

ARIZONA STATE UNIVERSITY
CSE 434, SLN 70569 — Computer Networks — Fall 2021

Lab #3

Available Sunday, 10/31/2021; due electronically before 11:59pm on Sunday, 11/14/2021

This lab is on static routing, to be completed on the equipment in BYENG 217. The objectives include:

1. How to turn a computer with multiple interfaces into a router.
2. How to set up static routing on Linux PC routers and Cisco routers.
3. How ICMP messages update routing table entries.

1 Static Routing on the Racks

1.1 Configuring a Linux PC as an IP Router

Every Linux PC with at least two network interfaces can be set up as an IP router. Configuring a Linux PC as an IP router involves two steps:

1. Modifying the configuration of Linux, so that IP forwarding is enabled, and
2. configuring the routing table.

Figure 1 shows the network topology used in this part of the lab. PC A and PC D are used as hosts, and PC B and Router A are set up as IP routers. The PCs and the Cisco router are connected by three Ethernet switches. In this lab, all routing table entries will be manually configured, which is known as static routing.

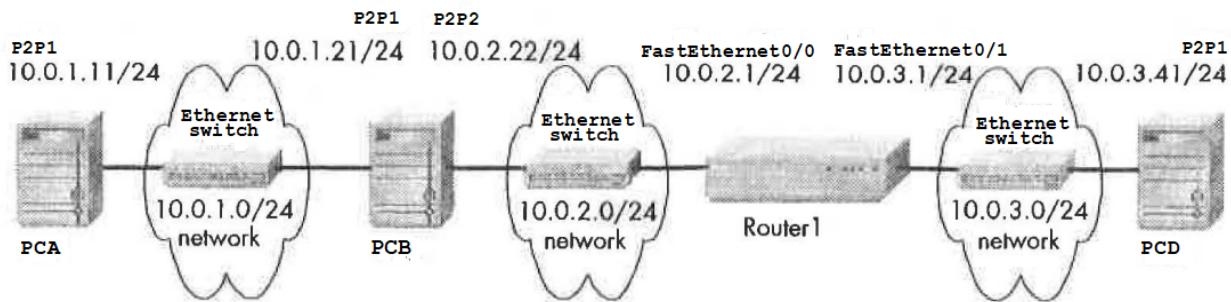


Figure 1: Network topology used in BYENG 217 for this lab.

Table 1: IP addresses for configuring topology in Figure 1.

Linux PC	Ethernet Interface <i>p2p1</i>	Ethernet Interface <i>p2p2</i>
Computer A	10.0.1.11/24	Disabled
Computer B	10.0.1.21/24	10.0.2.22/24
Computer D	10.0.3.41/24	Disabled
Cisco Router	Ethernet Interface FastEthernet0/0	Ethernet Interface FastEthernet0/1
Router A	10.0.2.1/24	10.0.3.1/24

1.1.1 Network Setup

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

2. Start to capture traffic on PC A with **wireshark**.
3. Issue a **ping** command from PC A to PC B, Router A, and PC D. Save the output of each **ping** command.


```
PCA% ping -c 5 10.0.1.21
PCA% ping -c 5 10.0.2.1
PCA% ping -c 5 10.0.3.41
```
4. Save the captured **wireshark** output.

Exercise 1.1: For the lab report, use your saved data to answer the following questions:

1. What is the output on PC A when the **ping** commands are issued?
2. Which packets, if any, are captured by **wireshark**?
3. Do you observe any ARP or ICMP packets? If so, what do they indicate?
4. Which destinations are not reachable? Explain.

1.2 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. To enable IP forwarding write a 1 in the file, with the command:

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

IP forwarding is disabled with the command:

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

The command has an immediate effect; changes are not permanent and are lost on a system reboot.

1. Enable PC B as an IP router.

1.3 Setting Static Routing Table Entries for a Linux PC

Next, you must set up the routing tables of the Linux PCs. PC A and PC D are hosts, and PC B is enabled as an IP router. Here, the routes are configured manually, which is referred to as *static routing*.

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

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

```
PCA% 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:

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

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

```
PCA% route add default gw 10.0.4.4
```

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

```
PCA% route del -net 10.21.0.0 netmask 255.255.0.0
```

```
PCA% route del -host 10.0.2.31
```

```
PCA% route del default
```

```

route add -net netaddress netmask mask gw gw_address
route add -net netaddress netmask mask dev iface
    Adds a routing table entry for the network prefix identified by IP address netaddress and netmask mask. The next-hop is identified by IP address gw_address or by interface iface.

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

route add default gw gw_address
    Sets the default route to IP address gw_address.

route del -net netaddress netmask mask gw gw_address
route del -host hostaddress gw gw_address
route del default gw gw_address
    Deletes an existing route from the routing table. It is not necessary to type all arguments. If enough arguments are provided that it can be matched with an existing routing entry, the first entry that matches the given arguments is deleted.

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

route -C
    Displays the routing table cache.

```

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.)

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

```

ping IPaddress
    Tests whether IPaddress can be reached
traceroute IPaddress
    Displays the route to the interface IPaddress

```

1. Configure the routing table entries of PC A and PC D. You can either specify a default route or insert separate routing entries for each remote network. For this exercise, add a route for each individual remote network. As a hint, here is the configuration information for PC D:
 PCD% **route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.3.1**
 PCD% **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 PC B. (The correctness of the routing entries will be tested after Router A has been set up.)
3. Display the routing table of PC A, PC B, and PC D with **netstat -rn** and save the output.

Exercise 1.2: For the lab report:

1. Include the saved output of the routing tables in step 3 in §1.3.
2. Explain the entries in the routing tables and discuss the values of the fields for each entry.

1.4 Configuring a Cisco Router

The setup of a Cisco router is more involved. The first step is to establish a physical connection to the router, so that configuration commands can be entered. There are different ways to connect to a Cisco router. In this lab, you establish a serial connection to the router. This is done with a serial cable that connects the serial port of a Linux PC to the console port of a Cisco router. The next step is to run a terminal emulation program on the Linux PC. In this lab, you use the `kermit` utility program to establish a serial connection between a Linux PC and a Cisco router. Lastly, we have to type IOS commands to configure the Cisco router. (The Cisco routers in the lab are running a recent version of the Cisco Internet Operating System (IOS).) The network setup for this part is as shown in Figure 1 and in Table 1.

1.4.1 Accessing a Cisco Router via the Console Port with kermit

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

The following steps access the console port of Router A from PC A:

1. Use a serial cable to connect the serial port of PC A to the console port of Router A. We will use the serial port labelled `ttyS1`.
2. Start `kermit` by typing:
`PCA% kermit`
3. This brings up the prompt:
`[/root]C-kermit>`
4. Use the `set line` command to select the `ttyS1` serial port.
`[/root]C-kermit> set line /dev/ttyS1`
5. Use the `set carrier-watch` command to disable the requirement for a carrier detect signal.
`[/root]C-kermit> set carrier-watch off`
6. Set the speed to 9600
`[/root]C-kermit> set speed 9600`
7. Connect to the router by issuing the following command:
`[/root]C-kermit> connect`

If the connection is successful, you see a command prompt (user EXEC prompt) from Router A:
`RouterA>`

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

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

1.4.2 Switching Cisco IOS Command Modes

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

1. Make sure that PC A is connected to Router A via a serial cable and that a `kermit` session is started.
2. When PC A is connected to the router, you see the prompt of the User EXEC mode (`RouterA>`). To see which commands are available in this mode, type a question mark:
`RouterA> ?`
3. To view and change system parameters of a Cisco router, you must enter the Privileged EXEC mode, by typing:
`RouterA> enable`

The routers in the lab are configured to go into the Privileged EXEC mode without a password.

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

```
RouterA# configure terminal  
RouterA(config)#
```

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

```
RouterA(config)# interface FastEthernet0/0  
RouterA(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:

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

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:

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

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

```
RouterA# disable  
RouterA>
```

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

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

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

1.4.3 Configuring IP Interfaces on a Cisco Router

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

1. Connect PC A to Router A via the serial cable and start a *kermit* session.
2. Configure Router A with the IP addresses given in Table 1.

```
RouterA> enable  
RouterA# configure terminal  
RouterA(config)# no ip routing  
RouterA(config)# ip routing  
RouterA(config)# interface FastEthernet0/0  
RouterA(config-if)# ip address 10.0.2.1 255.255.255.0  
RouterA(config-if)# no shutdown  
RouterA(config-if)# interface FastEthernet0/1  
RouterA(config-if)# ip address 10.0.3.1 255.255.255.0  
RouterA(config-if)# no shutdown  
RouterA(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:

```
RouterA# show interfaces  
RouterA# show running-config
```

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

Exercise 1.3: For the lab report, include the output from Step 3 of §1.4.3.
--

1.4.4 Setting Static Routing Entries on a Cisco Router

Next you must add static routes to the routing table of Router A. The routing table must be configured so that it conforms to the network topology shown in Figure 1 and Table 1.

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

IOS Mode: Privileged EXEC

`show ip route`

Displays the contents of the routing table.

`clear ip route *`

Deletes all routing table entries.

`show ip cache`

Displays the routing cache.

IOS Mode: Global Configuration

`ip route-cache`

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

`no ip route-cache`

Disables route caching.

`ip route destination mask gw_address`

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

`no ip route destination mask gw_address`

`no ip route destination mask Iface`

Deletes the routing 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. 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:

`RouterA(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:

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

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

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

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

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

`RouterA(config)# no ip route 10.21.0.0 255.255.0.0 10.11.1.4`

`RouterA(config)# no ip route 10.0.2.31 255.255.255.255 10.0.1.21`

`RouterA(config)# no ip route 0.0.0.0 0.0.0.0 10.0.4.4`

1. Display the contents of the routing table with `show ip route`. Note the routing entries that are already present. Save the output.
2. Add routing entries to Router A so that the router forwards datagrams for the configuration shown

in Figure 1. Routing entries should exist for the following networks: 10.0.1.0/24, 10.0.2.0/24, and 10.0.3.0/24.

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

Exercise 1.4: For the lab report,

1. Include the saved output of the routing table from Steps 1 and 2 of §1.4.4.
2. Explain the fields of the routing table entries of the Cisco router.
3. Explain how the routing table has changed from Step 1 to Step 3.

1.5 Finalizing and Exploring the Router Configuration

If the configuration of PC B and Router A was done correctly, it is now possible to send IP datagrams between any two machines in the network shown in Figure 1. However, if the network is not configured properly, you need to debug and test your setup. Table 2 summarizes several common problems that may arise; it is impossible to cover all scenarios.

Table 2: Troubleshooting network configurations.

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

1.5.1 Finalizing the Network Setup

Continue with the network configuration from §1.4.

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.

1.5.2 Testing Routes with traceroute

1. Start a `wireshark` session on PC A.
2. Execute a `traceroute` command from PC A to PC D and save the output.
`PCA> 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 PC A, PCB, PC D, and Router A.

Exercise 1.5: For the lab report, use the `wireshark` output and the previously saved routing table (in Exercise 1.4) to explain the operation of `traceroute`.

1.5.3 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 cache is a table that holds entries of the form <IPaddress, MAC address>. Some of the common uses of `arp` commands are listed:

Sample use cases of arp command

`arp -a`

Displays the content of the ARP cache.

`arp -d IPAddress`

Deletes the entry with the IP address IPAddress.

`arp -s IPAddress MACAddress`

Adds a static entry to the ARP cache that is never overwritten by network events. The MAC address is entered as 6 hexadecimal bytes separated by colons.

Example: `arp - s 10.0.1.12 00:02:2D:0D:68:C1`

The following list shows corresponding IOS commands for Cisco routers.

IOS Mode: Privileged EXEC

`show ip arp`

Displays the contents of the ARP cache

`clear arp`

Deletes the entire ARP cache

IOS Mode: Global Configuration

`arp IPaddress`

Adds an entry for *IPaddress* to the ARP cache

`no arp IPaddress`

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

1. Erase all ARP entries on PC A, PC B, PC D, and Router A.
2. Run `wireshark` on both PC A (interface p2p1) and PC D (interface p2p1).
3. Issue a `ping` command on PC A to PCD.
`PCA> ping -c 5 10.0.3.41`

- Save the packet transmissions triggered by the `ping` command, including ARP requests, ARP Reply, ICMP Echo Request, and ICMP Echo Reply on both PC A and PC D.

Exercise 1.6: For the lab report,

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

1.5.4 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.

- Add the following routes to the routing table of PC A:

```
PCA> route add -net 10.0.0.0 netmask 255.255.0.0 gw 10.0.1.71
PCA> 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
```

- Referring to the routing table, determine how many matches exist for the following IP addresses: 10.0.3.9, 10.0.3.14, and 10.0.4.1.

- Start an `wireshark` session on PC A and issue the following ping commands from PC A:

```
PCA> ping -c 1 10.0.3.9
PCA> ping -c 1 10.0.3.14
PCA> 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, PC A still sends ARP Request packets for these IP addresses.

- Save the output of `wireshark` and PC A's routing table.

Exercise 1.7: For the lab report,

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

1.5.5 Default Routes

- Delete the routing table entries added in Step 1 of Exercise 3(d). (Otherwise, the entries interfere with the remaining exercises in this lab.)
- Add default routes on PC A and PC B.
 - On PC A, add a default route with interface `p2p1` of PC B as the default gateway.
 - On PC B, add a default route with interface `FastEthernet0/0` of Router A as the default gateway.
- Start to capture traffic on PC A (on `p2p1`) and PC B (on both `p2p1` and `p3p1`) with `wireshark`.

4. Issue a `ping` command from PC A to a host on a network that does not exist.

```
PCA> ping -c 5 10.0.10.110
```

5. Save the *wireshark* output.

Exercise 1.8: For the lab report, use your saved data to answer the following questions:

1. What is the output on PC A when the `ping` command is issued?
2. Determine how far the ICMP Echo Request message travels?
3. Which, if any, ICMP Echo Reply message returns to PC A?

When you have finished the exercises, reconfigure the interfaces to 0.0.0.0 and remove the Ethernet cables.