Stand-Alone Labs

Lab 1: Connecting to a Router

Objective: Become familiar with the Cisco Router.

Lab Equipment: Router 1 from the **eRouters** menu

1. When the lab has finished loading, the Router 1 window will open, and the text "Press Enter to Start" will appear.
2. Click inside the Router 1 window, and press the ENTER key to get started. You are now connected to Router 1 and are at the user mode prompt. The prompt is broken into two parts: the host name and the mode. **Router** is Router 1's host name, and the **>** prompt indicates user mode.
   Press Enter to get Started
   Router>
3. Next, type the **enable** command to get to the privileged mode prompt.
   Router>enable
   Router#
4. To return to user mode, simply type **disable**. From user mode, type **logout** or **exit** to exit the router.
   Router#disable
   Router>
   Router>exit
   Router con0 is now available
   Press RETURN to get started

Lab 2: Introduction to the Basic User Interface

Objective: Become familiar with the command-line interface (CLI), user and privileged mode, and basic **help** and **show** commands.

Lab Equipment: Router 1 from the **eRouters** menu

1. Press the ENTER key to get to the router prompt.
   Router>
2. The interface is now in user mode. At the user mode prompt, type the command that is used to view all the commands available in user mode.
   Router>?
3. Type the command used to enter privileged mode.
   Router>enable
   Router#
4. Type the command that will allow you to view the available commands in privileged mode.
   Router#?

5. Type the command that will allow you to see all of the **show** commands.
   Router#show ?
6. Type the command that will allow you to see the active, or running, configuration.
   Router#show running-config
7. At the **MORE** prompt, press the SPACEBAR key to view the next page of information.
   SPACEBAR
8. Finally, type one of the commands that will log you out of the router.
   Router#exit
   OR
   Router#disable

Lab 3: Introduction to the Basic Show Commands

Objective: Become familiar with the **basic show** commands.

Lab Equipment: Router 1 from the **eRouters** menu

1. Press ENTER to get to the router prompt.
   Router>
2. Enter privileged mode.
   Router>enable
   Router#
3. Display the active configuration in memory. The currently active configuration script running on the router is referred to as the **running-config** in the router's command-line interface (CLI). Note that privileged mode is required in order to access the running configuration. The running configuration script is not automatically saved on a Cisco router and will be lost in the event of power failure. The running configuration must be manually saved with the **copy** command (discussed in a later lab).
   Router#show running-config
4. Display flash memory. Flash memory is a special kind of memory that contains the operating system image file(s) on the router. Unlike regular router memory, flash memory continues to maintain the file image even after power is lost.
   Router#show flash
5. By default, the router's CLI maintains in memory the last 10 commands entered. The **show history** command displays simultaneously all of the past commands still in router memory.
   Router#show history
6. Press the CTRL+P key combination to retrieve the previous command you typed.
7. Press the DOWN ARROW key or press the CTRL+N key combination to see the next command in the history buffer.
8. Use the **show protocols** command to view the status of the current Layer 3 routed protocols running on your router.
   Router#show protocols

9. The **show version** command is used to obtain critical information, such as router platform type, operating system revision, operating system last boot time and file location, amount of memory, number of interfaces, and configuration register.
   Router#show version
10. Use the **show clock** command to view the router's clock.
    Router#show clock
11. The **show hosts** command displays a cached list of hosts and all of their interfaces' IP addresses.
    Router#show hosts
12. Use the **show users** command to view a list of all users who are connected to the router.
    Router#show users
13. The **show interfaces** command displays detailed information about each interface.
    Router#show interfaces
14. The **show protocols** command displays the global and interface-specific status of any Layer 3 protocols.
    Router#show protocols

Lab 4: CDP

Objective: Learn how the Cisco Discovery Protocol (CDP) functions and what is required for Cisco devices to be discovered.

Lab Equipment: Router 1 and Router 4 from the **eRouters** menu

1. On Router 1, enter global configuration mode.
   Router>enable
   Router#conf t
   Router(config)#
2. Change the host name to **R1.**
   Router(config)#hostname R1
   R1(config)#
3. Connect to Router 4, and change the host name to **R4**.
   Router>enable
   Router#conf t
   Router(config)#hostname R4
   R4(config)#
4. Return to R1, and enable the serial 0 interface. By default, all interfaces are shut down (disabled).
   R1(config)#interface serial 0
   R1(config-if)#no shutdown
5. Now, enable the serial 0 interface on R4.
   R4(config)#interface serial 0
   R4(config-if)#no shutdown

6. Enable the Ethernet 0 interface on R1.
   R1(config)#interface Ethernet 0
   R1(config-if)#no shutdown
7. CDP allows devices to share basic configuration information and will operate without any protocol-specific information being configured. CDP, which is enabled by default on all interfaces, is a Data Link protocol that operates at Layer 2 of the OSI model. This is important to understand because CDP is not routable; it can only travel to directly connected devices.

   On R1, type the command that displays the status of all interfaces that are running CDP.
   R1(config-if)#exit
   R1(config)#exit
   R1#show cdp interface

   The sample output below shows that both interfaces are up and sending CDP packets:
   Serial0 is up, line protocol is up
   Encapsulation HDLC
   Sending CDP packets every 60 seconds
   Holdtime is 180 seconds
   <output omitted>
   R1#

   Now that the router has interfaces that are broadcasting and receiving CDP updates, you can use CDP to find out about directly connected neighbors.
8. On R1, type the command that provides information about directly connected neighbors.
   R1#show cdp neighbors

   Below is some sample output:
   Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
   S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Interface	Holdtime	Capability	Platform	Port ID
R4	Serial 0	148	R	1700	Serial 0

   R1#

   The first device on the directly connected neighbors list for R1 is R4 via the serial 0 link. R1 is receiving CDP updates from R4; the updates tell R1 to retain the information for a specified amount of time. At the time this command was entered, there were 148 seconds left in the hold time for R1's update. If that time expires before another update is received, R1's information will be removed from the table. R4 is a 1000 series router, as indicated in the **Platform** column. The final column, **Port ID**, indicates the port on the other device from which the updates are being sent.
9. On R1, type the command that provides more detailed information about directly connected neighbors.
   R1#show cdp neighbor detail

Below is some sample output:
Device ID: R4
Entry address(es):
Platform: cisco 2501, Capabilities: Router
Interface: Serial0, Port ID (outgoing port): Serial0
Holdtime : 162 sec

Version:
Cisco Internetwork Operating System Software
Software, Version 12.0(16), RELEASE SOFTWARE (fc2)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Fri 02-Mar-01 17:34 by dchih

The **show cdp neighbor detail** command shows devices one at a time. It is used to display Network layer address information. The command also displays IOS version information. Notice that the devices are listed in order. If you wanted to find out information about a device further down the list, you would need to scroll down using the SPACEBAR.

10. On R1, type the command to provide information about the specific device R4.
R1#show cdp entry R4

Below is some sample output:
Device ID: R4
Entry address(es):
Platform: cisco 1000, Capabilities: Router
Interface: Serial0, Port ID (outgoing port): Serial0
Holdtime : 148 sec

Version:
Cisco Internetwork Operating System Software
Software, Version 12.0(16), RELEASE SOFTWARE (fc2)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Fri 02-Mar-01 17:34 by dchih

R1#

The **show cdp entry** command provides the same information as the **show cdp neighbor detail** command, but it allows a single device to be specified. *Also, notice that this is one of the only case-sensitive commands that exist.*

11. On R1, type the command that shows how often CDP updates are being sent and how long a recipient should retain the update.
R1#show cdp

Below is some sample output:
Global CDP information:
Sending CDP packets every 60 seconds
Sending a holdtime value of 180 seconds
Sending CDPv2 advertisements is enabled
On R1, adjust the number of seconds between CDP updates to 45.
R1#conf t
R1(config)#cdp timer 45

Besides the update interval, the holdtime value may also be adjusted. This value tells the recipient of the update how long to retain the CDP information in the update. It is also a global parameter.

12. On R1, type the command to adjust the holddown timer to 60 seconds.
    R1#conf t
    R1(config)#cdp holdtime 60
13. On R1, type the command that will allow you to verify that the changes have been made.
    R1#show cdp

    Below is some sample output:
    R1#sh cdp
    Global CDP information:
    Sending CDP packets every 45 seconds
    Sending a holdtime value of 60 seconds
    Sending CDPv2 advertisements is enabled
    R1#
14. If there are no other directly connected Cisco devices on the network, or if you want to conserve bandwidth, you can disable CDP.

    On R1, type the command that disables CDP for the entire router.
    R1#conf t
    R1(config)#no cdp run

    At times, you may wish to disable CDP for a specific interface for security reasons, or simply because the interface has very low bandwidth.
15. On R1, type the command that turns CDP back on for the entire router.
    R1#conf t
    R1(config)#cdp run
16. On R1, disable CDP for only the specific interface Ethernet 0.
    R1(config)#interface Ethernet 0
    R1(config-if)#no cdp enable
17. On R1, verify that Ethernet 0 is no longer sending CDP updates. (If the Ethernet 0 interface does not show up as an entry in the output, you can conclude that it is not sending CDP updates.)
    R1#show cdp interface

Below is sample output from the command:
R1#show cdp interface
Serial0 is up, line protocol is up
Encapsulation HDLC
Sending CDP packets every 45 seconds
Holdtime is 60 seconds

Lab 5: Extended Basics

Objective: View and configure some basic areas of the router.

Lab Equipment: Router 1 from the **eRouters** menu

1. Press ENTER to get to the router prompt.
   Router>
2. Enter the command that is used to view all the commands available in user mode.
   Router>?
3. Enter privileged mode. This is the mode that gives you complete control of the router.
   Router>enable
   Router#
4. View the commands available in privileged mode.
   Router#?
5. Enter the command that provides access to global configuration mode.
   Router#config terminal
   Router(config)#
6. The router's host name is used for local identification. When you log on to the router, you see its host name in front of the prompt (either the **>** or the **#** prompt). The host name can be used to identify the location or function of the router. Set the router's host name to **Krang**.
   Router(config)#hostname Krang
   Krang(config)#
7. The enable password controls access to privileged mode. This is a very important password because, when it is configured, only those who know the password can make configuration changes in privileged mode. Set the enable password to **boson**.
   Krang(config)#enable password boson
8. Test the password. Exit the router, and try to enter privileged mode. Notice that you have to provide the password in order to enter privileged mode. Now, type the **conf term** command and proceed with the instructions in the next step.
   Krang(config)#exit
   Krang#exit
   Krang>enable
   Password:
   Krang#config term
   Krang(config)#

9. The only problem with the enable password is that it appears in plain text in the router's configuration file. If you need to obtain assistance in troubleshooting a problem, you may inadvertently compromise the security of your system by revealing the password. Set the enable secret password to **cisco**.
   Krang(config)#enable secret cisco
10. Now, test this password by logging out of the router and then typing **enable** at the user mode prompt. The enable secret password overrides the enable password. If you have set both passwords, you must use the enable secret password to enter privileged mode. The enable password is still configured but is now deactivated.
    Krang(config)#exit
    Krang#exit
    Krang>enable
    Password:
    Krang#

Lab 6: Banner MOTD

Objective*:* Configure a banner Message of the Day (MOTD). The MOTD is displayed when a user logs on to the router. The banner can also be used to display information about the router itself or to display a security message.

Lab Equipment*:* Router 1 from the **eRouters** menu

1. Connect to Router 1, and enter privileged mode.
   Router>
   Router>enable
   Router#
2. Enter configuration mode.
   Router#config t
   Router(config)#
3. Type the command to enter the banner message, and press ENTER. After you type **banner motd**, enter a delimiting character so the router knows when you are finished entering text for the banner. The easiest one to use is the letter Z.
   Router(config)#banner motd z
   Enter the text followed by the 'z' to finish
4. Now, all text that you type, until you type the letter Z, will be stored as the banner. Type the text "You do not have permission to be here. This router eats hackers for lunch! z", and press ENTER. This will set the banner.
   You do not have permission to be here. This router eats hackers for lunch! z
5. To view the banner, exit configuration mode, and then exit the router. Press ENTER to display the banner.
   Router(config#)exit
   Router#exit
   Router>exit
   Press RETURN to get started.
   You do not have permission to be here. This router eats hackers for lunch!

Lab 7: Copy Command

Objective: Become familiar with the router configuration and the **copy** commands available in the Cisco IOS.

Lab Equipment: Router 1 from the **eRouters** menu

1. Connect to Router 1, and enter privileged mode.
   Router>enable
   Router#
2. Display the active configuration in memory. The currently active configuration script running on the router is referred to as the **running-config** in the router's command-line interface (CLI). Note that privileged mode is required to display the active configuration. The running configuration script is not automatically saved on a Cisco router and will be lost in the event of power failure. The running configuration must be manually saved with the **copy** command.
   Router#show running-config
3. Try to display the configuration stored in NVRAM (known as the **startup-config**). You have not saved the configuration, so there is not one to show.
   Router#show startup-config
4. Copy the current active configuration to NVRAM. The current active configuration is in RAM; it should be saved so that the router will still boot up with the configuration in the event of a power outage.
   Router#copy running-config startup-config
5. Now, show the configuration stored in NVRAM.
   Router#show startup-config
6. If you decide that you would like to configure the router from scratch, you can erase the startup configuration and reload the router. This will enable you to completely delete all configurations on the router so that you can start from scratch. Type the command that will delete the configuration file in NVRAM. When prompted, confirm that you do want to erase the NVRAM file system by pressing the Y key.
   Router#erase startup-config
7. Now, type the command to reload the router, and press the Y key when prompted to confirm the reload.
   Router#reload
8. After the router reboots, look at the startup configuration file again. Because you did not save it before you reloaded, there is nothing there.
   Router>enable
   Router#show startup-config
9. Now, change the host name of the router to **Boson**.
   Router#config terminal
   Router(config)#hostname Boson
   Boson(config)#exit
   Boson#
10. Save your router configuration, and reload the router. Again, press the Y key when prompted to confirm the reload.
    Boson#copy run start
    Boson#reload

11. After the router reloads, the host name of **Boson** appears in the prompt. If you run the **show startup-config** command, nothing appears.
    Boson>enable
    Boson#show startup-config

Lab 8: Introduction to Interface Configuration

Objective: Learn to enable interfaces on a router, and learn what is required for the interface to be up.

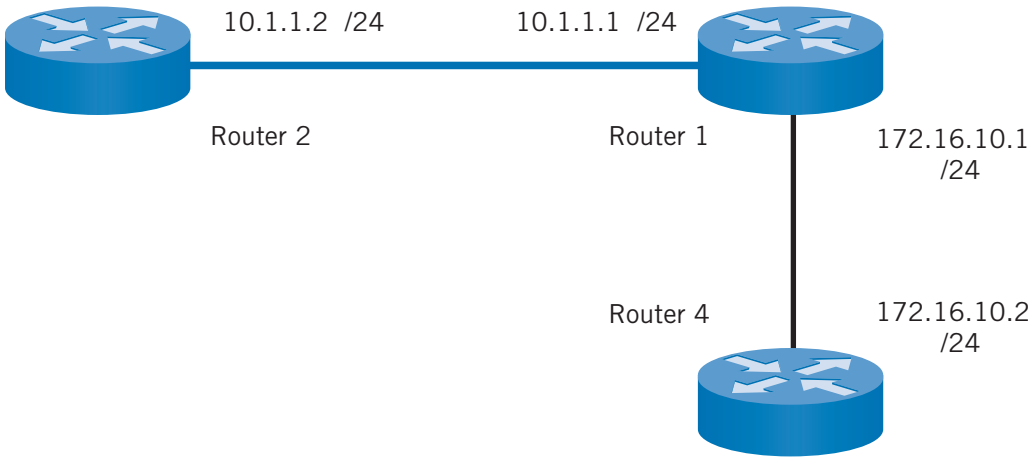
Lab Equipment: Router 1 and Router 2 from the **eRouters** menu

1. On Router 1, enter global configuration mode.
   Router>enable
   Router#conf t
   Router(config)#
   Router(config)#hostname Router1
2. Type the command to enter interface configuration mode for Ethernet 0.
   Router1(config)#interface Ethernet 0
   Router1(config-if)#
3. Display all the commands available in interface configuration mode by typing **?**.
   Router1(config-if)#?
4. The **shutdown** command shuts down the selected interface. You can often achieve the opposite of a command by typing **no** in front of it. Execute the command on Router 1 Ethernet 0 to enable the interface.
   Router1(config-if)#no shutdown
5. Add a description for this interface.
   Router1(config-if)#description Ethernet interface on Router 1
6. To view the interface description, exit back to privileged mode, and run the **show interface** command. You should see the description under Ethernet 0.
   Router1(config-if)#end
   Router1#show interface
7. Connect to Router 2, and assign it a host name of **Router2**.
   Router#conf t
   Router(config)#hostname Router2
8. Now, access the Ethernet 0 interface, and enable the interface.
   Router2(config)#interface Ethernet 0
   Router2(config-if)#no shutdown
9. Now that the interfaces on both sides of the Ethernet connection are enabled, they should be able to see one another through CDP. Use the **show cdp neighbor** command on Router2 to view all directly connected Cisco routers.
   Router2(config-if)#end
   Router2#show cdp neighbor

Lab 9: Introduction to IP

Objective: Configure Routers 1, 2, and 4 with Internet Protocol (IP) addresses, and ping between them to test connectivity.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu



1. Connect to Router 1, and assign it a host name of **Router1**.
   Router>enable
   Router#conf t
   Router(config)#hostname Router1
   Router1(config)#
2. Enter interface configuration mode for the Ethernet 0 interface.
   Router1(config)#interface ethernet 0
   Router1(config-if)#
3. Type the command that will set the IP address on the Ethernet 0 interface to 10.1.1.1 255.255.255.0, and enable the interface.
   Router1(config-if)#ip address 10.1.1.1 255.255.255.0
   Router1(config-if)#no shutdown
4. Set the IP address on the serial 0 interface of Router1 to 172.16.10.1 255.255.255.0, and enable the interface.
   Router1(config)#interface serial 0
   Router1(config-if)#ip address 172.16.10.1 255.255.255.0
   Router1(config-if)#no shut
5. Connect to Router 2, and assign it a host name of **Router2.**
   Router>enable
   Router#conf t
   Router(config)#hostname Router2
   Router2(config)#

6. Set the IP address for the Ethernet 0 interface to 10.1.1.2 255.255.255.0, and enable the interface.
   Router2(config)#interface Ethernet 0
   Router2(config-if)#ip address 10.1.1.2 255.255.255.0
   Router2(config-if)#no shutdown
7. Connect to Router 4, and assign it a host name of **Router4.**
   Router>enable
   Router#conf t
   Router(config)#hostname Router4
8. Configure an IP address of 172.16.10.2 255.255.255.0 on the serial 0 interface, and enable the interface.
   Router4(config)#interface serial 0
   Router4(config-if)#ip address 172.16.10.2 255.255.255.0
   Router4(config-if)#no shutdown
9. From Router1, try to ping Router2's Ethernet interface.
   Router1#ping 10.1.1.2
10. Try to ping Router4's serial 0 interface.
    Router1#ping 172.16.10.2
11. Verify that the lines and protocols are up for all of Router1's interfaces.
    Router1#show ip interface brief
12. Display Router1's running configuration, and verify that the IP addresses appear.
    Router1#show running-config
13. Display detailed IP information about each interface on Router1.
    Router1#show ip interface

Lab 10: ARP

Objective: Configure Routers 1 and 2 with IP addresses, and ping between them to test connectivity. Then, view the entries stored in the Address Resolution Protocol (ARP) table.

Lab Equipment: Router 1 and Router 2 from the **eRouters** menu

1. Connect to Router 1, and type the command to view the ARP table.
   Router>enable
   Router#show arp
2. Assign an IP address of 10.1.1.1 255.255.255.0 to the Ethernet 0 interface of Router 1.
   Router#conf terminal
   Router(config)#interface Ethernet 0
   Router(config-if)#ip address 10.1.1.1 255.255.255.0
   Router(config-if)# no shutdown
   Router(config-if)#exit
3. View the ARP table again.
   Router(config)#exit
   Router#show arp

4. Now, connect to Router 2, and configure its Ethernet 0 interface with an IP address of 10.1.1.2 /24.
   Router#conf terminal
   Router(config)#interface Ethernet 0
   Router(config-if)#ip address 10.1.1.2 255.255.255.0
   Router(config-if)# no shutdown
   Router(config-if)#exit
5. A connection should now exist between the Router 1 and Router 2 Ethernet interfaces. To ensure that the connection is functional, ping the IP address of Router 1's Ethernet 0 IP address from Router 2.
   Router(config)#exit
   Router# ping 10.1.1.1
6. View the ARP table on Router 2, and notice the entry.
   Router#show arp
7. Now, clear the ARP table.
   Router#clear arp
8. View the ARP table one last time, and notice what entries are there.
   Router#show arp

Lab 11: Creating a Host Table

Objective: Become familiar with the router's host table. Host tables can be used to set names for commonly used IP addresses, which helps with troubleshooting.

Lab Equipment: Router 1 from the **eRouters** menu

1. Connect to Router 1, and set the host name to **California**.
   Router>enable
   Router#config t
   Router(config)#hostname California
   California(config)#
2. Configure an IP address of 195.42.36.10 255.255.255.240 on the Ethernet 0 interface; be sure to enable the interface.
   California(config)#interface ethernet 0
   California(config-if)#ip address 195.42.36.10 255.255.255.240
   California(config-if)#no shutdown
3. Connect to Router 2, and set the host name to **Tampa**.
   Router>enable
   Router#config t
   Router(config)#hostname Tampa
   Tampa(config)#
4. Configure an IP address of 195.42.36.12 255.255.255.240 on the Ethernet 0 interface; be sure to enable the interface.
   Tampa(config)#interface ethernet 0

Tampa(config-if)#ip address 195.42.36.12 255.255.255.240
Tampa(config-if)#no shutdown
Tampa(config-if)#exit

5. Exit interface mode. You do not want to have to type California's Ethernet 0 IP address every time you try to ping it from Tampa, so set a host table entry for California using the IP address 195.42.36.10.
   Tampa(config)#ip host California 195.42.36.10
   Tampa(config#)exit
6. Now you should be able to ping California's Ethernet 0 IP address from Tampa just by typing **ping California**.
   Tampa#ping California
7. Use the **show hosts** command to verify that the entry is stored in the router's host table.
   Tampa#show hosts

Lab 12: Static Routes

Objective: Configure Routers 1, 2, and 4 with IP addresses, and then add static routes for all routers.

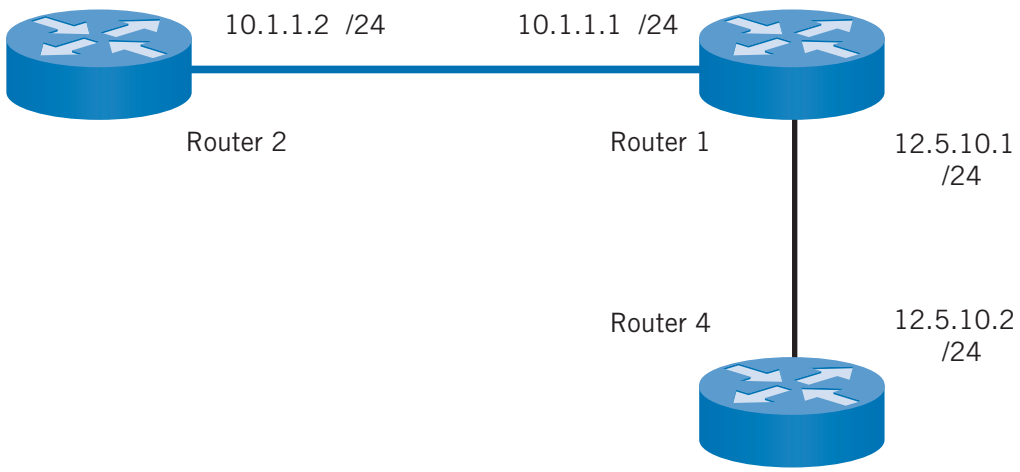
Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

Goals:

- Set the host name, and bring up the interfaces.
- Ping the directly connected interfaces.
- Configure static routes for the topology.
- Verify that you can ping all routers.

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table and diagram below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	10.1.1.1 /24	10.1.1.2 /24	
Serial 0	12.5.10.1/24		12.5.10.2 /24



2. On each router, verify that you can ping the directly connected neighbors.
   Router1#ping 10.1.1.2
   Router1#ping 12.5.10.2

   Router2#ping 10.1.1.1

   Router4#ping 12.5.10.1
3. Now you need to establish static routes on each router to any location that is not directly connected. Router1 is directly connected to both Router2 and Router4, so it will not need any static routes.

   On Router4, enter global configuration mode, and think about what the static route command should be. You know that you currently cannot reach Router2 because it is not directly connected. Off of Router4's serial interface is network 12.5.10.0, which is connected to Router1. Router1 is also connected to network 10.1.1.0, which you would also like to access. In this case, you will need a static route for network 10.1.1.0. On Router4, what command should you use to establish a static route to network 10.1.1.0?
   Router4#conf term
   Router4(config)#ip route 10.1.1.0 255.255.255.0 12.5.10.1

   You established a route to network 10.1.1.0. Now, whenever a packet of information leaves Router4 destined for network 10.1.1.0, it will first be sent to IP address 12.5.10.1 on Router1.
4. Now, try to ping Router1's serial 0 interface, Router1's Ethernet 0 interface, and Router2's Ethernet 0 interface.
   Router4#ping 12.5.10.1
   Router4#ping 10.1.1.1
   Router4#ping 10.1.1.2

   Consider why the ping to 10.1.1.2 (Router2's Ethernet 0 interface) was unsuccessful. A packet leaves Router4's serial 0 interface destined for 10.1.1.2. Because the destination address is on the 10.1.1.0 network and the static route on Router4 stipulates that traffic destined for that network should first be

sent to 12.5.10.1, the packet will travel to 12.5.10.1. When the packet reaches Router1, the router sends the packet out the interface that is directly connected to the 10.1.1.0 network. Router2 picks up that packet on its Ethernet 0 interface and attempts to send a response packet to confirm receipt. Router2 examines the source IP address of the received packet, which is 12.5.10.2 (Router4's serial 0 interface). Router2 does not have a route to network 12.5.10.0, so it drops the packet. This is why the ping was not successful.

5. Just to make sure the static route on Router4 worked, view the routing table to see if the static route has been added there.
   Router4#show ip route
6. To enable Router4 to ping 10.1.1.2, connect to Router2 and configure a static route back to Router4's network. Type the command that will set a static route on Router2 for the network 12.5.10.0.
   Router2#config term
   Router2(config)#ip route 12.5.10.0 255.255.255.0 10.1.1.1
   Router2(config)#exit

   Consequently, any data sent to network 12.5.10.0 will go to 10.1.1.1 first.
7. Connect to Router4 again, and make sure you can ping Router1's serial 0 interface, Router1's Ethernet 0 interface, and Router2's Ethernet 0 interface.
   Router4#ping 12.5.10.1
   Router4#ping 10.1.1.1
   Router4#ping 10.1.1.2
8. Examine the routing table on Router2.
   Router2#show 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
   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

   Gateway of last resort is not set

   C 10.1.1.0/24 is directly connected, 10.1.1.2
   S 12.5.10.0/24 [1/0] via 10.1.1.1

   In the **S 12.5.10.0/24 [1/0] via 10.1.1.1** line of output, the **S** denotes the static route. Next, the destination network and its subnet information **(12.5.10.0/24)** are displayed. The **[1/0]** represents the administrative distance, which is **1** by default, and the metric (hop count in this case), which is **0**. The word **via** signals the next hop address the packet should be sent to, which in this case is **10.1.1.1**.

Lab 13: RIP

Objective: Configure Routers 1, 2, and 4 with IP addresses and the Routing Information Protocol (RIP).

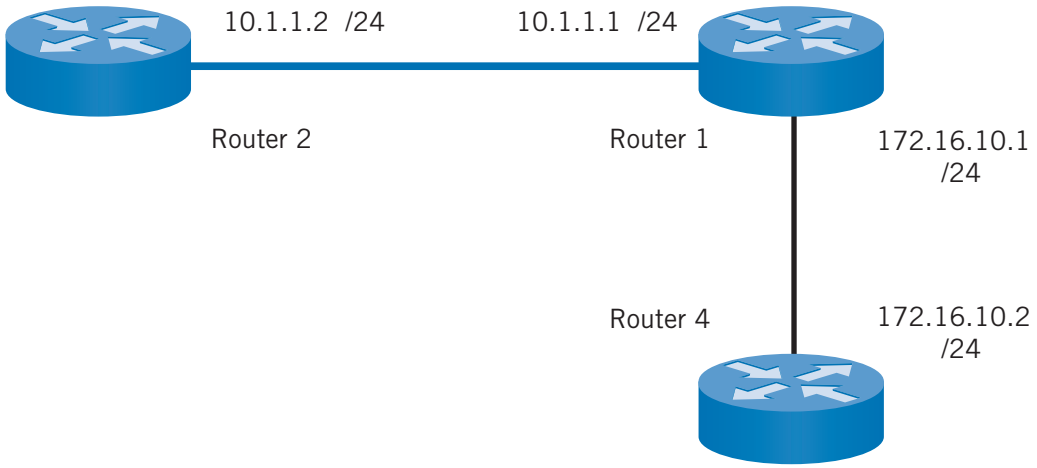
Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

Goals:

- Set the host name and bring up the interfaces.
- Configure RIP.
- Select the directly connected networks.
- Display the routing table.
- Display the RIP protocol information.

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table and diagram below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	10.1.1.1 /24	10.1.1.2 /24	
Serial 0	172.16.10.1 /24		172.16.10.2 /24



2. On each router, verify that you can ping the directly connected neighbors.
   Router1#ping 10.1.1.2
   Router1#ping 172.16.10.2

   Router2#ping 10.1.1.1

   Router4#ping 172.16.10.1

3. Add RIP to Router1.
   Router1#
   Router1#configure terminal
   Router1(config)#router rip
   Router1(config-router)#
4. Add the network(s) to which Router1 is directly connected.
   Router1(config-router)#network 10.0.0.0
   Router1(config-router)#network 172.16.0.0
5. Add RIP to Router2.
   Router2#
   Router2#config terminal
   Router2(config)# router rip
   Router2(config-router)#
6. Add the network(s) to which Router2 is directly connected.
   Router2(config-router)#network 10.0.0.0
7. Add RIP to Router4.
   Router4#
   Router4#config terminal
   Router4(config)# router rip
   Router4(config-router)#
8. Add the network(s) to which Router4 is directly connected.
   Router4(config-router)#network 172.16.0.0
9. Now, RIP should be running on all three routers. See if you can ping between routers that are not directly connected. For instance, from Router2 you should now be able to ping Router4's serial 0 interface.
   Router2#ping 172.16.10.2
10. Connect to Router4, and ping Router2's Ethernet 0 interface.
    Router4#ping 10.1.1.2

    If you can ping both devices, then you have correctly configured routing. If the pings were not successful, trace back through the steps.
11. Now, issue the command to display the routing table on Router4.
    Router4#show ip route
12. Finally, display specific IP routing protocol information on Router4.
    Router4#show ip protocol

Lab 14: Troubleshooting RIP

Objective: Configure IP addresses on Routers 1, 2, and 4 with Routing Information Protocol (RIP) as the routing protocol. Then, observe routing activity using the **debug ip rip** command, and examine routes using the **show ip route** command.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	192.168.1.1 /24	192.168.1.2 /24	
Serial 0	192.168.2.1 /24		192.168.2.2 /24

2. Use the proper network statements to configure RIP on all routers.

```
Router1#conf t
Router1(config)#router rip
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 192.168.2.0
Router1(config-router)#exit
Router1(config)#exit
Router1#

Router2#config t
Router2(config)#router rip
Router2(config-router)#network 192.168.1.0
Router2(config-router)#exit
Router2(config)#exit
Router2#

Router4#config t
Router4(config)#router rip
Router4(config-router)#network 192.168.2.0
Router4(config-router)#exit
Router4(config)#exit
Router4#
```

3. Use the **show ip route** command to confirm that the routes are being received on all routers.

```
Router1#show ip route

Router2#show ip route

Router4#show ip route
```

4. Once the routers have received the routes, execute the **debug ip rip** command at the privileged mode prompt on Router1.
   Router1#debug ip rip

   Observe the output on Router1's terminal screen. (The output could take up to 60 seconds to appear.)
5. To turn off the **debug** command, use the **no** keyword in front of the command (i.e., **no debug ip rip**).
   Router1#no debug ip rip
6. View the routing table entries on Router2 and Router4. Notice the administrative distances and metrics for these routes.
   Router2#show ip route

   Router4#show ip route
7. Make sure you can ping all devices on the network from every other device. If all pings do not succeed, then you will need to troubleshoot the router configurations to ensure you configured all settings correctly.
   Router1#ping 192.168.1.2
   Router1#ping 192.168.2.2

   Router2#ping 192.168.1.1
   Router2#ping 192.168.2.2

   Router4#ping 192.168.2.1
   Router4#ping 192.168.1.2

Lab 15: IGRP

Objective: Configure Routers 1, 2, and 4 with IP addresses and Interior Gateway Routing Protocol (IGRP).

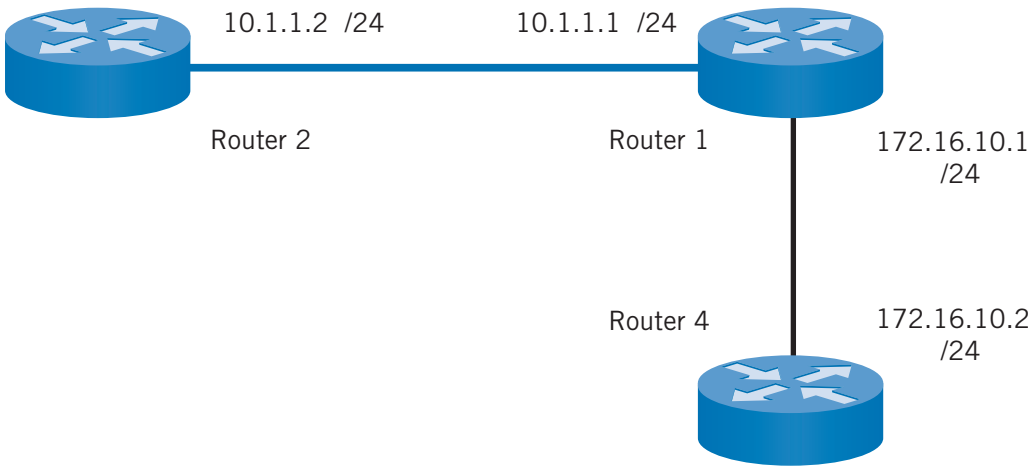
Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

Goals:

- Set the host name, and bring up the interfaces.
- Configure IGRP.
- Select the directly connected networks.
- Display the routing table.
- Display the IGRP protocol information.

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table and diagram below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	10.1.1.1 /24	10.1.1.2 /24	
Serial 0	172.16.10.1 /24		172.16.10.2 /24



2. After you have configured the correct IP address on each interface, verify that each router can ping its directly connected neighbors.
   Router1#ping 10.1.1.2
   Router1#ping 172.16.10.2

   Router2#ping 10.1.1.1

   Router4#ping 172.16.10.1
3. Access global configuration mode on Router1, and enter the command to configure IGRP as the routing protocol on Router1; use the autonomous system number **100**.
   Router1#config terminal
   Router1(config)#router igrp 100
   Router1(config-router)#
4. Add the network(s) to which Router1 is directly connected.
   Router1(config-router)#network 10.0.0.0
   Router1(config-router)#network 172.16.0.0

5. Now, enter global configuration mode on Router2, and add IGRP. Remember to use the same autonomous system number.
   Router2#config terminal
   Router2(config)#router igrp 100
   Router2(config-router)#
6. Add the network(s) to which Router2 is directly connected.
   Router2(config-router)#network 10.0.0.0
7. Now, enter global configuration mode on Router4, and add IGRP. Remember to use the same autonomous system number.
   Router4#config terminal
   Router4(config)#router igrp 100
   Router4(config-router)#
8. Add the network(s) to which Router4 is directly connected.
   Router4(config-router)#network 172.16.0.0
9. IGRP should now be running on all three routers. See if pings are successful between routers that are not directly connected. From Router2, you should now be able to ping Router4's serial 0 interface. From Router4, you should be able to ping Router2's Ethernet 0 interface.
   Router2#ping 172.16.10.2

   Router4#ping 10.1.1.2

   If you can ping both devices, then you have correctly configured routing. If the pings were not successful, trace back through the steps.
10. Now, display the routing table on Router4.
    Router4#show ip route
11. Finally, display specific IP routing protocol information on Router4.
    Router4#show ip protocol

Lab 16: PPP with CHAP Authentication

Objective: Understand how Point-to-Point Protocol (PPP) encapsulation works and how to secure the connection with Challenge Handshake Authentication Protocol (CHAP).

Lab Equipment: Router 1 and Router 4 from the **eRouters** menu

1. Enter global configuration mode on Router 1, and change the host name to **R1**.
   Router>enable
   Router#conf t
   Router(config)#hostname R1
   R1(config)#
2. The enable secret password will be used along with the host name to access the other router. Set R1's enable secret password to **sameone**.

R1(config)#enable secret sameone

3. On R1, configure a user name of **R4** with the password **myboson**.
   R1(config)#username R4 password myboson
4. Assign an IP address of 10.1.1.1 255.255.255.0 to R1's serial 0 interface.
   R1(config)#interface serial 0
   R1(config-if)#ip address 10.1.1.1 255.255.255.0
5. On R1, set the encapsulation for the serial 0 interface to PPP.
   R1(config-if)#encapsulation ppp
6. Next, set PPP authentication to CHAP on the serial 0 interface.
   R1(config-if)#ppp authentication chap
7. Now, make sure the serial 0 interface is enabled.
   R1(config-if)#no shutdown
   R1(config-if)#exit
   R1(config)#
8. Connect to Router 4, and configure a host name of **R4**.
   Router>enable
   Router#config t
   Router(config)#hostname R4
   R4(config)#
9. Set an enable secret password of **myboson** on R4.
   R4(config)#enable secret myboson
10. Add a user name of **R1** with a password of **sameone**.
    R4(config)#username R1 password sameone
11. Assign an IP address of 10.1.1.2 255.255.255.0 to the serial 0 interface on R4, and then enable the interface.
    R4(config)#interface serial 0
    R4(config-if)#ip address 10.1.1.2 255.255.255.0
    R4(config-if)#no shutdown
12. Configure the serial 0 PPP authentication to CHAP on R4.
    R4(config-if)#ppp authentication chap
13. Enable PPP encapsulation on the serial 0 interface of R4. Now, watch the interface state change to up.
    R4(config-if)#encapsulation ppp
    R4(config-if)#exit
    R4(config)#exit
    R4#
14. To verify that the configuration is correct, ping Router1's serial 0 interface from Router4.
    R4#ping 10.1.1.

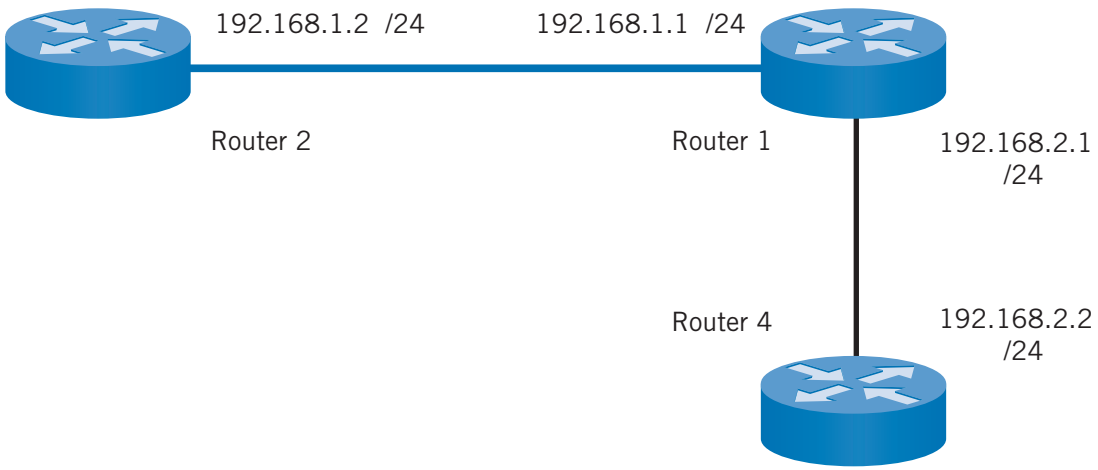
Lab 17: Connectivity Tests with Traceroute

Objective: Learn how to use the traceroute command. This command is used to map the IP addresses that a packet travels through to get from one device to another.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	192.168.1.1 /24	192.168.1.2 /24	
Serial 0	192.168.2.1 /24		192.168.2.2 /24



2. After you have configured the proper IP addresses, enable RIP routing across all three routers. Make sure you use the proper network statements.

```
Router1#conf t
Router1(config)#router rip
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 192.168.2.0
Router1(config-router)#exit
Router1(config)#exit
Router1#

Router2#config t
Router2(config)#router rip
Router2(config-router)#network 192.168.1.0
```

```
Router2(config-router)#exit
Router2(config)#exit
Router2#

Router4#config t
Router4(config)#router rip
Router4(config-router)#network 192.168.2.0
Router4(config-router)#exit
Router4(config)#exit
Router4#
```

3. From Router1, ping the directly connected routers and their interfaces, which are Router2 Ethernet 0 and Router4 serial 0.
   Router1#ping 192.168.1.2
   Router1#ping 192.168.2.2
4. Because RIP routing is enabled, you should be able to ping non-directly connected routers. Connect to Router2, and ping Router4's serial 0 interface.
   Router2#ping 192.168.2.2
5. The goal behind the **traceroute** command is to help you troubleshoot and determine the path a packet is taking to reach a destination device. In this example, there are three routers and only one path to any destination. Trace the route from Router2 to Router4's serial 0 interface.
   Router2#traceroute 192.168.2.2
6. Observe the output from the **traceroute** command. It lists Router1's Ethernet 0 IP address and then the destination IP address. This means that the packet leaves Router2's Ethernet 0 interface and passes through Router1's Ethernet 0 interface before reaching Router4's serial 0 interface.

Lab 18: Saving Router Configurations

Objective: Learn how to back up a router's configuration in case the configuration is accidentally deleted or the router fails.

Lab Equipment: Router 4 from the **eRouters** menu and PC 1 from the **eStations** menu

1. Connect to Router 4, and change the host name to **Tampa**.
   Router>enable
   Router#conf t
   Router(config)#hostname Tampa
   Tampa(config)#
2. Assign the IP address of 24.37.2.1 255.255.255.0 to the Ethernet 0 interface, and then enable the interface.
   Tampa(config)#interface ethernet 0
   Tampa(config-if)#ip address 24.37.2.1 255.255.255.0
   Tampa(config-if)#no shutdown

3. Connect to PC 1 by selecting it from the **eStations** menu. Type the command that will allow you to configure PC 1's IP address and default gateway. Set the IP address to 24.37.2.252 with a subnet mask of 255.255.255.0. Set the default gateway to Tampa's Ethernet 0 IP address (24.37.2.1).
   C:> winipcfg
4. From PC 1, ping Tampa's Ethernet 0 interface to make sure connectivity exists to the default gateway.
   C:> ping 24.37.2.1
5. Connect to Tampa again, exit interface configuration mode, and then exit global configuration mode. Copy the running configuration to the TFTP server on PC 1.
   Tampa(config-if)#exit
   Tampa(config)#exit
   Tampa# copy running-config tftp
6. When prompted for the address or name of the TFTP server, provide PC 1's IP address (24.37.2.252), press ENTER, and then provide the name of the configuration file that will be stored on PC 1. Name the configuration file Tampa_config.
   24.37.2.252
   Tampa_config

   After you press ENTER, the router will take a few seconds to establish the connection; then you will see it copy the configuration file and tell you how long it took.
7. Next, connect back to PC 1 and type the **show tftp-configs** command in order to display the configurations that are stored on the TFTP server. (**Note**: This command does not work on real PCs, just in the NetSim program.)
   C:>show tftp-configs

   If you see the configuration in the list, you have successfully completed the lab.

   Note: Lab 19 builds on this lab's configuration. To complete Lab 19, please continue with the instructions for Lab 19 in this lab. If you load another lab from the Lab Navigator, your changes will be lost and Lab 19 will not work properly.

Lab 19: Loading Router Configurations

Objective: Become familiar with the process of loading router configurations.

Lab Equipment: Router 4 from the **eRouters** menu (Tampa from Lab 18)

Prerequisite: You must have just completed Lab 18: Saving Router Configurations in order to complete this lab successfully.

1. Now that the configuration is stored on the TFTP server, change the host name of the router. This will prove that the configuration was copied from the TFTP server. Log on to Tampa, and enter global configuration mode.
   Tampa#config t
   Tampa(config)#

2. Change the host name to **Bad_Router**.
   Tampa(config)#hostname Bad_Router
3. Copy the configuration you stored on the TFTP server into the running configuration on Bad_Router.
   Bad_Router(config)#exit
   Bad_Router#copy tftp running-config
4. When the router prompts you for a name or an IP address, enter the IP address of the TFTP server.
   Address or name of remote host []?24.37.2.252
5. Enter the name of the configuration file that should be obtained from the TFTP server.
   Source filename []?Tampa_config
6. The router will download the configuration and load it into the running configuration. Afterward, the host name will be restored to what it was when the configuration was saved.
   Tampa#

Lab 20: Copying and Pasting Configurations

Objective: Learn to save, reload, and paste modified configurations from within the Simulator.

Lab Equipment: Router1 from the **eRouters** menu

Cisco Routers use a command-line parsing routine. Each time you press a carriage return, the router parses that command and executes the code that is required to carry out the command. The Simulator works the same way. When you are working with the Simulator, you can easily switch between devices using the menus across the top of the main window. The Simulator offers some built-in saving and loading options.

1. Set the host name of Router 1 to **Router1**.
   Router>enable
   Router(config)#hostname Router1
2. Select the **Save Single Device Config** option from the File menu. The program will ask for a file name; use **Router1**, and click **Save**. Save the files to a convenient location that you will remember easily.
3. After you have saved the file, exit the Simulator, and then start it again. Reload Stand-Alone Lab 20 from the Lab Navigator.
4. Select the **Load Single Device Config (overwrite)** option from the **File** menu. Select the Router1.rtr file that you just saved, and click **Open**.
5. The program will then open the file and execute all the commands that were previously saved on the device. Once it is finished, you will notice that the host name has been restored.
6. Two other options under the **File** menu offer similar functionality: the **Save Multi Devices Configs** option and the **Load Multi Devices Configs** option. These two options respectively will save and load the configurations for all the devices.
7. Saved files can be edited easily. Minimize the program, and double-click the Router1.rtr file that you just saved to your computer. When the operating system asks you which program you would like to use to open the file, select Microsoft Notepad.

8. Notepad will launch with Router1's running configuration displayed. You will see the **hostname** command a few lines down. Change this line from **hostname Router1** to **hostname Miami**. Save your changes.
9. Now, repeat step 4, and observe the host name change.
10. If you have created a configuration that you want to paste into the routers, the program offers a tool to allow you to do this.
11. First, make sure Router1 is open. Select the **Paste Real Router Configs** option from the **File** menu. This will open a window that will allow you to paste configuration files you would like to have executed on Router1. In the empty text box, type the following:
    hostname Router1
    interface Ethernet 0
    ip address 1.1.1.1 255.255.255.0
    no shutdown
    exit
    exit
12. After you have typed the commands above, click the **OK** button. The router will quickly execute the commands. Notice that the host name of the router will change back to Router1.
13. Execute the **show ip interface brief** command on Router1 to see that the IP address has been set for Ethernet 0.

Lab 21: ISDN

Objective: Learn how to set up Integrated Services Digital Network (ISDN) on Cisco routers.

Lab Equipment: Router 1 and Router 2 from the **eRouters** menu

1. Connect to Router 1, and assign it a host name of **Router1**.
   Router>enable
   Router#conf t
   Router(config)#hostname Router1
2. Connect to Router 2, and assign it a host name of **Router2**.
   Router>enable
   Router#conf t
   Router(config)#hostname Router2
3. Now, set up the connection between Router1 and Router2 using the BRI ports. Assign the BRI 0 interface of Router1 an IP address of 42.34.10.1 with a 255.255.255.0 subnet mask, enable the interface, and then exit interface configuration mode.
   Router1(config)#interface BRI0
   Router1(config-if)#ip address 42.34.10.1 255.255.255.0
   Router1(config-if)#no shut
   Router1(config-if)#exit
   Router1(config)#

4. Now, connect to Router2, and assign its BRI 0 interface an IP address of 42.34.10.121 with a 255.255.255.0 subnet mask. Enable the interface, and then exit interface configuration mode.
   Router2(config)#interface BRI0
   Router2(config-if)#ip address 42.34.10.121 255.255.255.0
   Router2(config-if)#no shut
   Router2(config-if)#exit
   Router2(config)#
5. Return to Router1, and start to configure ISDN. First, specify the ISDN switch type that will be used. If you use the Simulator defaults, the switch type is basic-ni. There are two different ways to configure the type of ISDN switch type the router should use. You can specify the command globally for all BRI interfaces on the router, or you can make the switch type interface-specific. In this instance, enter the switch type globally on your router.
   Router1(config)#isdn switch-type basic-ni
6. Configure some specific information for this BRI interface. First, assign it the ISDN SPID (Service Profile Identifier). Set the SPID on the BRI interface of Router1 by using the **isdn spid1** command. A SPID is a number supplied by the ISP to identify the line configuration of the BRI service. Each SPID points to line setup and configuration information on the ISP's ISDN switch. If you use the defaults for the ISDN switch, the SPID for Router1 will be 32177820010100.
   Router1(config)#interface bri 0
   Router1(config-if)#isdn spid1 32177820010100
7. Now that you have configured the switch type and SPID, Layer 1 connectivity should exist. Layer 1 connectivity occurs between the ISDN switch and the router. To verify that Layer 1 connectivity exists, use the **show isdn status** command at the privileged mode prompt. Make sure that the Layer 2 state is **Multiple_Frame_Established**.
   Router1(config-if)#exit
   Router1(config)#exit
   Router1#show isdn status
8. Now, configure the number that will need to be dialed on the ISDN switch to establish a Layer 3 connection; this is called the dialer string. Set the dialer string on Router1's BRI 0 interface. If you are using the default configuration, use 7782001.
   Router1#config t
   Router1(config)#interface bri 0
   Router1(config-if)#dialer string 7782001
9. Because ISDN costs money when the connection is up, the connection should only be active when it is being used. You can use dialer groups and dialer lists to accomplish this. A dialer list either permits or denies traffic. Specify a dialer list of **protocol ip permit**; consequently, all IP traffic will be permitted. To set up a dialer list, use the **dialer-list** command in global configuration mode.
   Router1(config-if)#exit
   Router1(config)#dialer-list 1 protocol ip permit
10. The dialer list must be associated with an interface. Add the dialer list to the ISDN BRI 0 interface by using the **dialer-group 1** command.
    Router1(config)#interface bri 0

Router1(config-if)#dialer-group 1

11. Now that you have set up ISDN on Router1, you need to perform the same steps for Router2, but with some slight modifications. Connect to Router2, and specify the ISDN switch type that you will be using. If you use the Simulator defaults, the switch type is basic-ni. Specify the switch type in global configuration mode on the router.
    Router2(config)#isdn switch-type basic-ni
12. Next, provide the SPID for this interface. If you use the Simulator defaults for the ISDN switch, the SPID for Router 2 will be 32177820020100.
    Router2(config)#interface bri 0
    Router2(config-if)#isdn spid1 32177820020100
13. Now that you have set up the switch type and SPID, Layer 1 connectivity should be established. To verify that Layer 1 connectivity exists, use the **show isdn status** command at the privileged mode prompt. Make sure that the Layer 2 state is **Multiple_Frame_Established**.
    Router2(config-if)#exit
    Router2(config)#exit
    Router2#show isdn status
14. Now, configure the dialer string that you will need to dial on the ISDN switch in order to establish a Layer 3 connection. Set the dialer string on Router2's BRI 0 interface. If you are using the default configuration, use 7782002.
    Router2#config t
    Router2(config)#interface bri 0
    Router2(config-if)#dialer string 7782002
15. Configure the dialer list named **protocol ip permit** on Router2 to permit all IP traffic.
    Router2(config-if)#exit
    Router2(config)#dialer-list 1 protocol ip permit
16. Use the **dialer-group 1** command to add the dialer list to the ISDN BRI 0 interface.
    Router2(config)#interface bri 0
    Router2(config-if)#dialer-group 1
17. Now that both routers are configured for ISDN, see if you can ping the router on the other side of the connection. From Router2, ping Router1's BRI 0 interface (IP address 42.34.10.1).
    Router2(config-if)#exit
    Router2(config)#exit
    Router2#ping 42.34.10.1
18. If the ping is successful, ISDN is working. Verify this by issuing the **show isdn status** command on Router2.
    Router2#show isdn status

    Examine the Layer 3 settings; there should be one active Layer 3 call. You should also see that the SPID is valid in Layer 2. This information is useful for troubleshooting.
19. Finally, view the configuration changes you have made by displaying the running configuration.
    Router2#show running-config

Lab 22: Introduction to the Switch

Objective: View some basic areas of a Cisco Catalyst 1900 switch.

Lab Equipment: Switch 1 from the **eSwitches** menu

1. Connect to Switch 1. You should see the user mode prompt.
   >
2. Enter the command to display the IOS version of the switch.
   >show version

   What version of the IOS is running? _____

   What is the model number of the switch? _____

   What is the Base Ethernet Address of the switch? _____
3. Display the interfaces of the switch.
   >show interfaces

   How many of the interfaces are 10 Mbps? _____

   How many ports are 100 Mbps Fast Ethernet? _____
4. Enter the command to view the MAC address table.
   >show mac-address-table

   How many dynamic entries have been learned? _____
5. Display the running configuration.
   >show running-config

Lab 23: Introduction to Basic Switch Commands

Objective: Become familiar with the basic configuration of the Cisco Catalyst 1912 switch.

Lab Equipment: Switch 1 from the **eSwitches** menu

1. Connect to Switch 1. You should see the user mode prompt.
   >
2. Display the list of commands available at this prompt.
   >?
3. Now, enter privileged mode.
   >enable
   #
4. Display the available commands in privileged mode.
   #?

5. Enter configuration mode.
   #config terminal
   (config)#
6. The host name is used for local identification. When you log on to the switch, you see the host name in front of the prompt (if a host name has been configured). The host name can be used to identify the location or function of the switch. Set the switch's host name to **Boson**.
   (config)#hostname Boson
   Boson(config)#
7. The enable password controls access to privileged mode. This is a very important password because, when it is configured, only those who know the password can make configuration changes in privileged mode.

   There is a difference in the syntax used to set the password for a router and the syntax used to set the password for a switch. On the 1900 series switch, levels need to be set when a password is declared. The different levels allow different sets of people to enter different commands on the switch. The password levels range from 1 to 15. Level 1 allows the user to log in to the router and use very basic **show** commands. Level 15 allows the user to do anything. The levels in between can be customized by the network administrator to allow certain commands.

   On Switch1, set the enable password to **Krang**.
   Boson(config)#enable password level 15 Krang
8. Test the password by first exiting the switch and then trying to enter privileged mode. Notice that you have to provide the enable password in order to get into privileged mode. Now, type **conf term** and proceed with the lab instructions in the next step.
   Boson(config)#exit
   Boson#exit
   Boson>enable
   Password:
   Boson#conf term
   Boson(config)#
9. The only problem with the enable password is that it appears in plain text in the switch's configuration file. If you need to obtain assistance while troubleshooting a problem, you may inadvertently compromise the security of your system by revealing the password. Set the enable secret password to **cisco**. Do not forget to use the level commands.
   Boson(config)#enable secret level 15 cisco
10. You can now test this password by logging out of the switch and then trying to access privileged mode. The enable secret password overrides the enable password. If you have set both passwords, the enable secret password is the password you should use to enter privileged mode. The enable password is now deactivated.
    Boson(config)#exit
    Boson#exit
    Boson>enable
    Password:
    Boson#

Lab 24: Frame Relay

Objective: Learn to establish a Frame Relay connection.

Lab Equipment: Router 1 and Router 2 from the **eRouters** menu

1. Connect to Router 1, and configure the host name to **R1**.
   Router>enable
   Router#conf t
   Router(config)#hostname R1
   R1(config)#
2. Assign an IP address of 10.1.1.1 255.255.255.0 to the serial 0 interface, and enable the interface.
   R1(config)#interface serial 0
   R1(config-if)#ip address 10.1.1.1 255.255.255.0
   R1(config-if)#no shut
3. Now, connect to Router 2 and change the host name to **R2**.
   Router>en
   Router#config t
   Router(config)#hostname R2
   R2(config)#
4. Assign an IP address of 10.1.1.2 255.255.255.0 to the serial 0 interface, and enable the interface.
   R2(config)#interface serial 0
   R2(config-if)#ip address 10.1.1.2 255.255.255.0
   R2(config-if)#no shut
5. On R1, set the encapsulation for the serial 0 interface to Frame Relay. Notice that the interface is still down.
   R1(config-if)#encapsulation frame-relay
6. Next, set the Frame Relay interface data-link connection identifier (DLCI) for the connection from R1 to R2. Because the default Frame Relay network is being used, the DLCI will be **102**.
   R1(config-if)#frame-relay interface-dlci 102
7. On R2, set the encapsulation for the serial 0 interface to Frame Relay. Notice that the serial 0 interface is still down.
   R2(config-if)#encapsulation frame-relay
8. Now, set the Frame Relay interface DLCI for the connection from R2 to R1. Because the default Frame Relay network is being used, the DLCI will be **201**.
   R2(config-if)#frame-relay interface-dlci 201

   You should have seen the output from the router saying that the DLCI changed to the active state. This means you have established a connection from R1 through the Frame Relay switch to R2.
9. From R2, verify that the configuration is correct by first trying to ping the serial 0 IP address on R1.
   R2(config-if)#exit
   R2(config)#exit
   R2#ping 10.1.1.1

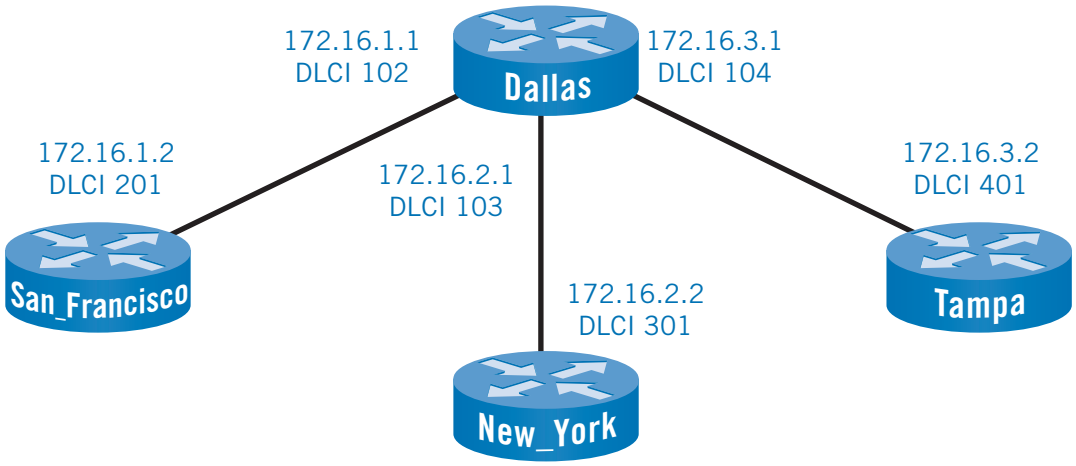
10. Next, use the Frame Relay **show** commands to prove that the connection is active. The **show frame-relay lmi** command displays the Local Management Interface (LMI) traffic that has been exchanged between the router and the Frame Relay switch.
    R2#show frame-relay lmi
11. The **show frame-relay traffic** command displays the global Frame Relay statistics since the last reload of the router.
    R2#show frame-relay traffic
12. The **show frame-relay map** command displays the mappings of the Layer 2 DLCI to the Layer 3 IP address.
    R2#show frame-relay map
13. The **show frame-relay PVC** command displays all of the permanent virtual circuit (PVC) mappings for the router. These mappings are only locally significant between the router and the Frame Relay switch.
    R2#show frame-relay pvc

Lab 25: Frame Relay Hub-and-Spoke Topology

Objective: Learn to configure a hub-and-spoke topology.

Lab Equipment: Router 1, Router 2, Router 3, and Router 4 from the **eRouters** menu

Your company's corporate office is in Dallas, and its sales offices are in San Francisco, New York, and Tampa. You want to implement a hub-and-spoke topology in which all of the sales offices connect to the corporate office to send all data, including communications between sales offices.



1. First, assign the host names of **Dallas**, **San_Francisco**, **New_York**, and **Tampa** to Router 1, Router 2, Router 3, and Router 4, respectively.
2. Now, enter interface configuration mode for the serial 0 interface on Dallas, and set the encapsulation type to Frame Relay. Be sure to enable the interface.
   Dallas(config)#interface serial 0
   Dallas(config-if)#encapsulation frame-relay
   Dallas(config-if)#no shutdown

3. Next, create a subinterface for the connection from Dallas to the San Francisco sales office.
   Dallas(config-if)#exit
   Dallas(config)#interface serial 0.100 point-to-point
   Dallas(config-subif)#
4. Assign the subinterface the DLCI number for the connection from Dallas to San_Francisco, and configure the subinterface with the appropriate IP address. Remember to enable the subinterface.
   Dallas(config-subif)#frame-relay interface-dlci 102
   Dallas(config-subif)#ip address 172.16.1.1 255.255.255.0
   Dallas(config-subif)#no shutdown
5. Create a subinterface for the connection from Dallas to the sales office in New York.
   Dallas(config-subif)#exit
   Dallas(config)#interface serial 0.200 point-to-point
   Dallas(config-subif)#
6. Add the correct DLCI for the connection from Dallas to New_York, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
   Dallas(config-subif)#frame-relay interface-dlci 103
   Dallas(config-subif)#ip address 172.16.2.1 255.255.255.0
   Dallas(config-subif)#no shutdown
7. Create a subinterface for the connection from Dallas to the sales office in Tampa.
   Dallas(config-subif)#exit
   Dallas(config)#interface serial 0.300 point-to-point
   Dallas(config-subif)#
8. Add the correct DLCI for the Dallas to Tampa connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
   Dallas(config-subif)#frame-relay interface-dlci 104
   Dallas(config-subif)#ip address 172.16.3.1 255.255.255.0
   Dallas(config-subif)#no shutdown
9. Access the serial 0 interface on San_Francisco, set the encapsulation to Frame Relay, and enable the interface.
   San_Francisco(config)#interface serial 0
   San_Francisco(config-if)#encapsulation frame-relay
   San_Francisco(config-if)#no shutdown
10. Because subinterfaces are not necessary for single connections, add the appropriate DLCI value.
    San_Francisco(config-if)#frame-relay interface-dlci 201
11. Set the IP address for this interface, and enable the interface.
    San_Francisco(config-if)#ip address 172.16.1.2 255.255.255.0
    San_Francisco(config-if)# no shutdown
12. Access the serial 0 interface on New_York, and set the encapsulation to Frame Relay.
    New_York(config)#interface serial 0
    New_York(config-if)#encapsulation frame-relay

13. Add the appropriate DLCI value.
    New_York(config-if)#frame-relay interface-dlci 301
14. Set the IP address for this interface, and enable the interface.
    New_York(config-if)#ip address 172.16.2.2 255.255.255.0
    New_York(config-if)#no shutdown
15. Access the serial 0 interface on Tampa, and set the encapsulation to Frame Relay
    Tampa(config)#interface serial 0
    Tampa(config-if)#encapsulation frame-relay
16. Add the appropriate DLCI value.
    Tampa(config-if)#frame-relay interface-dlci 401
17. Configure the IP address for this interface, and enable the interface.
    Tampa(config-if)#ip address 172.16.3.2 255.255.255.0
    Tampa(config-if)#no shutdown
18. Now, all interfaces should be up and up. To confirm this, connect to Dallas and try to ping each of the three sales offices.
    Dallas(config-if)#exit
    Dallas(config)#exit
    Dallas#ping 172.16.1.2
    Dallas#ping 172.16.2.2
    Dallas#ping 172.16.3.2

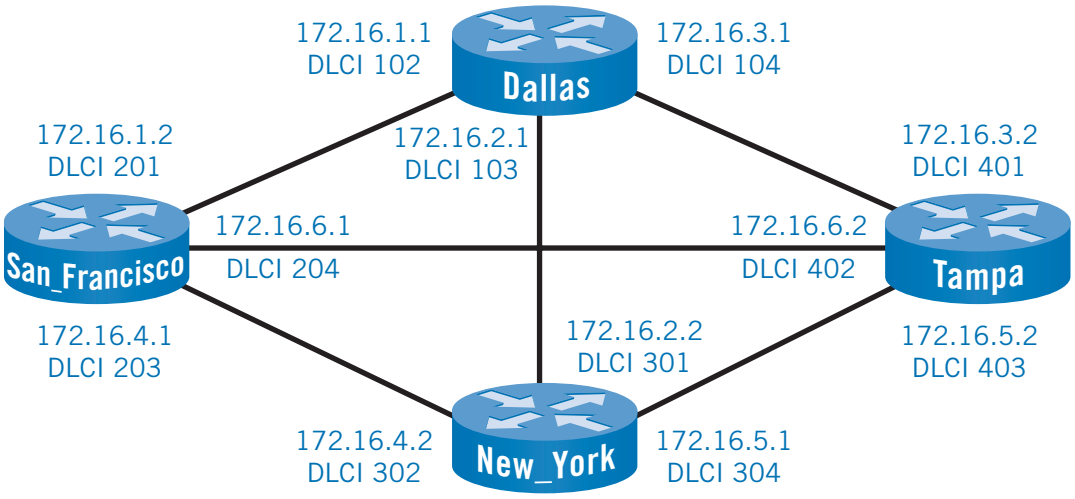
Lab 26: Frame Relay Full Mesh Topology

Objective: Learn to configure a full mesh topology.

Lab Equipment: Router 1, Router 2, Router 3, and Router 4 from the **eRouters** menu

Again, the company's corporate office is in Dallas and its sales offices are in San Francisco, New York, and Tampa. The sales offices should be able to access all company resources. You want to establish a full mesh topology in which a point-to-point Frame Relay connection links the corporate office to each sales office and links each sales office to every other sales office.

The difference between the Frame Relay hub-and-spoke topology and the full mesh topology is that, with a full mesh topology, every sales office has a direct connection to every other sales office and the corporate office. This is a very redundant topology so, if one of the connections fails, data can still be transferred to every site by using a different path.



1. First, assign the host names of **Dallas**, **San_Francisco**, **New_York**, and **Tampa** to Router 1, Router 2, Router 3, and Router 4, respectively.
2. Now, enter interface configuration mode for the serial 0 interface on Dallas, and set the encapsulation type to Frame Relay. Be sure to enable the interface.
   Dallas(config)#interface serial 0
   Dallas(config-if)#encapsulation frame-relay
   Dallas(config-if)#no shutdown
3. Next, create a subinterface for the connection between Dallas and the San Francisco sales office.
   Dallas(config-if)#exit
   Dallas(config)#
   Dallas(config)#interface serial 0.100 point-to-point
   Dallas(config-subif)#
4. Assign the subinterface the DLCI number for the connection from Dallas to San_Francisco, and configure the subinterface with the appropriate IP address. Remember to enable the subinterface.
   Dallas(config-subif)#frame-relay interface-dlci 102
   Dallas(config-subif)#ip address 172.16.1.1 255.255.255.0
   Dallas(config-subif)#no shutdown
5. Create a subinterface for the connection from Dallas to the sales office in New York.
   Dallas(config-subif)#exit
   Dallas(config)#interface serial 0.200 point-to-point
   Dallas(config-subif)#
6. Add the correct DLCI for the connection from Dallas to New_York, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.

Dallas(config-subif)#frame-relay interface-dlci 103
Dallas(config-subif)#ip address 172.16.2.1 255.255.255.0
Dallas(config-subif)#no shutdown

7. Create a subinterface for the connection from Dallas to the sales office in Tampa.
   Dallas(config-subif)#exit
   Dallas(config)#interface serial 0.300 point-to-point
   Dallas(config-subif)#
8. Add the correct DLCI for the connection from Dallas to Tampa, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
   Dallas(config-subif)#frame-relay interface-dlci 104
   Dallas(config-subif)#ip address 172.16.3.1 255.255.255.0
   Dallas(config-subif)#no shutdown
9. Access the serial 0 interface of San_Francisco, set the encapsulation to Frame Relay, and enable the interface.
   San_Francisco(config)#interface serial 0
   San_Francisco(config-if)#encapsulation frame-relay
   San_Francisco(config-if)#no shutdown
10. Create the first subinterface for the connection from San_Francisco to the corporate office in Dallas.
    San_Francisco(config-if)#interface serial 0.100 point-to-point
    San_Francisco(config-subif)#
11. Add the correct DLCI for the connection from San_Francisco to Dallas, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    San_Francisco(config-subif)#frame-relay interface-dlci 201
    San_Francisco(config-subif)#ip address 172.16.1.2 255.255.255.0
    San_Francisco(config-subif)#no shutdown
12. Create a subinterface for the connection from San_Francisco to New_York.
    San_Francisco(config-subif)#exit
    San_Francisco(config)#interface serial 0.200 point-to-point
13. Add the correct DLCI value for the connection from San_Francisco to New_York, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    San_Francisco(config-subif)#frame-relay interface-dlci 203
    San_Francisco(config-subif)#ip address 172.16.4.1 255.255.255.0
    San_Francisco(config-subif)#no shutdown
14. Create the subinterface for the connection from San_Francisco to Tampa.
    San_Francisco(config-subif)#exit
    San_Francisco(config)#interface serial 0.300 point-to-point
15. Add the correct DLCI value for the San_Francisco to Tampa connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    San_Francisco(config-subif)#frame-relay interface-dlci 204
    San_Francisco(config-subif)#ip address 172.16.6.1 255.255.255.0
    San_Francisco(config-subif)#no shutdown

16. Access the serial 0 interface of New_York, set the encapsulation to Frame Relay, and enable the interface.
    New_York(config)#interface serial 0
    New_York(config-if)#encapsulation frame-relay
    New_York(config-if)#no shutdown
17. Create the first subinterface for the connection from New_York to the corporate office in Dallas.
    New_York(config-if)#exit
    New_York(config)#interface serial 0.100 point-to-point
18. Add the correct DLCI value for the connection from New_York to Dallas, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    New_York(config-subif)#frame-relay interface-dlci 301
    New_York(config-subif)#ip address 172.16.2.2 255.255.255.0
    New_York(config-subif)#no shutdown
19. Create the subinterface for the connection from New_York to San_Francisco.
    New_York(config-subif)#exit
    New_York(config)#interface serial 0.200 point-to-point
20. Add the correct DLCI value for the connection from New_York to San_Francisco, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    New_York(config-subif)#frame-relay interface-dlci 302
    New_York(config-subif)#ip address 172.16.4.2 255.255.255.0
    New_York(config-subif)#no shutdown
21. Create the subinterface for the connection from New_York to Tampa.
    New_York(config-subif)#exit
    New_York(config)#interface serial 0.300 point-to-point
22. Add the correct DLCI value for the New_York to Tampa connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    New_York(config-subif)#frame-relay interface-dlci 304
    New_York(config-subif)#ip address 172.16.5.1 255.255.255.0
    New_York(config-subif)#no shutdown
23. Access the serial 0 interface on Tampa, set the encapsulation to Frame Relay, and enable the interface.
    Tampa(config)#interface serial 0
    Tampa(config-if)#encapsulation frame-relay
    Tampa(config-if)#no shutdown
24. Create the first subinterface for the connection from Tampa to the corporate office in Dallas.
    Tampa(config-subif)#exit
    Tampa(config-if)#interface serial 0.100 point-to-point
25. Add the correct DLCI value for the Tampa to Dallas connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    Tampa(config-subif)#frame-relay interface-dlci 401
    Tampa(config-subif)#ip address 172.16.3.2 255.255.255.0
    Tampa(config-subif)#no shutdown

26. Create the subinterface for the connection from Tampa to San_Francisco.
    Tampa(config-subif)#exit
    Tampa(config)#interface serial 0.200 point-to-point
27. Add the correct DLCI value for the Tampa to San_Francisco connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    Tampa(config-if)#frame-relay interface-dlci 402
    Tampa(config-subif)#ip address 172.16.6.2 255.255.255.0
    Tampa(config-subif)#no shutdown
28. Create the subinterface for the connection from Tampa to New_York.
    Tampa(config-subif)#exit
    Tampa(config)#interface serial 0.300 point-to-point
29. Add the correct DLCI value for the Tampa to New_York connection, and configure the appropriate IP address for the subinterface. Remember to enable the subinterface.
    Tampa(config-subif)#frame-relay interface-dlci 403
    Tampa(config-subif)#ip address 172.16.5.2 255.255.255.0
    Tampa(config-subif)#no shutdown
30. Now, all interfaces should be up and up. To test the configuration, connect to Dallas and try to ping each of the three sales offices.
    Dallas(config-if)#exit
    Dallas(config)#exit
    Dallas#ping 172.16.1.2
    Dallas#ping 172.16.2.2
    Dallas#ping 172.16.3.2
31. Connect to San_Francisco, and try to ping the other three offices.
    San_Francisco(config-subif)#exit
    San_Francisco(config)#exit
    San_Francisco#ping 172.16.1.1
    San_Francisco#ping 172.16.4.2
    San_Francisco#ping 172.16.6.2

Lab 27: Standard Access Lists

Objective: Gain experience configuring standard access lists.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

If you feel confident about configuring IP addresses and RIP, establish the configuration in the table below, and then continue with step 10.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Enable RIP	on E0 and S0	on E0	on S0
Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240

1. Connect to Router 1, assign it a host name of **Router1**, and set the IP address on the Ethernet 0 interface to 24.17.2.1 255.255.255.240. Set the IP address on the serial 0 interface to 24.17.2.17 255.255.255.240. Remember to enable both interfaces.
   Router>
   Router#
   Router#config t
   Router(config)#hostname Router1
   Router1(config)#interface ethernet0
   Router1(config-if)#ip address 24.17.2.1 255.255.255.240
   Router1(config-if)#no shutdown
   Router1(config-if)#exit
   Router1(config)#interface serial0
   Router1(config-if)#ip address 24.17.2.17 255.255.255.240
   Router1(config-if)#no shutdown
   Router1(config-if)#exit
   Router1(config)#exit
2. Connect to Router 2, assign it a host name of **Router2**, and set the IP address on the Ethernet 0 interface to 24.17.2.2 255.255.255.240. Remember to enable the interface.
   Router>
   Router>enable
   Router#config t
   Router(config)#hostname Router2
   Router2(config)#interface ethernet0
   Router2(config-if)#ip address 24.17.2.2 255.255.255.240
   Router2(config-if)#no shutdown
   Router2(config-if)#exit
   Router1(config)#exit
3. From Router2, ping Router1's Ethernet 0 interface to ensure a connection exists.
   Router2#ping 24.17.2.1
4. Connect to Router 4, assign it a host name of **Router4**, and set the IP address on the serial 0 interface to 24.17.2.18 255.255.255.240. Then, ping Router1's serial 0 interface.

```
Router>
Router>enable
Router#config t
Router(config)#hostname Router4
Router4(config)#interface serial0
Router4(config-if)#ip address 24.17.2.18 255.255.255.240
Router4(config-if)#no shutdown
Router4(config-if)#exit
Router4(config)#exit
Router4#ping 24.17.2.17
```

5. Now that IP addresses have been configured on all interfaces, you need to implement a routing protocol to facilitate communication between Router2 and Router4. Enable Routing Information Protocol (RIP) on **Router1, and add the network for Ethernet 0 and serial 0.**

```
Router1#config t
Router1(config)#router rip
Router1(config-router)#network 24.0.0.0
Router1(config-router)#exit
Router1(config)#exit
```

6. On Router2, enable RIP and add the network for Ethernet 0.

```
Router2#conf t
Router2(config)#router rip
Router2(config-router)#network 24.0.0.0
Router2(config-router)#exit
Router2(config)#exit
```

7. On Router4, enable RIP and add the network for serial 0.

```
Router4#conf t
Router4(config)#router rip
Router4(config-router)#network 24.0.0.0
Router4(config-router)#exit
Router4(config)#exit
```

8. Verify that you can ping Router2's Ethernet 0 interface from Router4.

```
Router4#ping 24.17.2.2
```

9. Now, configure a standard access list to block Router4 from being able to ping Router2. You should configure this access list on Router2. First, connect to Router2 and enter global configuration mode.

```
Router2#conf t
Router2(config)#
```

10. Create access list 1 to block the single IP address 24.17.2.18. Here are three ways to accomplish this:

```
Router2(config)#access-list 1 deny host 24.17.2.18
```

OR

```
Router2(config)#access-list 1 deny 24.17.2.18 0.0.0.0
```

OR

```
Router2(config)#access-list 1 deny 24.17.2.18
```

11. Next, issue the **access-list 1 permit any** command.
    Router2(config)#access-list 1 permit any
12. Now you need to apply the access list to the Ethernet 0 interface. You must specify the direction of traffic flow upon which the access list should apply. The **in** parameter configures the access list to apply to packets coming in from the network and traveling to the router. The **out** parameter configures the access list to apply to packets traveling from the router out the interface to the network. In this scenario, you should use the **in** parameter.
    Router2(config)#interface ethernet0
    Router2(config-if)#ip access-group 1 in
    Router2(config-if)#exit

    Note: This completes the Standard Access Lists lab. Please continue on to Lab 28: Verify Standard Access Lists without accessing the Lab Navigator.

Lab 28: Verify Standard Access Lists

Objective: Verify that the standard access list created in the previous lab is configured correctly.

Lab Equipment: Router 2 and Router 4 from the **eRouters** menu

Prerequisite: You must have just completed Lab 27: Standard Access Lists in order to complete this lab successfully.

1. First, see if you can still ping Router2 from Router4. Connect to Router4, and try to ping Router2's Ethernet 0 interface (24.17.2.2).
   Router>enable
   Router4#ping 24.17.2.2
2. If you see **UUUUU**, indicating that the ping was not successful, then your access list is working correctly.
3. Now, connect to Router2 and view the running configuration in order to verify that the access list is running on the interfaces.
   Router>enable
   Router2#show running-config
4. You can also view which access lists are applied to the interfaces by using the **show ip interface** command.
   Router2#show ip interface
5. The **show access-lists** command will display which access lists have been created on the router. It will also tell you which lines have been used and how many packets have been either permitted or denied.
   Router2#show access-lists

   Note: Continue on to Lab 29: Extended Access Lists without accessing the Lab Navigator. This will save you the trouble of configuring the same IP addresses again.

Lab 29: Extended Access Lists

Objective: Gain experience configuring extended access lists.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

1. If you have just completed Lab 28: Verifying Standard Access Lists, then all you need to do is execute the **no ip access-group 1 in** command on the Ethernet 0 interface of Router2, and then start this lab at step 10.
   Router2>enable
   Router2#conf t
   Router2(config)#interface ethernet0
   Router2(config-if)#no ip access-group 1 in

   Note: If you have not completed Lab 28: Verifying Standard Access Lists and you feel confident about configuring IP addresses and RIP, establish the configuration in the table below and then continue with step 10.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Enable RIP	on E0 and S0	on E0	on S0
Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240

2. Connect to Router 1, assign it a host name of **Router1**, and set the IP address on the Ethernet 0 interface to 24.17.2.1 255.255.255.240. Set the IP address on the serial 0 interface to 24.17.2.17 255.255.255.240. Remember to enable both interfaces.
   Router>
   Router>enable
   Router#conf t
   Router(config)#hostname Router1
   Router1(config)#interface ethernet0
   Router1(config-if)#ip address 24.17.2.1 255.255.255.240
   Router1(config-if)#no shutdown
   Router1(config-if)#exit
   Router1(config)#interface serial0
   Router1(config-if)#ip address 24.17.2.17 255.255.255.240
   Router1(config-if)#no shutdown
   Router1(config-if)#exit
   Router1(config)#exit

3. Connect to Router 2, assign it a host name of **Router2**, and set the IP address on the Ethernet 0 interface to 24.17.2.2 255.255.255.240. Remember to enable the interface.
   Router>
   Router>enable
   Router#config t
   Router(config)#hostname Router2
   Router2(config)#interface ethernet0
   Router2(config-if)#ip address 24.17.2.2 255.255.255.240
   Router2(config-if)#no shutdown
   Router2(config-if)#exit
   Router2(config)#exit
4. Ping Router1's Ethernet 0 interface to ensure that a connection exists.
   Router2#ping 24.17.2.1
5. Connect to Router 4, assign it a host name of **Router4**, and set the IP address on the serial 0 interface to 24.17.2.18 255.255.255.240. Then, ping Router1's serial 0 interface.
   Router>
   Router>enable
   Router#conf t
   Router(config)#hostname Router4
   Router4(config)#interface serial0
   Router4(config-if)#ip address 24.17.2.18 255.255.255.240
   Router4(config-if)#no shutdown
   Router4(config-if)#exit
   Router4(config)#exit
   Router4#ping 24.17.2.17
6. Now you need to implement a routing protocol to facilitate communication between Router2 and Router4. Enable Routing Information Protocol (RIP) on Router1, and add the network for Ethernet 0 and serial 0.
   Router1#config t
   Router1(config)#router rip
   Router1(config-router)#network 24.0.0.0
   Router1(config-router)#exit
   Router1(config)#exit
7. On Router2, enable RIP and add the network for Ethernet 0.
   Router2#conf t
   Router2(config)#router rip
   Router2(config-router)#network 24.0.0.0
   Router2(config-router)#exit
   Router2(config)#exit
8. On Router4, enable RIP and add the network for serial 0.
   Router4#conf t
   Router4(config)#router rip

Router4(config-router)#network 24.0.0.0
Router4(config-router)#exit
Router4(config)#exit

9. Verify that you can ping Router2's Ethernet 0 interface from Router4.
   Router4#ping 24.17.2.2
10. The extended access lists you create should accomplish two things. First, allow only Telnet traffic from the subnet off of Router1's serial 0 interface to come into Router1. Next, allow any traffic from Router1's Ethernet 0 subnet to travel anywhere. Connect to Router1, and enter global configuration mode.
    Router1#conf t
    Router1(config)#
11. To allow only Telnet traffic from the 24.17.2.16 subnet, create access list 101. Use the **log** keyword to display output to the router every time this line on the access list is invoked.
    Router1(config)#access-list 101 permit tcp 24.17.2.16 0.0.0.15 any eq telnet log
12. To permit all traffic from the 24.17.2.0 subnet, create access list 102, and use the **log** keyword.
    Router1(config)#access-list 102 permit ip 24.17.2.0 0.0.0.15 any log
13. Now, apply these access lists to the interfaces. First, enter interface configuration mode for the serial 0 interface of Router1, and apply access list 101 inbound.
    Router1(config)#interface serial0
    Router1(config-if)#ip access-group 101 in
    Router1(config-if)#exit
14. For Ethernet 0 on Router1, apply access list 102 inbound.
    Router1(config)#interface ethernet0
    Router1(config-if)#ip access-group 102 in
    Router1(config-if)#exit

    Note: To make sure the access lists are configured correctly, continue on to Lab 30: Verify Extended Access Lists without accessing the Lab Navigator.

Lab 30: Verify Extended Access Lists

Objective: Verify that the extended access lists created in Lab 29 are configured correctly.

Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

Prerequisite: You must have just completed Lab 29: Extended Access Lists in order to complete this lab successfully.

1. Test whether the extended access lists created in Lab 29 are working properly. Connect to Router4, and try to ping Router1's serial 0 interface. If the access lists are configured correctly, you should not be able to ping the serial interface.
   Router4>enable
   Router4#ping 24.17.2.17
2. Now that you have verified that the access lists are blocking pings to Router1 from the subnet off of Router1's serial 0 interface, verify that Telnet traffic from that subnet is allowed to reach Router1. Con-

nect to Router1, enable Telnet access, and then set the password to **boson**.

```
Router1(config)#
Router1(config)#line vty 0 4
Router1(config-line)#login
Router1(config-line)#password boson
Router1(config-line)#exit
```

3. Connect to Router4 again, and try to telnet into Router1's serial 0 interface.
   ```
   Router4#telnet 24.17.2.17
   ```
4. If Telnet access is permitted, you should see the host name in the router prompt change to **Router1**. Now, press the CTRL+SHIFT+6 key combination followed by the X key to return to Router4. Then, type **disconnect 1** to close the connection to Router1.
   Press CTRL+SHIFT+6, then press X
   ```
   Router4#disconnect 1
   ```
5. Connect to Router2, and see if you can ping Router4's serial 0 interface.
   ```
   Router2>enable
   Router2#ping 24.17.2.18
   ```

   Consider why the ping is unsuccessful. The packet starts at Router2, travels through Router1, and reaches Router 4. Once it arrives at Router4, it is repackaged and sent back to Router1. When Router4 repackages the packet, the packet's source IP address becomes the destination IP address, and the destination IP address becomes the source IP address. When the packet encounters the access list on Router1's serial 0 interface, it is blocked because the packet's source IP address is Router4's serial 0 address.
6. See if you can ping Router1's Ethernet 0 interface from Router2.
   ```
   Router2#ping 24.17.2.1
   ```
7. Now, try to telnet into Router1's Ethernet 0 interface from Router2. If Telnet access is permitted, you should see the host name in the router prompt change to Router1. Press the CTRL+SHIFT+6 key combination followed by the X key to return to Router4. Then, type **disconnect 1** to close the connection to Router1.
   ```
   Router2#telnet 24.17.2.1
   ```
   Press CTRL+SHIFT+6, then press X
   ```
   Router2#disconnect 1
   ```
8. To verify that the access lists are configured on the interfaces, display the running configuration.
   ```
   Router1#show running-config
   ```
9. You can also view which access lists are applied to the interfaces by using the **show IP interface** command.
   ```
   Router1#show ip interface
   ```
10. The **show access-lists** command displays which access lists have been created on a router. The output of this command also tells you which lines of the access list have been used and how many packets have been permitted or denied.
    ```
    Router1#show access-lists
    ```

Lab 31: Named Access Lists

Objective: Create a named access list that will deny all ping traffic from PC 1 to Router 1, but will enable all access from Router 4 to Router 1. For this lab, the access list must be added on Router 1.

Lab Equipment: Router 1 and Router 4 from the **eRouters** menu and PC 1 from the **eStations** menu

1. Establish the configurations outlined in the table below. Use the **winipcfg** command on PC 1 to configure the IP address.

Device	Router 1	Router 4	PC 1
Host Name	Router1	Router4	PC 1
Ethernet 0		192.168.1.17 /28	192.168.1.18 /28
Serial 0	192.168.1.1 /28	192.168.1.2 /28	
Default Gateway			192.168.1.17

2. Configure RIP on the two routers. Be sure to use the proper network statements.
   Router1(config)#router rip
   Router1(config-router)#network 192.168.1.0

   Router4(config)#router rip
   Router4(config-router)#network 192.168.1.0
3. Use the **show ip route** command on each router to make sure that the routes have been received.
   Router1#show ip route

   Router4#show ip route
4. Verify that you can ping Router1 from PC 1.
   C:>ping 192.168.1.1
5. Create an access list that prevents ping traffic originating from PC 1 and destined for Router1 from reaching Router1. Typically, this access list could be placed on either Router4 or Router1. It often makes more sense to place the access list on the router closest to the source as possible; this helps keep unnecessary traffic off the backbone. For this example, however, the access list will be placed on Router1 for inbound traffic.
   Router1(config)#ip access-list extended deny_ping
   Router1(config-ext-acl)#deny icmp host 192.168.1.18 192.168.1.1 0.0.0.0 log
   Router1(config-ext-acl)#permit ip any any log

   The first statement above defines the access list as extended. The second line denies any ICMP traffic with a source IP address of 192.168.1.18 that is destined for 192.168.1.1; the wildcard mask of **0.0.0.0** in this line means that the IP address must be matched exactly. Notice how the **host** command is used for the first part of the access list and the wildcard mask of **0.0.0.0** is used for the second part of the ac-

cess list. The **host** command and the wildcard mask of **0.0.0.0** both do the same thing. The **log** keyword allows you to double-check your work.

6. Next, apply the access list to inbound traffic on the serial 0 interface of Router1.
   Router1(config-ext-acl)#exit
   Router1(config)#interface serial 0
   Router1(config-if)#ip access-group deny_ping in
7. Now, connect to PC 1 and send a test ping to Router1. Is the ping successful? Connect to Router4, and send a test ping to the serial 0 interface of Router1.
   C:>ping 192.168.1.1

   Router4#ping 192.168.1.1
8. Connect to Router1 again; you should see two separate log messages. The first one is denying the ping from PC 1, and the second is allowing the ping from Router4.

Lab 32: Advanced Extended Access Lists

Objective: Configure extended access lists to filter out network-to-network traffic, host-to-host traffic, and network-to-host traffic.

Lab Equipment: Router 1 and Router 2 from the **eRouters** menu and PC 1, PC 2, PC 3, PC 4, and PC 5 from the **eStations** menu

1. Establish the configurations outlined in the tables below.

Device	Router 1	Router 2
Host Name	Router1	Router2
FA0/0	192.168.3.1 /24	192.168.1.129 /25
FA0/1		192.168.1.1 /25
Serial 0	192.168.2.1 /24	192.168.2.2 /24

Host	IP Address	Subnet Mask	Default Gateway
PC 1	192.168.3.2	255.255.255.0	192.168.3.1
PC 2	192.168.1.130	255.255.255.128	192.168.1.129
PC 3	192.168.1.131	255.255.255.128	192.168.1.129
PC 4	192.168.1.2	255.255.255.128	192.168.1.1
PC 5	192.168.1.3	255.255.255.128	192.168.1.1

2. Configure RIP on the two routers. Be sure to use the proper network statements.

```
Router1#conf t
Router1(config)#router rip
Router1(config-router)#network 192.168.2.0
Router1(config-router)#network 192.168.3.0

Router2#conf t
Router2(config)#router rip
Router2(config-router)#network 192.168.1.0
Router2(config-router)#network 192.168.2.0
```

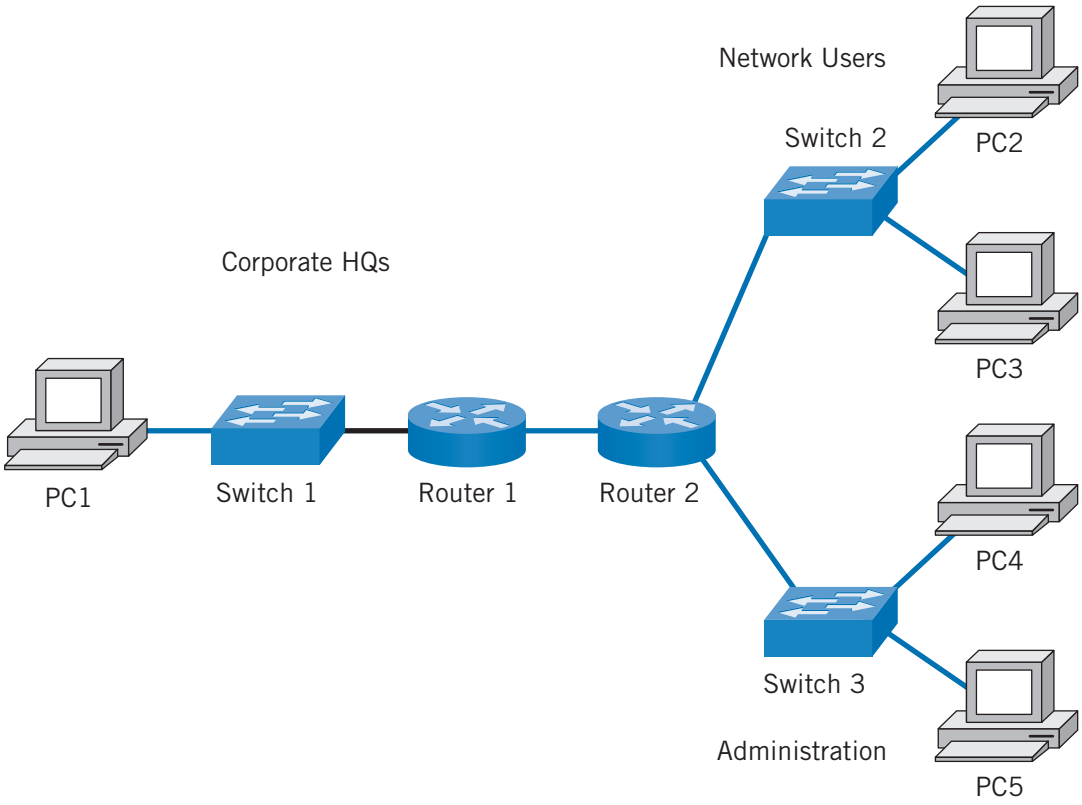
3. Use the **show ip route** command on each router to make sure that the routes have been received.
   Router1#show ip route

   Router2#show ip route
4. Verify that you can ping PC 1 from PC 2.
   C:>ping 192.168.3.2

Network-to-Network Access List

5. Examine the network diagram below. The first access list you create should allow only traffic from the Administration network (PC 4 and PC 5) destined for PC 1 on the Corporate HQ network. To accomplish this, use an extended access list. Because you are allowing all traffic, you should use IP as the protocol. The access list should look something like the following:
   Router1(config)#access-list 100 permit ip 192.168.1.0 0.0.0.127 192.168.3.0 0.0.0.255 log
   Router1(config)#access-list 100 permit ip 192.168.2.0 0.0.0.0 any

   This access list is very simple because you are only allowing two types of traffic and denying all other traffic. Because there is an implicit deny at the end of all access lists, you only need a **permit** statement for the pings and a **permit** statement for the RIP broadcasts.



6. Now you need to apply the access list to the interface. Because the traffic is coming from Router2 and going to Router1, you should place the access list on Router1's serial 0 interface. The access list will check all inbound traffic.
   Router1#conf t
   Router1(config)#interface serial 0
   Router1(config-if)#ip access-group 100 in
7. To test the access list, try to ping PC 1 from PC 2, PC 3, PC 4, and PC 5. PC 2 and PC 3 should not be able to ping PC 1, but PC 4 and PC 5 should. If this access list works, continue on to the next step
   C:>ping 192.168.3.2

Host-to-Host Access List

8. In this portion of the lab, you will block an individual PC from accessing the central file server. PC 2 is being used by a new employee whom you do not want to have access to the file server (PC 5) for 30 days. To accomplish this, you decide to implement an access list on Router2 that will block access to PC 5 only from PC 2. In this instance, you are setting the access list manually. The list must be manually removed after 30 days.

   For lab scenario purposes, you should use the **log** keyword. This will show logging output on the screen of Router2 when the access list is invoked. For this part of the lab, the log will show up on the screen only when you deny access from PC 2.
   Router2(config)#access-list 101 deny ip host 192.168.1.130 192.168.1.3 0.0.0.0 log
   Router2(config)#access-list 101 permit ip any any

9. Apply the access list to Router2's Fast Ethernet 0/0 interface.
   Router2#conf t
   Router2(config)#interface FastEthernet 0/0
   Router2(config-if)#ip access-group 101 in
10. Connect to PC 2, and verify that you cannot ping PC 5. Connect to PC 3, and verify that you can ping PC 5.
    C:>ping 192.168.1.3
11. Finally, connect to Router2, and verify that the log statements displayed on the console match the corresponding pings sent from the PCs.

Network-to-Host Access List

12. Before you create this access list, remove the preceding access lists from Router1 and Router2.
    Router1(config)#interface serial 0
    Router1(config-if)#no ip access-group 100 in

    Router2(config)#interface FastEthernet 0/0
    Router2(config-if)#no ip access-group 101 in
13. Create an extended access list that blocks all traffic to PC 1 from the Network Users area in the topology. The access list should look something like the following:
    Router2(config)#access-list 102 deny ip 192.168.1.128 0.0.0.127 host 192.168.3.2 log
    Router2(config)#access-list 102 permit ip any any
14. Apply this access list to outbound traffic on the serial 0 interface of Router2.
    Router2(config)#interface serial 0
    Router2(config-if)#ip access-group 102 out
15. To test this access list, try to ping PC 1 from PC 2 or PC 3. The pings should fail. You can also view the log file on Router2.
    C:>ping 192.168.3.2

Lab 33: Telnet

Objective: Learn to establish a Telnet session between two routers.

Lab Equipment: Router 1 from the **eRouters** menu

Note: The Simulator has limited Telnet support beyond the commands shown within this lab.

1. Connect to Router 1, and set the host name to **Router1**. Then, access the Telnet lines. Each line in a router represents an active Telnet session that the router can support. Routers in the Simulator support five Telnet lines, so use the **line vty 0 4** command.
   Router>enable
   Router#conf t
   Router(config)#hostname Router1
   Router1(config)#line vty 0 4
   Router1(config-line)#

2. Configure the router to require the use of a login password.
   Router1(config-line)#login
3. Configure **boson** as the password that will be used to establish a Telnet session.
   Router1(config-line)#password boson
4. Now, assign the IP address of 34.25.67.18 255.255.255.224 to Router1's Ethernet 0 interface, and enable the interface.
   Router1(config-line)#exit
   Router1(config)#interface Ethernet 0
   Router1(config-if)#ip address 34.25.67.18 255.255.255.224
   Router1(config-if)#no shut
5. Next, connect to Router 2, set its host name to **Router2**, and then access its Ethernet 0 interface.
   Router>en
   Router#conf t
   Router(config)#hostname Router2
   Router2(config)#interface Ethernet 0
   Router2(config-if)#
6. Assign the IP address 34.25.67.2 255.255.255.224 to Router2's Ethernet 0 interface, and enable the interface.
   Router2(config-if)#ip address 34.25.67.2 255.255.255.224
   Router2(config-if)#no shutdown
   Router2(config-if)#end
7. From Router2, telnet into Router1's Ethernet 0 interface.
   Router2#telnet 34.25.67.18
8. You will be prompted for a password. Type the **boson** password, and press ENTER. You will see a dialog box informing you that NetSim provides limited support for Telnet. Notice that the router host name changes from **Router2** to **Router1**, which indicates that you have established a Telnet session to Router1. Now, press the CTRL+SHIFT+6 key combination, then immediately press the X key. Notice that the host name changes back to **Router2**.
   Password:
   Router1#
   Press CTRL+SHIFT+6, then press X
   Router2#
9. Type the **show sessions** command to view all active Telnet sessions. To resume a Telnet session, specify the number of the session you would like to resume. In this case, there is only one Telnet session, so type the **resume 1** command.
   Router2#show sessions
   Router2#resume 1
   Router1#
10. Because you have telneted into Router1 again, the host name has changed to Router1 again. Press the CTRL+SHIFT+6 key combination followed by the X key to return to Router2.
    Router1#

CTRL+SHIFT+6 followed by X
Router2#

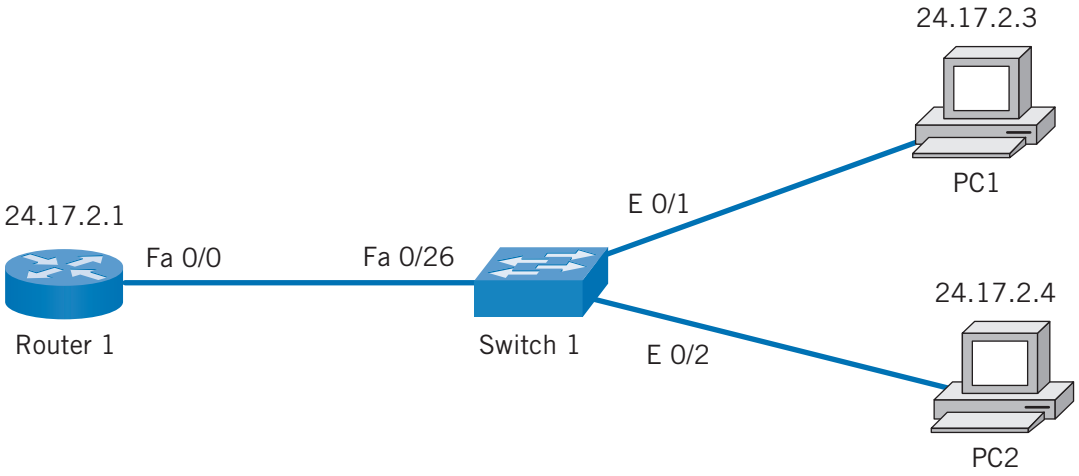
11. To disconnect the session, type the **disconnect 1** command.
    Router2#disconnect 1

Lab 34: VLANs

Objective: Become familiar with the benefits of VLANs on a LAN while using a Cisco Catalyst 1900 series switch.

Lab Equipment: Router 1 from the **eRouters** menu, Switch 1 from the **eSwitches** menu, and PC 1 and PC 2 from the **eStations** menu

In this lab, you are going to configure a router and a switch to support VLANs. First, you will set up PC 1 and PC 2 so that they can ping each other through the switch. You will then change the VLANs on the switch and observe that the PCs can no longer ping each other or the router. Next, you will change the configuration on the switch so that the PCs are on the same VLAN; they will then be able to ping each other again. You will configure the network to the specifications shown in the diagram below.



1. Connect to Router 1, assign it a host name of **Router1**, and configure the IP address of 24.17.2.1 255.255.255.0 on the Fast Ethernet 0/0 interface.
   Router>enable
   Router#
   Router#conf t
   Router(config)#hostname Router1
   Router1(config)#interface FastO/0
   Router1(config-if)#ip add 24.17.2.1 255.255.255.0
   Router1(config-if)#no shut
2. Connect to PC 1, and set the IP address to 24.17.2.3 255.255.255.0 with a default gateway of 24.17.2.1.
   C:>winipcfg

3. Connect to PC 2, and set the IP address to 24.17.2.4 255.255.255.0 with a default gateway of 24.17.2.1.
   C:>winipcfg
4. You should now be able to ping Router1 and PC 1 from PC 2.
   C:>ping 24.17.2.1
   C:>ping 24.17.2.3
5. Now, connect to Switch 1 and set up the VLANs. The switch automatically has VLAN 1 set up on all ports. In this case, you need to set up a separate VLAN for the PCs. Start by creating VLAN 22.
   >enable
   #config t
   (config)#vlan 22 name pcs
6. Now you need to assign the ports to the new VLAN. Start by assigning port 1 for PC 1 to VLAN 22.
   (config)#int e0/1
   (config-if)#vlan-membership static 22
7. Connect to PC 2 again, and try to ping Router1 and PC 1.
   C:>ping 24.17.2.1
   C:>ping 24.17.2.3

   Consider the result. You were able to ping from PC 2 to Router1, but not from PC 2 to PC 1. Why? On the switch, you set VLAN 22 to only cover port 1. That means ports 2 through 12 and the two Fast Ethernet ports were still on VLAN 1. So, when the ping packets came into the switch from PC 2, they were tagged with VLAN 1 and could only travel out of ports tagged with VLAN 1. (Although there are exceptions to this rule, they will not be covered in this lab manual.) Consequently, the ping packets could not go out port 1 to PC 1.
8. Connect to the switch again and configure port 2, which is where PC 2 is connected, to be included in VLAN 22.
   (config-if)#exit
   (config)#int e0/2
   (config-if)#vlan-membership static 22
9. Connect to PC 2 once again, and repeat the pings to Router1 and PC 1.
   C:>ping 24.17.2.1
   C:>ping 24.17.2.3

   What did you notice that was different? You should have been able to ping PC 1 but not Router 1. When the ping packets came in, they were tagged with VLAN 22. Consequently, the packets could only travel out port 1 to PC 1. This is what you wanted to accomplish.
10. Connect to the switch again, and view the VLAN port assignments by using the **show VLAN** and **show VLAN-membership** commands.
    (config-if)#end
    #show vlan
    #show vlan-membership

11. On the switch, assign FastEthernet 0/26 to VLAN 22. This will allow you to ping all devices.
    #conf t
    (config)#interface FastEthernet 0/26
    (config-if)#vlan-membership static 22
12. Send test pings from Router1 to PC 1 and PC 2, and from PC 1 and PC 2 to Router1.
    Router1#ping 24.17.2.3
    Router1#ping 24.17.2.4

    C:>ping 24.17.2.1

    C:>ping 24.17.2.1

Lab 35: VTP

Objective: Configure VLANs on Cisco Catalyst 2950 switches.

Lab Equipment: Switch 3 and Switch 4 from the **eSwitches** menu

Goals:

- Assign VLANs to multiple ports.
- Configure VLAN Trunking Protocol (VTP) to establish a server and client connection.
- Create a trunk line between the two switches to carry the VLANs.
- Test the configuration.

1. Start by assigning host names and IP addresses to Switch 3 and Switch 4 according to the table below.

Device	Switch 3	Switch 4
Host Name	Switch3	Switch4
IP Address (VLAN1)	10.1.1.1	10.1.1.2
Subnet Mask	255.255.255.0	255.255.255.0

Switch3#conf t
Switch3(config)#interface vlan1
Switch3(config-if)#ip address 10.1.1.1 255.255.255.0
Switch3(config-if)#no shutdown
Switch3(config-if)#end
Switch3#

Switch4#conf t
Switch4(config)#interface vlan1
Switch4(config-if)#ip address 10.1.1.2 255.255.255.0
Switch4(config-if)#no shutdown

```
Switch4(config-if)#end
Switch4#
```

2. Verify that the switches are connected to each other by pinging Switch3 from Switch4.
   Switch4#ping 10.1.1.1
3. Add VLAN 8 and VLAN 14 to Switch3, assign ports 2 through 5 to VLAN 8, and assign ports 6 through 10 to VLAN 14.

```
Switch3#vlan database
Switch3(vlan)#vlan 8
Switch3(vlan)#vlan 14
Switch3(vlan)#exit
Switch3#conf t
Switch3(config)#interface range fast0/2 – 5
Switch3(config-range)#switchport access vlan 8
Switch3(config-range)#exit
Switch3(config)#interface range fast 0/6 – 10
Switch3(config-range)#switchport access vlan 14
Switch3(config-range)#exit
Switch3(config)#exit
Switch3#
```

4. Use the **show vlan** command on Switch3 to verify that your configurations are correct.
   Switch3#show vlan
5. By default, a Catalyst switch is configured as a VTP server. Configure Switch3 as a VTP server, and configure Switch4 as a VTP client. Also, change the VTP domain to **Boson** and add a VTP password of **rules**.

```
Switch3#vlan database
Switch3(vlan)#vtp server
Switch3(vlan)#vtp domain Boson
Switch3(vlan)#vtp password rules
Switch3(vlan)#exit
Switch3#

Switch4#vlan database
Switch4(vlan)#vtp client
Switch4(vlan)#vtp domain Boson
Switch4(vlan)#vtp password rules
Switch4(vlan)#exit
Switch4#
```

6. Next, create the trunk link that will transport the VLAN configurations from Switch3 to Switch4. To accomplish this, enable trunking on the port that links between the two switches. The encapsulation method will be 802.1q because that is the only supported encapsulation for the 2950 switch.

```
Switch3# conf t
Switch3(config)#interface fast 0/12
Switch3(config-if)#switchport mode trunk
```

Switch3(config-if)#end
Switch4#conf t
Switch4(config)#interface fast 0/12
Switch4(config-if)#switchport mode trunk
Switch4(config-if)#end

7. After this configuration, you should be able to view the VLANs from Switch3 on Switch4. To verify the VLAN configurations, use the **show vlan** command on Switch4. Also, the **show vtp status** command will display some VTP-specific information.
Switch4# show vlan
Switch4# show vtp status

Lab 36: OSPF Single Area Configuration and Testing

Objective: Configure Routers 1, 2, and 4 with IP addresses and the Open Shortest Path First (OSPF) Routing Protocol.

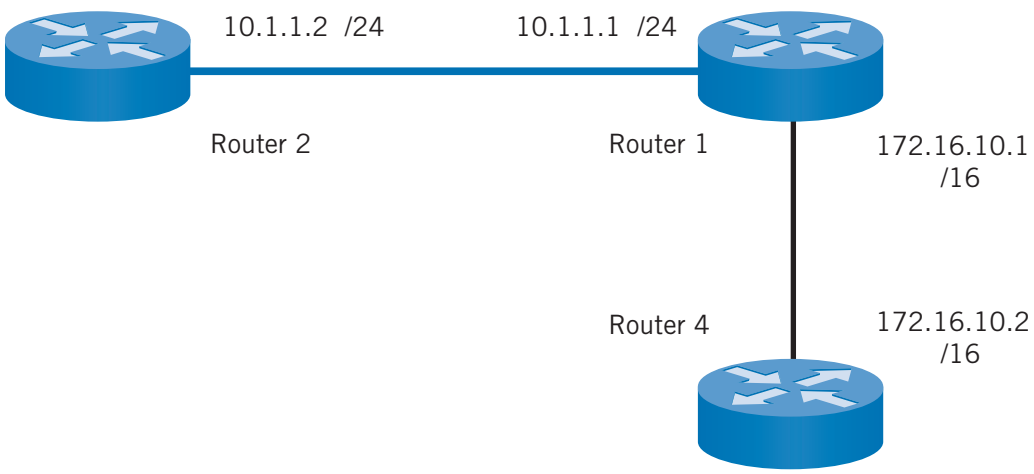
Lab Equipment: Router 1, Router 2, and Router 4 from the **eRouters** menu

Goals:

- Set the host name, and bring up the interfaces.
- Configure the OSPF routing protocol.
- Select the directly connected networks.
- Display the routing table.
- Display the OSPF protocol information.

1. Configure Routers 1, 2, and 4 to the specifications outlined in the table and diagram below.

Device	Router 1	Router 2	Router 4
Host Name	Router1	Router2	Router4
Ethernet 0	10.1.1.1 /24	10.1.1.2 /24	
Serial 0	172.16.10.1 /16		172.16.10.2 /16



2. Verify that each router can ping its directly connected neighbors.
   Router1#ping 10.1.1.2
   Router1#ping 172.16.10.2

   Router2#ping 10.1.1.1

   Router4#ping 172.16.10.1
3. Add OSPF to Router1; use the Process ID number 100.
   Router1#
   Router1#config terminal
   Router1(config)# router ospf 100
   Router1(config-router)#
4. Add the network(s) to which Router1 is directly connected.
   Router1(config-router)#network 10.1.1.0 0.0.0.255 area 0
   Router1(config-router)#network 172.16.0.0 0.0.255.255 area 0
5. Now, add OSPF to Router2.
   Router2#
   Router2#config terminal
   Router2(config)#router ospf 100
   Router2(config-router)#
6. Add the network(s) to which Router2 is directly connected.
   Router2(config-router)#network 10.1.1.0 0.0.0.255 area 0
7. Now, add OSPF to Router4.
   Router4#
   Router4#config terminal
   Router4(config)#router ospf 100
   Router4(config-router)#

8. Add the network(s) to which Router4 is directly connected.
   Router4(config-router)#network 172.16.0.0 0.0.255.255 area 0
9. OSPF should now be running on all three routers. Press CTRL+Z to exit to privileged mode, and see if you can ping non-directly connected routers. From Router2, you should now be able to ping Router4's serial 0 interface.
   Router2#ping 172.16.10.2
10. Next, connect to Router4 and ping Router2's Ethernet 0 interface.
    Router4#ping 10.1.1.2

    If you can ping both devices, then you have correctly configured routing. If you were not successful, trace back through the steps.
11. Now, display the routing table on Router2.
    Router2#show ip route
12. Display the specific IP routing protocol information on Router2.
    Router2#show ip protocols
13. Type the command that will display the OSPF database.
    Router2#show ip ospf database
14. Type the command that will display all of the OSPF neighbors.
    Router2#show ip ospf neighbor
15. Finally, type the command that will display all router interfaces that are running OSPF.
    Router2#show ip ospf interface