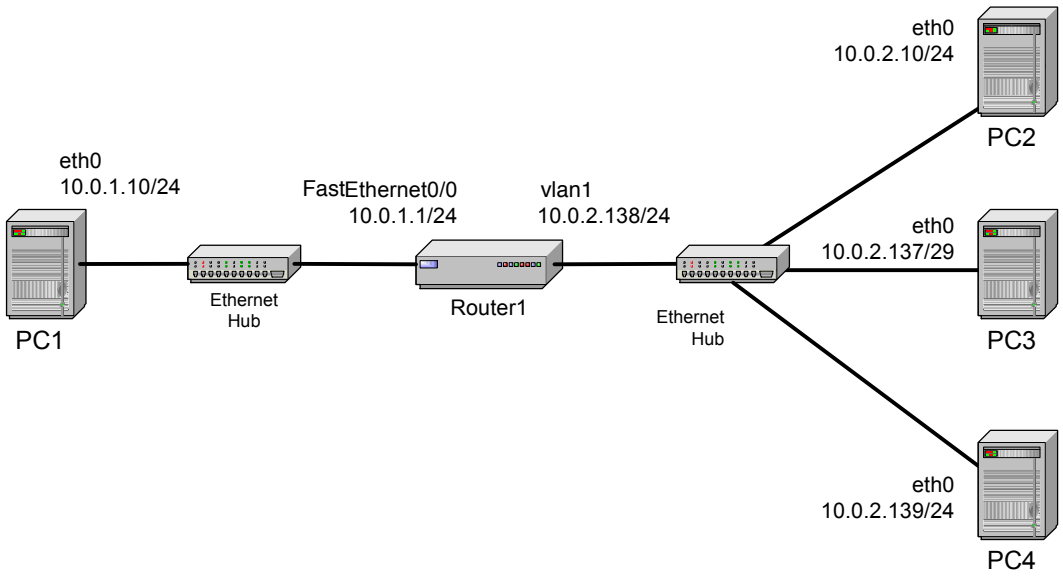

Figure 3.6: Network configuration for Part 7

Linux PC	eth0	eth1
PC1	10.0.1.10/24	Disabled
PC2	10.0.2.10/24	Disabled
PC3	10.0.2.137/29	Disabled
PC4	10.0.2.139/24	Disabled
Cisco Router	**FastEthernet0/0**	**vlan1**
Router1	10.0.1.1/24	10.0.2.138/24

Table 3.4: IP addresses for Part 7

Exercise 7.

In this exercise, you explore how hosts that are connected to the same local area network, but that have different network addresses or netmasks, communicate or fail to communicate.

1. Configure the hosts and the router to conform to the topology shown in Figure 3.6, using the IP addresses as given in Table 3.4. Note that PC2, PC3, and PC4 have different network addresses and different netmasks.

2. Add Router1 as default gateway on all hosts. For example, for PC1, the command is:

```
PC1% route add default gw 10.0.1.1
```

3. Issuing ping commands from PC1:
   a. Clear the ARP table on all hosts.
   b. Start Wireshark on PC1 and on PC4, and set the capture filter to capture ICMP and ARP packets only.
   c. Check the ARP tables, routing tables and routing caches of each host. Save the output. (Make a note that these are the table entries from Step 3 before the ping is issued.)
   d. Issue a `ping` command from PC1 to PC2 and PC3

```
PC1% ping -c 2 10.0.2.10
PC1% ping -c 2 10.0.2.137
```

   e. Save the ARP tables, routing tables and routing caches of each host (Make a note that these are the table entries from Step 3 after the ping is issued.)
   f. Save the output of the ping command at PC1 and the output of Wireshark on PC1 and PC4.

4. Issuing a ping command from PC3 to PC4:
   a. Clear the ARP table on all hosts.
   b. Start Wireshark on PC3, and set the capture filter to capture ICMP and ARP packets only.
   c. Check the ARP tables, routing tables and routing caches of each host. Save the output. (Make a note that these are the table entries from Step 4 before the ping is issued.)
   d. Issue a `ping` from PC3 to PC4.

```
PC3% ping -c 3 10.0.2.139
```

   e. Save the ARP tables, routing tables and routing caches of PC3 (Make a note that these are the table entries from Step 4 after the ping is issued.)
   f. Save the output of the `ping` command and the output of Wireshark on PC3.

5. Repeat Step 4, but this time issue a ping from PC3 to PC2. Note that once an entry is made in the routing cache, you cannot repeat the above experiment and obtain the same results; you have to wait until the routing cache is reset (which take some time).

Question 7.1)

Explain what you observed in Steps 3, 4 and 5. Use the saved data to support your answers. Provide explanations of the observations. Try to explain each observed phenomenon, e.g., if you observe more ICMP Echo Requests than ICMP Echo Replies, try to explain the reason.

Step 3: Since the router is used as the default gateway for PC1, every ping is sent to the router. The ping to PC2 is delivered to PC2 without a problem and the subnets match up, so nothing is out of the ordinary. The ping to PC3 succeeds too because interface vlan1 of the router (10.0.2.138/24) is c̈oincidentallyẅithin what PC3 considers its subnet (which contains all IP addresses in the range 10.0.2.137-142). For the ping to PC4, the situation is the same as with the ping to PC1.

Step 4: As before, because PC4's IP address is within what PC3 considers its subnet, the ping succeeds again.

Step 5: The ping from PC3 to PC2 succeeds because the ping request is sent to the default gateway (the router), and the router then forwards the request successfully because PC2 is in its subnet, as is PC3, so the replies are also sent back successfully.

Question 7.2)

If PC3 had no default entry in its table, would you have seen the same results? Explain for each of the pings above what would have been different.

No. The pings from PC1 to PC3 and from PC3 to PC2 would end in a n̈etwork is unreachable error.T̈his is because PC1 and PC2 are not in the same subnet as PC3 from PC3's point of view (HostMin: 10.0.2.137, HostMax: 10.0.2.142) .