LAB THREE – STATIC ROUTING

In this lab you will work with four different network topologies. The topology for Parts 1-4 is shown in Figure 3.1. These parts address router configuration on Linux PCs 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.

For further lab works, you might work with two network interfaces from each PCs. To configure the network two interfaces from GNS3 and Virtual Box, please follow the instrucitons.

1. Stop the instance of host from GNS3.
2. Go to Virtual Box from the corresponding VM -> go to settings -> Network -> Select another adapter -> Click enable check box
3. Go to GNS3 preferences -> Select the VM -> Select edit -> Select Network tab -> Increase number of adapter to 2 from drop down box -> Uncheck the check box below that says it can select any adapter -> Apply -> Ok
4. Start the host from GNS3.

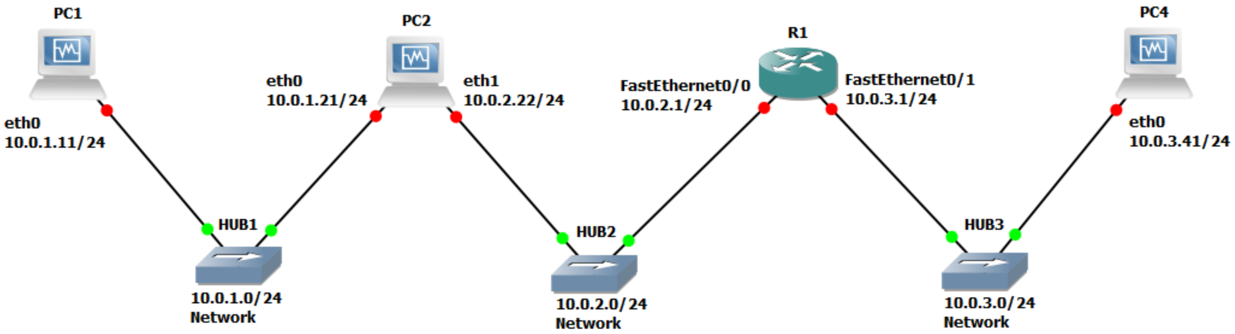

Figure 3.1 Network Topology for Parts 1-4

Virtual Machines	eth0	eth1
PC1	10.0.1.11 / 24	None / Disabled
PC2	10.0.1.21 / 24	10.0.2.22 / 24
PC4	10.0.3.41 / 24	None / Disabled
Cisco Router	**FastEthernet0/0**	**FastEthernet0/1**
Router1	10.0.2.1 / 24	10.0.3.1 / 24

Table 3.1 IP addresses for Parts 1-4

PART 1. Configuring a CISCO Router

The setup of the Cisco router is more involved. There are different ways to connect to a Cisco router such as by the Serial or Ethernet ports or connections. The first step is to start the router in GNS3, and then open the console window so that the configuration commands can be entered. Once in the console you have to type IOS 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 1(A). Switching Cisco IOS Command Modes

This exercise demonstrates how to log into a router and how to work with 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. Connect the Ethernet interfaces of the Linux PCs and the Cisco router as shown in Figure 3.1. Do not turn on the Linux PCs yet.

2. Right-click on Router1 and choose `Start`.

3. Right-click on Router1 and choose `Console`. Wait a few seconds until the router is initialized. If everything is fine, you should see the prompt shown below. This is the User EXEC mode. If the prompt does not appear, try to restart GNS3 and repeat the setup again.

```
Router1>
```

4. To see which commands are available in this mode, type `?`:

```
Router1> ?
```

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

```
Router1> enable
Router1#
```

6. Type the following command to disable the Privileged EXEC mode

```
Router1#
disable
```

NOTE
The Cisco routers in GNS3 sometimes start up in Privileged instead of the User EXEC mode. There is no explanation as to why that happens.

7. To modify system wide configuration parameters, you must enter the global configuration mode. This mode is entered by typing:

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

Tip:
Almost all terminal commands can be reduced to shorter commands.

Example: `configure terminal` can be reduced to `conf t`

8. 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`.

9. 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#
```

10. To terminate the console session from the User EXEC mode, type `logout` or `exit`:

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

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

Exercise 1(B). Configuring a Cisco Router via the console

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

1. Right-click on Router1 and choose `Start`.

2. Right-click on Router1 and choose `Console`. Wait some seconds until the initial console window is set up. When the router is ready to receive commands, proceed to the next step.

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

IOS MODE: GLOBAL CONFIGURATION

`ip routing` `no ip routing`

Enables or disables IP forwarding. When it is disabled, it also deletes the content of the routing table.

IOS MODE: INTERFACE CONFIGURATION

`no shutdown` `shutdown`

Enables or disables, respectively, a network interface.

```
Router1> enable
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
Router1(config-if)# no shutdown
Router1(config-if)# interface FastEthernet0/1
Router1(config-if)# ip address 10.0.3.1 255.255.255.0
Router1(config-if)# no shutdown
Router1(config-if)# end
```

Tip:
"no ip routing" is used to guarantee that the routing cache is empty, not routing table.

Tip to save time:
In future labs when you are configuring Cisco routers, it will be very tiring to manually type in the configuration data for a router, every time you set up a lab, as you just did for Router1 above. For the purpose of this lab, type it out so that you have a feel for the commands. In future labs you can save time by typing all the configuration lines into an Excel file and simply copying and pasting it into the router console window. To do this you will need to type every required command line in an Excel sheet and then copy all of them at once. Then go to the router console window and simply right click and all of the copied data should be pasted into the router console window hence saving you all the repetitive typing.

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

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

Exercise 1(C). Setting static routing table entries on a Cisco router

In this exercise, you will 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 routes are configured manually, which is also referred to as *static routing*.

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 below.

IOS MODE: PREVILEGED 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
```

```
no ip route-cache
```

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

```
ip route destination mask gw_address
```

```
no ip route destination mask
```

Adds or deletes 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
```

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

Adds or delets 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`).

We next show some examples for adding and deleting routing table entries in IOS. Note that whenever an IP address is configured for a network interface on a router, routing table entries for the directly connected network are added automatically.

The command for adding a route for the network address `10.0.1.0/24` with `10.0.2.22` as the next-hop address is

```
Router1(config)# ip route 10.0.1.0 255.255.255.0 10.0.2.22
```

NOTE
This is very important because if you do not set up the IP routes between the routers, the routers will never be able to ping each other from remote networks.

The command bellow is to add a host route to host IP address `10.0.2.65` with the next-hop set to `10.0.1.21.` In IOS, a host route is identified by a 32 bit prefix.

```
Router1(config)# ip route 10.0.2.65 255.255.255.255 10.0.1.21
```

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 to delete the any specific entry use the `no ip route` command. For example:

```
Router1(config)# no ip route 10.0.1.0 255.255.255.0 10.0.2.22
Router1(config)# no ip route 10.0.2.65 255.255.255.255 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 and operates correctly for the configuration shown in Figure 3.1. Routing entries should exist for the following networks.
   a) `10.0.1.0/24`
   b) `10.0.2.0/24`
   c) `10.0.3.0/24`

3. Display the routing table again with `show ip route` and save the output.

Lab Questions

- Explain the fields of the routing table entries of the Cisco router.

PART 2. Configuring a LINUX PC as an IP 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. PC1 and PC4 are hosts, and PC2 will be set up as an IP router.

Exercise 2(A). Network setup

1. Start all the Linux PCs on GNS3. Then, configure the IP addresses of the interfaces as given in Table 3.1.

2. Start Wireshark to capture traffic on PC1.

3. Issue a `ping` command from PC1 to PC2, Router1 and PC4, respectively.

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

4. Save the captured Wireshark output.

Lab Questions

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 Wireshark?
- Do you observe any ARP or ICMP packets? If so, what do they indicate?
- Why are some of the destinations not reachable? Which ones are they?

Exercise 2(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. The command has an immediate effect. However, changes are not permanent and are lost when the system is rebooted.

IP forwarding is disabled with the command

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

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 `/etc/sysctl.conf` take effect the next time Linux is rebooted.

1. Check whether PC2 is enabled to forward IP packets with the following command:

```
PC2% sysctl net.ipv4.ip_forward
```

2. Set up PC2 as an IP router as explained earlier.

3. Use the `sysctl` command from Step 1 again to check that PC2 has IP forwarding enabled.

Exercise 2(C). Setting static routing table entries for a Linux PC

Next, you will 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.

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 below.

```
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 identifies 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 with specific arguments.

```
route -e
```

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

```
ip route flush table main & ip route flush cache
```

first one deletes all entries in the routing table on a PC, the second one deletes all entries in a routing cache on a PC
Please note that the default route should be added before adding any other static route entries, once the table and cache flush happens.

```
route -C
```

Displays the routing table cache.

Example: The command for adding a route for the network address `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
```

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.

Tip: The listed commands are helpful to get information on routing and to find mistakes in the routing setup. The `ping` command tests whether **`IPaddr`** can be reached or not, and the `traceroute` command displays the route to an **`IPaddr`**.

```
ping IPaddr
```

```
traceroute IPaddr
```

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 soon.

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

Lab Questions

- Explain the entries in the routing table and discuss the values of the fields for each entry.

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. In most real network configurations, the network configuration requires additional changes before all hosts and routers can send and receive IP datagrams. However, if the network is not configured properly, you will 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 attain for your lab experiments to work well.

Problem	Possible Causes	Debugging
Traffic does not reach destinations on local subnet.	Network interface not configured correctly. Incorrectly connected or loose cables.	Verify the interface configuration with `show protocols` (in IOS) or `ifconfig` (in Linux). Verify the connection of the interfaces on GNS3.
Traffic reaches router, but is not forwarded to remote subnets.	IP forwarding is not enabled. Routing tables are not configured correctly.	Use `show protocols` to display forwarding status in IOS and `sysctl` in Linux 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.	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.	Flush the ARP table. In Linux, delete entries with `arp -d IPAddress`. In IOS, use the command `clear arp.`

Exercise 3(A). Finalizing the network setup

1. 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 software configuration of routers and hosts. If all `ping` commands are successful, the network configuration is correct, and you can proceed to the next step.

2. Start a Wireshark session on PC1.

3. Execute a `traceroute` command from PC1 to PC4, and save the output.

```
PC1% traceroute 10.0.3.41
```

4. Execute a `trace` command from Router1 to PC4, and save the output.

```
Router1# trace 10.0.3.41
```

5. Stop Wireshark and save the captured traffic. Observe how `traceroute` commands gather route information.

6. Save the routing table of PC1, PC4, PC2, and Router1.

Lab Question

Using the Wireshark output and the previously saved routing tables, explain the operation of `traceroute` command.

Exercise 3(B). 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, it modifies the destination address in the Ethernet header.

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
```

```
no arp IPaddress
```

Adds or deletes an ARP entry for `IPaddress` to the ARP cache.

1. Erase all ARP entries on PC1, PC2, PC4, and Router1

2. Run Wireshark on both PC1 and PC4.

3. Issue a `ping` command on PC1 to PC4.

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

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

Lab Questions

- 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 answers above to explain how the source and destination Ethernet and IP addresses are changed when a datagram is forwarded by a router.

Exercise 3(C). Order of the routing table lookup

A router or host uses a routing table to determine the next hop of the path of an IP datagram. Generally, 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:
   a) `10.0.3.9`
   b) `10.0.3.14`
   c) `10.0.4.1`

3. Start a Wireshark session on PC1, and issue the following `ping` commands from PC1:

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

   Note that gateways with IP addresses `10.0.1.61, 10.0.1.71,` and `10.0.1.81` do not exist. However, Router1 still sends ARP Request packets for these IP addresses.

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

Lab Question

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.

Exercise 3(D). Default routes

1. Delete the routing table entries added in Step 1 of Exercise 3(C) above using the "route del" command. (Otherwise, the entries will interfere with the remaining exercises in this lab.)

2. Add default routes on PC1 an 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 `FastEthernet0/0` of Router1 as the default gateway.

3. Start to capture traffic on PC1 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 10.0.10.110 -c 5
```

5. Save the Wireshark output.

Lab Questions

Use the saved output 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) enables a host to send packets to remote subnets 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 subnets for a host that is on another of its connected subnets. 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 subnet.

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 below.

IOS MODE: INTERFACE CONFIGURATION

```
ip proxy-arp
```

```
no ip proxy-arp
```

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

Exercise 4. Observing Proxy ARP

1. Erase both 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 subnet `10.0.0.0/8`, not `10.0.3.0/24`.

3. Run Wireshark on PC4, PC2 (`eth1`), and PC1. Set a display or capture filter to ICMP and ARP packets only.

4. Issue a ping from PC4 to PC1:

```
PC1% ping 10.0.1.11 –c 2
```

   a) 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` response.

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 subnet mask of PC1 to its original value of `255.255.255.0`. Re-enable Proxy ARP on Router1.

Lab Question

- 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.

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. Before a Linux system performs a routing table lookup, it first inspects the routing cache. If no matching entry is round 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.

Recall the following commands to display the contents of the routing cache in Linux (Ubuntu):

```
route -C
```

```
ip route show cache
```

To clear the route cache in Linux (Ubuntu):

```
ip route flush cache
```

Similarly for IOS the commands are:

```
show ip cache
```

```
clear ip cache
```

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

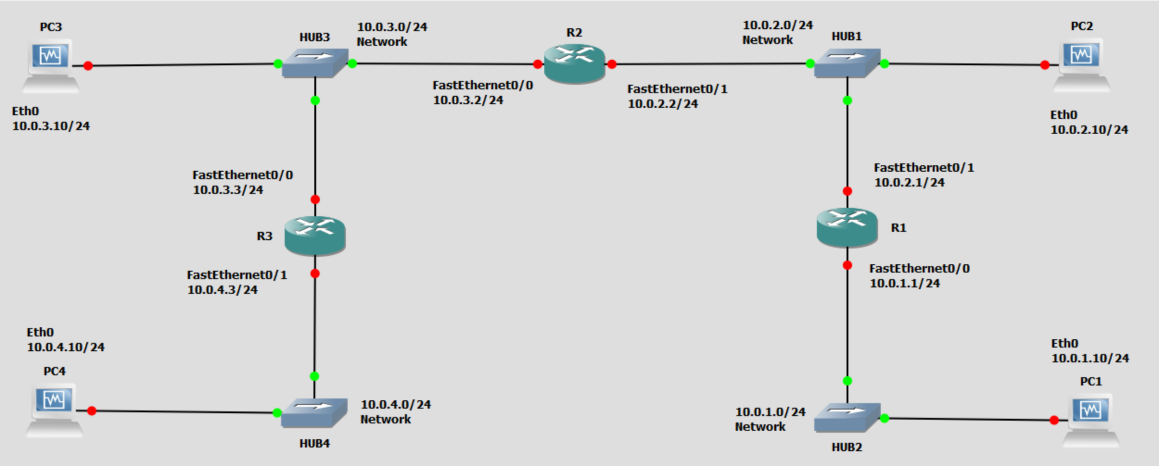

Figure 3.2 Network topology for Part 5

Cisco Routers	**FastEthernet0/0**	**FastEthernet0/1**
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	**Ethernet Interface eth0**	**Ethernet Interface 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 address for Part 5

Exercise 5. Observing ICMP Redirect

In the network shown in Figure 3.2, 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 will create the above scenario. 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 3.2.

2. Delete the routing table entries and ARP cache entries on all PCs and on Router1.

3. Delete the routing cache on PC2

```
PC2% ip route flush cache
```

4. Delete all static routes on Router1.

5. Build a new static routing entry on Router1 for network 10.0.3.0/24 to Router2 (FastEthernet0/1)

6. ICMP redirect messages can be used to attack a network. For this reason, hosts by default ignore ICMP redirect messages. As a similar process to `ip_forward` in Exercise 2(B), on a Linux system, the `accept_redirects` variable controls whether the host can accept or not a redirect ICMP message.

a) Use `sysctl` command to verify the current ICMP redirect status on PC2

```
PC2% sysctl net.ipv4.conf.all.accept_redirects
```

b) If the file `/proc/sys/net/ipv4/conf/all/accept_redirects` contains a `1`, it is accepting ICMP redirect messages, and if it contains a `0`, it is disabled. Enable PC2 to accept ICMP redirect messages.

```
PC2% echo 1 > /proc/sys/net/ipv4/conf/all/accept_redirects
```

c) Use the `sysctl` command again to ensure that the parameter change occurred.

7. Set up the routing table of PC2 in such a way that it provokes the transmission of an ICMP route redirect message as discussed above. In other words, force it to send the packet to a router other than the one you would expect to have it use to get to PC3.

8. Save the contents of the routing table and the routing cache on each of Router1, Router2, and PC2.

9. Set up the routing table of PC3 so that it can reach PC2.

10. Use Wireshark to capture the ICMP messages being sent, and issue a ping from PC2 to PC3.

11. Save the network traffic and the contents of the routing table and the routing cache of PC2, and Router1, Router2 after the ICMP redirect messages.

12. Wait a few minutes with no action and check the contents of the routing cache again. Save the output.

Lab Questions

- Is there a difference between the contents of the routing table and the routing cache immediately after the ICMP route redirect message?
- When you viewed the cache a few minutes later, what did you observe?
- Describe how the ICMP route redirect works using the outputs you saved. Include only relevant data from your saved output to support your explanations.
- 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`?

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 will intentionally configure a loop in the network configuration and observe what happens to network traffic in such a situation.

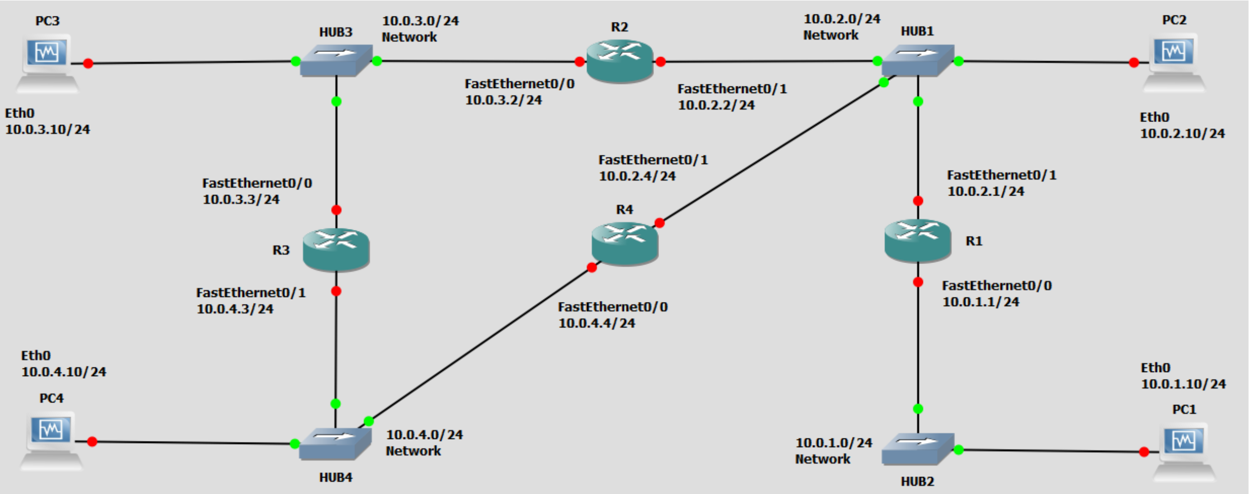

Figure 3.3 Network topology for Part 6

Cisco Routers	FastEthernet0/0	FastEthernet0/1
Router4	10.0.4.4 / 24	10.0.2.4 / 24

Table 3.3 IP addresses for Part 6

Exercise 6. Observing forwarding loops

1. Add Router4 to the network topology and configure the interfaces as shown in Figure 3.3 and Table 3.3.

2. Set up all the PC routing tables to allow communication between all 4 PCs.

3. 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.

4. Start Wireshark sessions on PC2, PC3, and PC4.

5. Issue a ping from PC4 to PC1. You should observe the same ICMP Echo Request message looping in Wireshark.

6. 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.

Lab Questions

- Are the two packets that you saved identical? If not, what is different?
- Why does the ICMP Echo Request packet not loop forever in the network?

PART 7. NETWORK PREFIXES and ROUTING

In this exercise you study how the network prefixes (netmasks) play a role 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 3.4. 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.

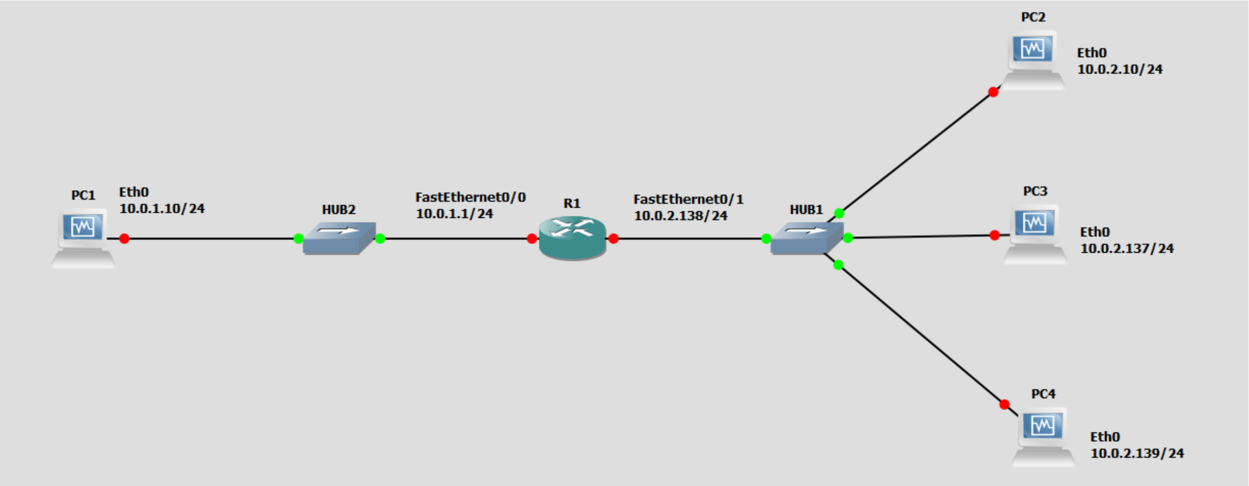

Figure 3.4 Network topology for Part 6.

Linux PC	**Ethernet Interface eth0**	**Ethernet Interface 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 Routers	**FastEthernet0/0**	**FastEthernet0/1**
Router1	10.0.1.1 / 24	10.0.2.138 / 24

Table 3.4. IP addresses for Part 6.

Exercise 7. Exploring the role of prefixes at hosts

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

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

2. Add Router1 as default gateway on all hosts. (PC1, PC2, PC3, and PC4.)

3. **Issue ping commands from PC1**

   a) Clear the ARP table on all PCs.

   b) Start Wireshark on PC1 and on PC4, and set the capture filter to capture ICMP and ARP packets only.

   c) Issue a `ping` command from PC1 to PC2 and PC3 for at least two sends (`-c 2`).

   d) Save the output of the `ping` command at PC1 and the output of Wireshark on PC1 and PC4.

   e) Save the ARP tables, routing tables, and routing caches of each host. Please note that these are the tables entries from Step 3 after the ping commands are issued.

4. **Issue ping commands from PC3 to PC4**

   a) Clear the ARP table on all PCs.

   b) Start Wireshark on PC3, and set the capture filter to capture ICMP and ARP packets only.

   c) Check the ARP table, routing table, and routing cache of each host. Save the output. Please note that these are the table entries from Step 4 before the ping is issued.

   d) Issue a `ping` command from PC3 to PC4 for at least three sends (`c -3`)

   e) Save the output of the `ping` command and the output of Wireshark on PC3.

   f) Save the ARP table, routing table, and routing cache of PC3. Please note that these are the tables entries from Step 4 after the ping commands are issued.

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 previous experiment to obtain the same results. You have to wait until the routing cache is reset or you can delete all the routing caches on all devices.

Lab Questions

- 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 Echo Replies, try to explain the reason.
- 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.