Experiment 2

Router Configuration + Static Routing

Objectives

- Learn how to configure a Cisco IOS router using the IOS command-line interface (CLI).
- Learn how to use router simulator.

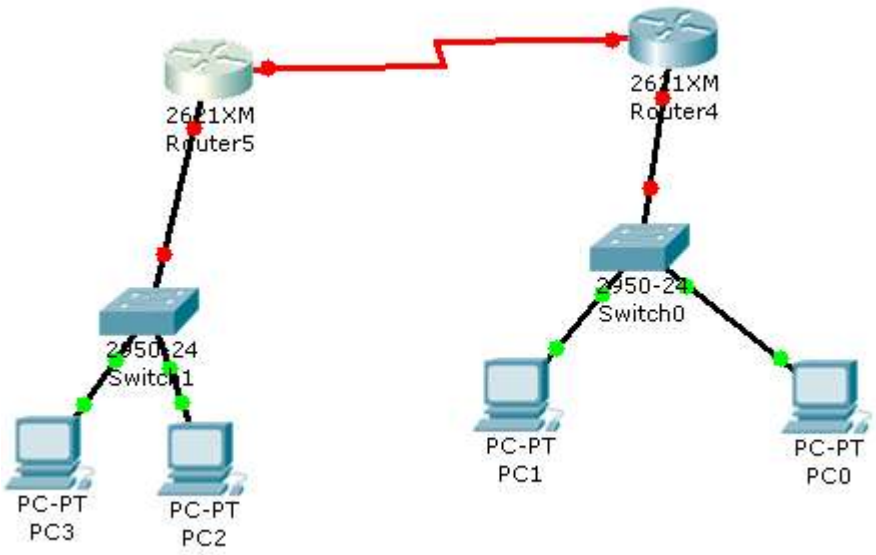
Lab setup:

- Two Cisco router
- Two PCs.
- Two CAT5 straight-wired cable
- **Background:**

In the previous experiments you have build simple (single) networks in which you used TCP/IP to enable the devices in these networks to communicate with each other. To connect more than such networks together, as in Figure 1 , we need an entity in this network that is capable to deliver data packets from one network to the correct destination network. This device is called router, its main role is to rout packets to the correct destination. Traditionally the router is called a layer-3 device, therefore it uses the IP address (layer-3 address) to build its path toward the destination. Each network is called a segment (subnet). May be the main reason for having subnets is to control the traffic. Each node in any segment can hear all packets transmitted by other nodes in the segment. Based on routing information (routing table) a router can determine the next node toward the destination. The router uses the destination IP address of the packet to find the correct path. There are two main types of routing protocols, static and dynamic. In static routing, it is the role of the administrator to update the router with new routing information (add segment or remove a segment). In Dynamic routing the routing information will be updated automatically.

Pre-lab:

1. Compare between static and dynamic routing with respect to router configuration, flexibility, performance, and security.
2. Why do we need subnetting? Discuss three reasons?
3. Install at home a simulator from CISCO called **Packet Tracer** , build the network in Figure 1 , give PC0 , PC1 , PC2 , PC3 Ip addresses , show that (without configuring the router) it is possible to send a packet from PC0 to PC1 , but it is not possible to send from PC0 to PC2. (Use ping command and print screen to show the results)
4. use static routing to allow the connectivity between the two parts of the network. You must show me a demo at the beginning of the lab.



Cisco Routers:

A. The Cisco Router User Interface

The *Cisco Internetwork Operating System (IOS)* is the kernel of Cisco routers and most switches.

A kernel is the basic part of an operating system that allocates resources and manages things such as low-level hardware interfaces and security.

The Cisco IOS was created to deliver network services and enable networked applications. It runs on most Cisco routers and on some Cisco Catalyst switches, such as the Catalyst 2950. These are some of the important things the Cisco router IOS software is responsible for:

- Carrying network protocols and functions
- Connecting high-speed traffic between devices
- Adding security to control access and stop unauthorized network use
- Providing scalability for ease of network growth and redundancy
- Supplying network reliability for connecting to network resources

B. Connecting to a Cisco Router

There are different ways to connect to a Cisco router to configure it, verify its configuration, and check statistics.

1. The console port.

The *console port* is usually an RJ-45 (8-pin Modular) connection located at the back of the router—by default, there's no password set.

2. Auxiliary port

You can also connect to a Cisco router through an **auxiliary port** which is really the same thing as a console port, so it follows that you can use it as one. But this auxiliary port also allows you to configure modem commands so that a modem can be connected to the router. This is a cool Feature—

it lets you dial up a remote router and attach to the auxiliary port if the router is down and you need to configure it "out-of-band" (which means, basically, "out-of-the-network").

3. Telnet

The third way to connect to a Cisco router is in-band, through the program, Telnet is a terminal emulation program that acts as though it's a dumb terminal. You can use Telnet to connect to any active interface on a router like an Ethernet or serial port.

Procedure:

1. Connecting to a Cisco Router

Once you have the correct cable connected from your PC to the Cisco router or switch, you can start HyperTerminal to create a console connection and configure the device. Set the configuration as follows:

1) From start menu select**: programs> accessories >communication> HyperTerminal**.
2) Open HyperTerminal and enter a name for the connection. It is irrelevant what you name it, then click OK.
3) Choose the communications port—either COM1 or COM2, whichever is open on your PC.
4) Now set the port settings. The default values (2400bps and no flow control) will not work; Notice that the bit rate is now set to 9600 and the flow control is set to none. At this point, you can click OK and press the Enter key, and you should be connected to your Cisco device console port.

2. Setup Mode and Command Line interface mode(CLI)

A router initializes by loading the bootstrap, the operating system, and a configuration file. If the router cannot find a configuration file, then it enters setup mode. The router stores, in NVRAM, a backup copy of the new configuration from setup mode.

- The goal of the startup routines for Cisco IOS software is to start the router operations. The router must deliver reliable performance in its job of connecting the user networks it was configured to serve.
- To exit setup mode press ctrl+Z

Because it's so much more flexible, the *command-line interface (CLI)* truly is the best way to configure a router. Using CLI you can create advanced configurations on Cisco routers and switches. To use the CLI, just say No to entering the initial configuration dialog. After you do that, the router will respond with messages that tell you all about the status of each and every one of the router's interfaces. Here's an example:

Would you like to enter the initial configuration dialog?

[yes]:n

Would you like to terminate autoinstall? [yes]:[Enter]

Press RETURN to get started!

3. Logging into the Router

After the interface status messages appear and you press Enter, the

Router>

prompt will appear. This is called *user exec mode* (user mode) and is mostly used to view statistics, but it's also a stepping-stone to logging into privileged mode. You can only view and change the configuration of a Cisco router in *privileged exec mode* (privileged mode), which you get into with the enable command.

Here's how you would do that:

Router>

*Router>**enable***

Router#

You now end up with a Router# prompt, which indicates you're in *privileged mode*, where you can both view and change the router's configuration. You can go back from privileged mode into user mode by using the disable command, as seen here:

*Router#**disable***

Router>

At this point, you can type **logout** to exit the console:

*Router>**logout***

Or you could just type **logout** or **exit** from the privileged-mode prompt to log out:

*Router>**en***

*Router#**logout***

4. Editing and Help Features

You can use the Cisco advanced editing features to help you configure your router. If you type in a question mark (**?**) at any prompt.

Here's a shortcut: To find commands that start with a certain letter, use the letter and the question mark with no space between them:

Router#**c?**

clear clock configure connect copy

Router#**c**

By typing **c?**, we received a response listing all the commands that start with *c*.

To find the next command in a string, type the first command and then a question mark:

Router#**clock ?**

set Set the time and date

Router#**clock set ?**

hh:mm:ss Current Time

Router#**clock set 10:30:10 ?**

<1-31> Day of the month

MONTH Month of the year

Router#**clock set 10:30:10 28 ?**

MONTH Month of the year

Router#**clock set 10:30:10 28 august ?**

<1993-2035> Year

Router#**clock set 10:30:10 28 august 2003 ?**

<cr>

Router#

By typing the **clock ?** command, you'll get a list of the next possible parameters and what they do. Notice that you should just keep typing a command, a space, and then a question mark until <cr> (carriage return) is your only option.

Command	Meaning
Ctrl+P or up arrow	Shows last command entered
Ctrl+N or down arrow	Shows previous commands entered
show history	Shows last 10 commands entered by default
Ctrl+A	Moves your cursor to the beginning of the line
Ctrl+E	Moves your cursor to the end of the line
Esc+B	Moves back one word
Ctrl+F	Moves forward one character
Esc+F	Moves forward one word
Ctrl+B	Moves back one character
Ctrl+D	Deletes a single character
Backspace	Deletes a single character
Ctrl+R	Redisplays a line
Ctrl+U	Erases a line
Ctrl+W	Erases a word
Ctrl+Z	Ends configuration mode and returns to EXEC
Tab	Finishes typing a command for you

Table 1 Enhanced Editing Command

5. Overview of Router Modes

To configure from a CLI, you can make global changes to the router by typing configure terminal (or **config t** for short), which puts you in global configuration mode and changes what's known as the running-config. A global command (a command run from global config) is one that is set once and affects the entire router.

You can type **config** from the privileged-mode prompt and then just press Enter to take the default of terminal, as seen here:

Router#**config**

Router(config)#

At this point, you make changes that affect the router as a whole, hence the term global configuration mode.

To change the running-config—the current configuration running in dynamic RAM (DRAM) you use the configure terminal.

6. Gathering Basic Routing Information

The show version command will provide basic configuration for the system hardware as well as the software version, the names and sources of configuration files, and the boot images. Here is an example:

Router#**sh version**

7. Router Interfaces

To make changes to an interface, you use the interface command from global configuration mode:

Router(config)#**interface ?**

Router(config)#**interface fastethernet 0/0**

Router(config-if)#

Router(config)#**interface s0**

Router(config-if)#

Interface configuration is one of the most important router configurations, because without interfaces, a router is a totally useless thing. Plus, interface configurations must be exact to enable communication with other devices. Some of the configurations used to configure an interface are Network layer addresses, media type, bandwidth, and other administrator commands.

Different routers use different methods to choose the interfaces used on them. For instance, the following command shows a Cisco 2522 router with 10 serial interfaces,

labeled 0 through 9:

Router(config)#**int serial ?**

<0-9> Serial interface number

Now it's time to choose the interface you want to configure. Once you do that, you will be in interface configuration for that specific interface. The command to choose serial port 5, for

xample, would be:

Router(config)#**int serial 5**

Router(config)-if)#

The 2522 router has one Ethernet 10BaseT port, and typing **interface ethernet 0/0** can

configure that interface, as seen here:

Router(config)#**int ethernet ?**

<0-0> Ethernet interface number

Router(config)#**int ethernet 0**

Router(config-if)#

The 2500 router, as previously demonstrated, is a fixed configuration router, which means that when you buy that model, you're stuck with that physical configuration. To configure an interface, you always use the interface *type number* sequence, but the 2600, 3600, 4000, and 7000 series routers use a physical slot in the router, with a port number on the module plugged into that slot. So on a 2600 router, the configuration would be

interface *type slot/port*, as seen here:

Router(config)#**int fastethernet ?**

<0-1> FastEthernet interface number

Router(config)#**int fastethernet 0**

% Incomplete command.

Router(config)#**int fastethernet 0?**

/

Router(config)#**int fastethernet 0/?**

<0-1> FastEthernet interface number

8. Bringing Up an Interface

You can turn an interface off with the interface command shutdown, and turn it on with the no shutdown command.

Router(config)#**int ethernet0**

Router(config-if)#**no shutdown**

Router(config-if)#**^Z**

To show interface **configuration:**

Router#**sh int ethernet0**

9. Configuring an IP Address on an Interface

Even though you don't have to use IP on your routers, it's most often what people use. To configure IP addresses on an interface, use the ip address command from interface configuration mode:

Router(config)#**int e0**

Router(config-if)#**ip address 172.16.10.2 255.255.255.0**

Router(config-if)#**no shut**

You can verify that both addresses are configured on the interface with the show running-config

Router#**sh run**

10. Serial Interface Commands

Before you jump in and configure a serial interface, there are a couple of things you need to know. First, the interface will usually be attached to a CSU/DSU type of device that provides clocking for the line to the router. But if you have a back-to-back configuration (for example, one that's used in a lab environment), one end—the data communication equipment (DCE) end of the cable—must provide clocking. By default, Cisco routers are all data terminal equipment (DTE) devices, so you must tell an interface to provide clocking if you need it to act like a DCE device.

You configure a DCE serial interface with the clock rate command:

Router#**config t**

Router(config)#**int s1**

Router(config-if)#**clock rate ?**

Router(config-if)#**clock rate 64000**

Here is an example of using the bandwidth command:

Router(config-if)#**bandwidth ?**

<1-10000000> Bandwidth in kilobits

Router(config-if)#**bandwidth 64**

Did you notice that, unlike the clock rate command, the bandwidth command is configured in kilobits?

11. Hostnames

You can set the identity of the router with the hostname command. This is only locally significant, which means it has no bearing on how the router performs name lookups or how the router works on the internetwork.

Here is an example:

Router#**config t**

Router(config)#**hostname RouterA**

RouterA (config)#**hostname RouterB**

RouterB (config)#

Even though it's pretty tempting to configure the hostname after your own name, it's a better idea to name the router something pertinent to the location.

12. Passwords

You can secure your system by using passwords to restrict access. Passwords can be established both on individual lines and in the privileged EXEC mode.

- **line console 0** -- establishes a password on the console terminal
- **line vty 0 4** -- establishes password protection on incoming Telnet sessions
- **enable password** -- restricts access to privileged EXEC mode
- **enable secret** password (from the system configuration dialog to set up global parameter uses a Cisco proprietary encryption process to alter the password character string

1. Set your enable secret password by typing **enable secret *cisco*** (the third word cisco should be your own personalized password) and pressing Enter.

2. Now let's see what happens when you log all the way out of the router and then log in. Log out by pressing Ctrl+Z, then type **exit** and press Enter. Go to privileged mode. Before you are allowed to enter privileged mode, you will be asked for a password. If you successfully enter the secret password, you can proceed.

3. Remove the secret password. Go to privileged mode, type **config t**, and press Enter. Type **no enable secret** and press Enter. Log out and then log back in again, and now you should not be asked for a password.

4. To set the Telnet or VTY password, type **line vty 0 4** and then press Enter. The 0 4 is the range of the five available virtual lines used to connect with Telnet. If you have an enterprise IOS, the number of lines may vary. Use the question mark to determine the last line number available on your router.

5. One more command you need to set for your VTY password is password. Type **password** *cisco* to set the password. (*cisco* is your password.)

6. Here is an example of how to set the VTY passwords:

config t

line vty 0 4

login

password cisco

7. Set your console password by first typing **line console 0** or **line con 0**.

13. Viewing and Saving Configurations

If you run through setup mode, you'll be asked if you want to use the configuration you just created. If you say Yes, then it will copy the configuration running in DRAM, (known as the running-config), into NVRAM, and name the file startup-config.

You can manually save the file from DRAM to NVRAM by using the copy runningconfig startup-config command (you can use the shortcut copy run start also):

Atlanta#**copy run start**

14. Verifying Your Configuration

Obviously, show running-config would be the best way to verify your configuration, and show startup-config would be the best way to verify the configuration that'll be used the next time the router is reloaded—right?

Well, once you take a look at the running-config, if all appears well, you can verify your configuration with utilities such as Ping and Telnet. Ping sends a packet to a remote host, and if that host responds, you know that the host is alive. But you don't know if it's alive and also *well*—just because you can ping an NT server does not mean you can log in! Even so, Ping is an awesome starting point for troubleshooting an internetwork. Did you know that you can ping with different protocols? You can, and you can test this by typing **ping ?** at either the router user-mode or privileged mode prompt:

Router#**ping ?**

15. Verifying with the *show ip interface* Command

The show ip interface command will provide you with information regarding the layer 3 configurations of a router's interfaces:

Router#**sh ip interface**

The status of the interface, the IP address and mask, information on whether an access list is set on the interface, and basic IP information are included in this output.

16.Using the *show ip interface brief* Command

The show ip interface brief command is probably one of the most helpful commands that you can ever use on a Cisco router. This command provides a quick overview of the router's interfaces including the logical address and status:

Router#**sh ip int brief**

Practice in the Lab:

Connect the network as in the figure (2).

1. Access the console of the router and:
    - Set Lab _A router name to R2500. and set Lab_B router name to R2600
    - Set enable password to R2500 and R2600 to **cisco.**

- Set Telnet password to R2500 and R2600 to **cisco**.

2. Configure the host_A Ethernet interface with the following parameters:

   IP: 172.16.10.2 SN Mask: 255.255.0.0 Gateway: 172.16.10.1

3. Configure the host_B Ethernet interface with the following parameters:

   IP: 172.16.20.2 SN Mask: 255.255.0.0 Gateway: 172.16.20.1

4. Ping from host_A to F 0/0 on Lab_A.
4. Ping from host_B to F 0/0 on Lab_B.
5. Ping from Lab_B to serial 0 on Lab_A.
6. Try a telnet from host_B to F 0/0 on Lab_A.
7. Ping from host_B to host_A, what is the result Explain?

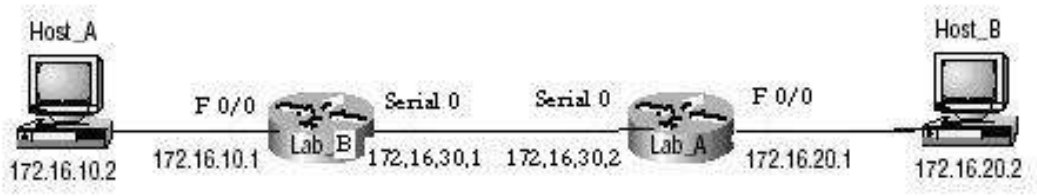


Figure 2: Simple Network practice

Exercise1 in the home:

A. Using Packet Tracer, build the network in the last practice and simulate it.

B. Write and check out the command or commands for the following questions:

1) What command is used to set a serial interface to provide clocking to another router at 64k?
2) If you telnet into a router and get the response "connection refused, password not set," What would you do on the destination router to stop receiving this message and not be prompted for a password?
3) If you type **show int e0** and notice the port is administratively down, what would you do?
4) If you wanted to delete the configuration stored in NVRAM, what would you type?
5) If you wanted to set a user-mode password for the console port, what would you type?
6) If you wanted to set the enable secret password to *lab1*, what would you type?
7) If you wanted to see if a serial interface needed to provide clocking, what command would use?
8) What command would you use to see the terminal history size?
9) How would you set the name of a router to *Birzeit*?

Routing Basics

Once you create an internetwork by connecting your WANs and LANs to a router, you'll need to configure logical network addresses, such as IP addresses, to all hosts on the internetwork so that they can communicate across that internetwork.

The term *routing* is used for taking a packet from one device and sending it through the network to another device on a different network. Routers don't really care about hosts—they only host is used to

get packets to a network through a routed network, then the hardware address of the host is used to deliver the packet from a router to the correct destination host.

If your network has no routers, then it should be apparent that you are not routing. Routers route traffic to all the networks in your internetwork. To be able to route packets, a router must know, at a minimum, the following:

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

Static Routing & Dynamic Routing

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table that describes how to find the remote networks. If a network is directly connected, then the router already knows how to get to it. If a network isn't connected, the router must learn how to get to the remote network in two ways: by using *static routing*, meaning that someone must hand-type all network locations into the routing table or through something called *dynamic routing.*

In *dynamic routing*, a protocol on one router communicates with the same protocol running on neighbor routers. The routers then update each other about all the networks they know about and place this information into the routing table. If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the event. If *static routing* is used, the administrator is responsible for updating all changes by hand into all routers. Typically, in a large network, a combination of both dynamic and static routing is used.

Static Routing

Static routing occurs when you manually add routes in each router's routing table.

Static routing has the following benefits:

- There is no overhead on the router CPU, which means you could possibly buy a cheaper router than if you were using dynamic routing.
- There is no bandwidth usage between routers, which means you could possibly save money on WAN links.
- It adds security, because the administrator can choose to allow routing access to certain networks only.

Static routing has the following disadvantages:

- The administrator must really understand the internetwork and how each router is connected in order to configure routes correctly.
- If a network is added to the internetwork, the administrator has to add a route to it on all routers—by hand.
- It's not feasible in large networks because maintaining it would be a full-time job in itself.

Procedure:

IP Routing in a Larger Network

In the exercise gave you in the previous lab, the router already has both IP networks in its routing table because the networks are directly connected to it. But what if we add another router? Figure 3 shows three routers: Lab_A, Lab_B, and Lab_C. Remember—by default, these routers only know about networks that are directly connected to them.

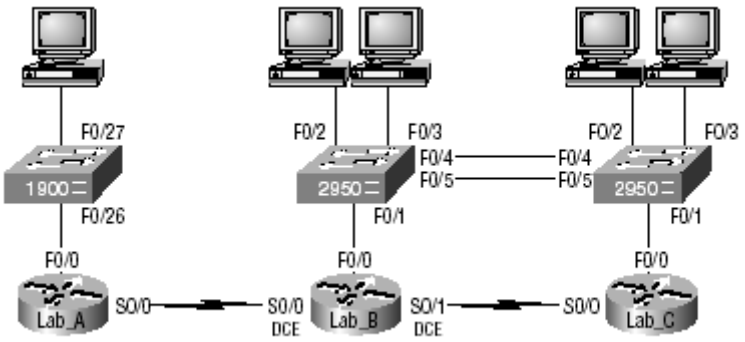


Figure 3: IP routing example with more routers

Figure 3 shows the three routers connected via a WAN. Each router also has an Ethernet network connected. The idea is that each router must know about all five networks. The first step is to configure each router correctly. Table 2 shows the IP address scheme. Each network in the following table has a default Class C 24-bit subnet mask (255.255.255.0).

Router	Network Address	Interface	Address
Lab_A	192.168.10.0	fa0/0	192.168.10.1
Lab_A	192.168.20.0	s0/0	192.168.20.1
Lab_B	192.168.20.0	s0/0	192.168.20.2
Lab_B	192.168.40.0	s0/1	192.168.40.1
Lab_B	192.168.30.0	fa0/0	192.168.30.1
Lab_C	192.168.40.0	s0/0	192.168.40.2
Lab_C	192.168.50.0	fa0/0	192.168.50.1

Table 2 Network Addressing for the IP Network

Firstly we go to use to configure the network. After we go over how the network is configured, how to configure IP routing. Router configuration is really a pretty straightforward process, since you just need to add IP addresses to your interfaces and then perform a no shutdown on those interfaces.

Lab_A Configuration

To configure the Lab_A router, you just need to add an IP address to interface FastEthernet 0/ 0 as well as the serial0/0. Configuring the hostnames of each router will make identification easier. And

why not set the interface descriptions, banner, and router passwords, too? You really should get in the habit of configuring these commands on every router.

Here is how you do all that:

Router>**en**

Router#**config t**

Router(config)#**hostname Lab_A**

Lab_A(config)#**enable secret todd**

Lab_A(config)#**interface fa0/0**

Lab_A(config-if)#**ip address 192.168.10.1 255.255.255.0**

Lab_A(config-if)#**description Lab_A LAN Connection**

Lab_A(config-if)#**no shut**

Lab_A(config-if)#**interface serial 0/0**

Lab_A(config-if)#**ip address 192.168.20.1 255.255.255.0**

Lab_A(config-if)#**description WAN Connection to Lab_B**

Lab_A(config-if)#**no shut**

Lab_A(config-if)#**exit**

Lab_A(config)#**line console 0**

Lab_A(config-line)#**password todd**

Lab_A(config-line)#**login**

Lab_A(config-line)#**line aux 0**

Lab_A(config-line)#**password todd**

Lab_A(config-line)#**login**

Lab_A(config-line)#**line vty 0 4**

Lab_A(config-line)#**password todd**

Lab_A(config-line)#**login**

Lab_A(config-line)#**exit**

Lab_A(config)#**banner motd #**

This is the Lab_A router

#

Lab_A(config)#**^z**

Lab_A#**copy running-config startup-config**

Destination filename [startup-config]? ***[Enter]***

Lab_A#

If you have a hard time understanding this configuration process, refer back to To view the IP routing tables created on a Cisco router, use the command show ip route. The command output is shown as follows:

Lab_A#**sh ip route**

Codes: C - connected, S - static, I - IGRP, R - RIP,

M - mobile, B – BGP D - EIGRP, EX - EIGRP external, O -

OSPF, IA - OSPF inter area N1 - OSPF NSS external type
1, N2 - OSPF NSSA external type 2 E1 - OSPF external
type 1, E2 - OSPF external type 2, E – EGP i - IS-IS,
L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default, U - per-user static route, o - ODR, P -
periodic downloaded static route, T - traffic
engineered route
Gateway of last resort is not set
C 192.168.10.0/24 is directly connected, FastEthernet0/0
C 192.168.20.0/24 is directly connected, Serial 0/0

Lab_A#

Notice that only the configured, directly connected networks are shown in the routing table. This means that the router only knows how to get to networks 192.168.10.0 and 192.168.20.0. Did you notice the C? When you see it there, it means that the network is directly connected. The codes for each type of connection are listed at the top of the show ip route command with their abbreviations.

Lab_B Configuration

Do the Lab_B configuration in the same way of Lab_A configuration.

Lab_C Configuration

Do the Lab_C configuration in the same way of Lab_A configuration.

Static Routing

The command syntax you use to add a static route to a routing table:

ip route [destination_network] [mask] [next-hop_address or exitinterface]
[administrative_distance] [permanent]

This list describes each command in the string:

ip route The command used to create the static route.

destination_network The network you're placing in the routing table.

mask The subnet mask being used on the network.

next-hop_address The address of the next-hop router that will receive the packet and forward it to the remote network. This is a router interface that's on a directly connected network.

You must be able to ping the router interface before you add the route. If you type in the wrong next-hop address, or the interface to that router is down, the static route will show up in the router's configuration, but not in the routing table.

exitinterface You can use it in place of the next-hop address if you want, but it's got to be on a point-to-point link, such as a WAN. This command won't work on a LAN such as Ethernet.

administrative_distance By default, static routes have an administrative distance of 1 (or even 0 is you use an exit interface instead of a next hop address). You can change the default value by adding an administrative weight at the end of the command. I'll talk a lot more about this subject when we get to the Lab on dynamic routing.

permanent If the interface is shut down, or the router can't communicate to the next-hop router, the route will automatically be discarded from the routing table. Choosing the permanent option keeps the entry in the routing table no matter what happens.

To understand how static routes work, we will demonstrate the configuration on the internetwork shown previously in Figure 3.

Lab_A

Each routing table automatically includes directly connected networks. To be able to route to all networks in the internetwork, the routing table must include information that describes where these other networks are located and how to get there.

The Lab_A router is connected to networks 192.168.10.0 and 192.168.20.0. For the Lab_ A router to be able to route to all networks, the following networks have to be configured in its routing table:

- 192.168.30.0
- 192.168.40.0
- 192.168.50.0

The following router output shows the configuration of static routes on the Lab_A router and the routing table after the configuration. For the Lab_A router to find the remote networks, an entry is placed in the routing table describing the network, the mask, and where to send the packets. Notice that each static route sends the packets to 192.168.20.2, which is the Lab_A router's next hop.

Lab_A(config)#**ip route 192.168.30.0 255.255.255.0**
192.168.20.2
Lab_A(config)#**ip route 192.168.40.0 255.255.255.0**
192.168.20.2
Lab_A(config)#**ip route 192.168.50.0 255.255.255.0**
192.168.20.2

After the router is configured, you can type **show running-config** and **show ip route** to see the static routes:

Lab_A#**sh ip route**
[output cut]
S 192.168.50.0 [1/0] via 192.168.20.2
S 192.168.40.0 [1/0] via 192.168.20.2
S 192.168.30.0 [1/0] via 192.168.20.2
C 192.168.20.0 is directly connected, Serial 0/0
C 192.168.10.0 is directly connected, FastEthernet0/0
Lab_A#

Remember that if the routes don't appear in the routing table, it's because the router cannot communicate with the next-hop address you configured. You can use the permanent parameter to keep the route in the routing table even if the next-hop device can't be contacted.

The S in the routing table entries above means that the network is a static entry. The [1/0] is the administrative distance and metric, which I'll discuss below, to the remote network. Here the next hop interface is 0, indicating that it's directly connected.

The Lab_A router now has all the information it needs to communicate with the other remote networks. However, if the Lab_B and Lab_C routers are not configured with all the same information, the packets will be discarded at Lab_B and at Lab_C. We need to fix this with static routes.

Lab_B

The Lab_B router is connected to the networks 192.168.20.0, 192.168.30.0, and 192.168.40.0.

The following static routes must be configured on the Lab_B router:

- 192.168.10.0
- 192.168.50.0

Do the configuration for the Lab_B router.

Lab_C

The Lab_C router is directly connected to networks 192.168.40.0 and 192.168.50.0. Three routes need to be added:

- 192.168.30.0
- 192.168.20.0
- 192.168.10.0

Do the configuration for the Lab_C router.

6. Verifying Your Configuration

Once all the routers' routing tables are configured, they need to be verified. The best way to do this, besides using the show ip route command, is with the Ping program. By pinging from routers Lab_A and Lab_C, the whole internetwork will be tested end-to-end.

Really, the best test would be to use the Telnet program from one host to another.

Ping from Lab_A and Lab_C?

Telnet from Lab_A to Lab_C?