Boson®

Education by Simulation

Stand Alone Labs For CCNA

Cisco Press, 800 East 96th Street, Indianapolis, Indiana, 46240, or toll-free 800-858-7674.

License

Boson Trademarks

BOSON®, BOSON.COM®, BOSON ROUTER SIMULATOR®, QUIZWARE®, BOSONSOFTWARE®, BOSON TRAINING®, BOSON NETSIM®, BOSON SWITCH SIMULATOR™, BOSON STATION SIMULATOR™, BOSON NETWORK DESIGNER™, BOSON CERTIFIED LABS™, BOSON NETWORK SIMULATOR™, BOSON NETWORK EMULATOR™, BOSON CLASS IN A BOX™, BOSON ESWITCH™, BOSON EROUTER®, and BOSON ESTATION™, are trademarks or registered trademarks of Boson Software, Inc. in the United States and certain other countries.

Other Trademarks

Cisco®, Cisco Systems®, CCDA®, CCNA®, CCDP®, CCNP®, CCIE®, IOS®, CCSI™ the Cisco Systems logo and the CCIE logo are trademarks or registered trademarks of Cisco Systems, Inc. in the United States and certain other countries. Windows® is a trademark or registered trademark of Microsoft Corporation. Pentium® is a trademark or registered trademark of Intel Corporation. Athlon® is a trademark or registered trademark of Advanced Micro Devices, Inc. Adobe® and Acrobat® are trademarks or registered trademarks of Adobe Systems, Inc. Norton Personal Firewall™ is a trademark or registered trademark of Symantec Corporation. ZoneAlarm™ is a trademark or registered trademark of Zone Labs, Inc.

All other trademarks and/or registered trademarks are the property of their respective owners. Any use of a third party trademark does not constitute a challenge to said mark.

Any use of a product name or company name herein does not imply any sponsorship of, recommendation of, endorsement of, or affiliation with, Boson Software, its licensors, licensees, partners, affiliates, and/or publishers.

Version: 060104a

ISBN: 1-58720-131-3

First Edition June 2004

Contents

Lab 1: Connecting and Logging on to a Cisco Router

Objective: To introduce the Cisco Router.
Lab Equipment: We will be using Router 1. To select Router 1 click on the button labeled "Router 1" at the top of your screen..

1. If you have not done so already, click on the eRouters button located at the top of your screen and select "Router 1" . 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 down into two parts, the hostname and the mode. "Router" is the Router 1's hostname and ">" means you are in user mode.

```
Press RETURN to get Started
Router>
```

3. Next type the command enable to get to the privileged mode prompt.

```
Router>enable
Router#
```

4. To get back to the user mode, simply type disable. From the user mode type logout or exit to leave 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: To introduce ourselves to the Command Line Interface; user and privileged mode, basic help and show commands.
Lab Equipment: We will be using Router 1. To select Router 1 click on the button labeled "Router 1" at the top of your screen.

1. Press <enter> to get to the router prompt.

```
Router>
```

2. You are now in *User mode*. Type the command that is used to view all the available commands at this prompt.

```
Router>?
```

3. Type the command used to enter *Privilege mode*.

```
Router>enable
Router#
```

4. View the available commands in *Privilege 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, hit the key that will show you the next page of information.

```
<space bar>
```

8. Type one of the commands that will log you out of the router.

```
Router#exit
or
Router#disable
```

Basic User Interface Review

This review will require the use of the simulator to help with your responses.

1. You connect to Router 1 and wish to view all the available commands. What command would you use to do this? ______________

2. You need to now enter Privilege mode. What command would you use?
____

3. You want to view all available commands for Privilege mode. What command would you use? __________________

4. How would you view a list of all available show commands? What would you type?
____

5. Take a look at the routers running configuration. What command would you use?
____

6. How would you return tp the User mode. What command did you use?
____

Basic Lab Summary

This lab will introduce the Cisco Internetwork Operating System (IOS) command line interface (CLI). You will need to logon to a router and become familiar with the different levels of access on the router. You will also become familiar with the commands available to you in each mode (user or privileged) and the router help facility, history, and editing features.

User vs. Privileged Mode

User mode is indicated with the '>' next to the router name. You can look at settings but can not make changes from user mode. In Privilege mode (indicated by the '#', you can do anything). To get into privilege mode the keyword is ENABLE**.**

```
Router>
Router>enable
Password:
Router#
```

HELP

To view all commands available from this mode type: ? and press: enter This will give you the list of all available commands for the router in your current mode. You can also use the question mark after you have started typing a command. For example if you want to use a show command but you do not remember which one it uses 'show ?' will output all commands that you can use with the show command.

```
Router#show ?
access-expression          List access expression
access-lists               List access lists
backup                     Backup status
cdp                        CDP information
clock                      Display the system clock
cls                        DLC user information
compress                   Show compression statistics
configuration              Contents of Non-Volatile memory
--More--
```

Configuration Mode

From privilege mode you can enter configuration mode by typing CONFIG T you can exit configuration mode type END or **<CTL>+z**

```
Router#config t
Router(config)#end
```

Lab 3: Introduction to Basic Show Commands

Objective: To become familiar with the basic show commands.
Lab Equipment: We will be using Router 1. To select Router 1 click on the drop down box located in the top center of the screen.

1. Get to the router prompt.

```
Router>
```

2. Enter Privilege Mode.

```
Router>enable
Router#
```

3. Show the active configuration in memory. The currently active configuration script running on the router is referred to as the *running-config* on the routers command-line interface. Note that privileged mode is required. 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. Flash memory is a special kind of memory on the router that contains the operating system image file(s). Unlike regular router memory, Flash memory continues to maintain the file image even after power is lost.

```
Router#show flash
```

5. The routers Command Line Interface (CLI) maintains by default the last 10 commands you have entered in memory.   What command will view all of the past commands still in router memory at the same time?

```
Router#show history
```

6. What two commands will let you retrieve the previous command you typed?

```
Press the up arrow
or
<ctrl> P
```

7. What two commands will let you use the next command in the history buffer?

```
Press the down arrow
or
<ctrl> N
```

8. What command will let you view the status of the current layer 3 routed protocols running on your router?

```
Router#show protocols
```

9. What 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. How can you view the router's clock?

```
Router#show clock
```

11. What command will display a cached list of hosts and all of their interfaces IP addresses?

```
Router#show hosts
```

12. How can you view a list of all users who are connected to the router?

```
Router#show users
```

13. What command will give you detailed information about each interface?

```
Router#show interfaces
```

14. What command will show the global and interface-specific status of any layer 3 protocols?

```
Router#show protocols
```

Basic Show Commands Review

This review will require the use of the simulator to help with your responses.

1. You want to login to the Router and get to the Privileged Mode Prompt(#). What commands will perform this? ___________________

2. View your running configuration, what command would you use? ________________

3. You want to display the contents of the Flash memory. How would you do this? ________________

What is the name of the IOS in Flash? ____________________

How big is the IOS in Flash? ___________________________

How much Flash memory is free? _______________________

4. You want to display the information about all of the layer-three protocols that are currently being routed in the router. What command did you use to do this? ____________________

What protocols are enabled on the router?__________________

How many interfaces are UP ______ and how many are Administratively down? ____________

5. To view the list of the commands you have entered so far. What command would you use?_____________________

How many commands have you entered so far for this review? ______________

6. What keystroke(s) would enable you to bring up the previous command you entered? _______________ and ____________

7. What command will let you view critical information such as: router uptime, router platform type, operating system revision, amount of memory, number of interfaces and the configuration register? ____________________

Where is the IOS Stored?__________________

What is the Router Platform?________________

Total amount of NVRAM on the Router?________________

What is the configuration Register value? _______________

How many Ethernet Interfaces does this router have?______ How many Serial Interfaces?_________

8. To display the router’s time and date, what command will accomplish this?__________________

9. What time does the router think it is? _____________________

10. To list all the host entries in your router, what command will do this?__________________________________

11. What command will display all the users connected to your router?____________________________

12. What command will display the global and interface specific layer 3 information?

____

Show Lab Summary

This lab will introduce the Cisco Internetwork Operating System (IOS) command line interface (CLI). You will need to logon to a router and become familiar with the different levels of access on the router. You will also become familiar with the commands available to you in each mode (user or privileged) and the router help facility, history, and editing features.

Show Version

The show version command gives you a lot more information than at first you may think. Use show version 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
Krang Operating System Software
Router uptime is 2 minutes
System returned to ROM by power-on
System image file is "flash:c2500.bin"
[[[OUTPUT DELETED]]]
1 Ethernet/IEEE 802.3 interface(s)
1 Serial(sync/async) network interface(s)
1 ISDN Basic Rate interface(s)
32K bytes of non-volatile configuration memory.
4096K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

Layer 3 Interface Information

To view the layer 3 information for all interfaces currently configured on the router, use the show protocols command.

```
Router>show protocols
Global values:
Internet Protocol routing is enabled
BRI0 is administratively down, line protocol is Down
Ethernet0 is administratively down, line protocol is Down
Serial0 is administratively down, line protocol is Down
```

Flash Memory

Flash memory is a special kind of memory on the router that contains the operating system image file(s). Unlike regular router memory, Flash memory continues to maintain the file image even after power is lost.

```
Router>show flash

System flash directory:
File Length Name/status
1 3015588 c2500.bin
[3015652 bytes used, 1178652 available, 4194304 total]
4096K bytes of processor board System flash (Read/Write)
```

Running Configuration

The currently active configuration script running on the router is referred to as the *running-config* on the routers command-line interface. Note the privilege mode required. The running configuration script is **not** automatically saved on a Cisco router, and will be lost in the event of a power failure. The running configuration must be manually saved with the *copy* command (discussed in a later lab).

```
Router>
Router>enable
Router#show running-config
Building configuration...

Current configuration:
!
version 12.0
!
hostname Router
!
interface Serial0
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
interface Ethernet0
no ip address
shutdown
!
line con 0
line aux 0
line vty 0 4
!
end

Router#
```

Command History

The routers Command Line Interface (CLI) maintains by default the last 10 commands you have entered in memory, for later retrieval. You can change this default value. You cycle through previous router commands entered (since the last power loss), using one of two methods. To view all of the past commands still in router memory at the same time, use the show history command. For single line retrieval, use either the Arrow-Up (for previous command) and Arrow-Down (for next command), or Control-P (for previous command) and Control-N (for next command).

```
Router>show history
show version
show protocols
show flash
enable
show running-config
disable
show history
```

Clock

The router keeps its own clock that you can use to synchronize devices to. To view the clock use the show clock command.

```
Router#show clock
*00:38:35.755 UTC Mon Mar 1 1993
Router#
```

Host Table

You can create a list of host names on your router. You can view the entries (if any) by typing show hosts.

```
Router#show hosts
Default domain is not set
Name/address lookup uses static mappings

Host Flags Age Type Address(es)
Router#
```

Show users

The show users command displays users who are connected to the router.

```
Router#show users
Line User Host(s) Idle Location
* 0 con 0 idle 00:00:00

Router#
```

Show Interfaces

The show interfaces command will display statistics for all interfaces configured on the router.

```
Router#show interfaces
BRI0 is administratively down, line protocol is down
Hardware is BRI
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Conversations 0/0/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 5 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
--More--
```

Notice the **--More--** This means that there is more information pertaining to the last command. To view more commands line by line, press: enter To exit the output and return to the router prompt, press: e (this can be any letter, it's just easy to remember that e is for exit) To view more output one screen at a time, press the space bar.

Show Protocols

The show protocols command displays global and interface specific status of layer 3 routed protocols.

```
Router#show protocols
Global values:
Internet Protocol routing is enabled
BRI0 is administratively down, line protocol is down
Ethernet0 is administratively down, line protocol is down
Serial0 is administratively down, line protocol is down
Serial1 is administratively down, line protocol is down
Serial2 is administratively down, line protocol is down
```

Lab 4: CDP

Objective: To understand how the Cisco Discovery Protocol functions and what it takes for Cisco devices to be discovered.
Lab Equipment: We will be using Router 1 & Router 4. To select Router 1 click on the click on the button Router1 at the top of your screen.

1. On Router 1, enter global configuration mode

```
Router>enable
Router#conf t
Router(config)#
```

2. On Router 1, and change the hostname to **R1**

```
Router(config)#hostname R1
R1(config)#
```

3. Select Router 4 by clicking on the button Router4 at the top of your screen, and change the hostname to **R4**

```
Router>enable
Router#conf t
Router(config)#hostname R4
R4(config)#
```

Note: By default, all interfaces are shutdown (disabled).

4. Enable the Serial 0 interface on R1.

```
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
```

CDP allows devices to share basic configuration information. CDP will operate without any protocol specific information configured. CDP is enabled by default on all interfaces. CDP is a Data link Protocol occurring at Layer 2 of the OSI model.   This is important to understand because CDP is not routable. It can only traverse to directly connected devices.

7. On R1, type the command to give 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 we can use CDP to find out about directly connected neighbors.

8. On R1, type the command to provide 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    Holdtme    Capability    Platform    Port ID
R4              Serial 0          148          R           1700      Serial 0

R1#
```

The first device on the list for R1 is R4 via the Serial0 link. R1 is receiving CDP updates from R4, the updates tell R1 to hold on to the information for a specified amount of time. At the time this command was pressed there were 148 seconds left in the hold time for R1's update. If that time expires before receiving another update R1's information will be removed from the table. R4 is a 1000 series router; this is shown in the platform column. The final column, Port ID, is the port on the other device from which the updates are being sent.

9. On R1, type the command to provide more detailed information about directly connected neighbors.

```
R1#show cdp neighbors 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
```

This command shows devices one at a time. It is used to display Network Layer address information. At this point there are no configured IP, IPX or AppleTalk addresses so this field is blank. The command also displays IOS version information. Notice that the devices are listed in order. If one wants to find out information about a device further down the list, one would need to scroll down using the space bar.

10. On R1, type the command to provide information about the specific device "R4"

```
R1#show cdp entry R4
```

The below is 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#
```

This command gives the same information as the show cdp neighbor detail command, but 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 to see how often CDP updates are being sent and how long a recipient is to hold on to the update.

```
R1#show cdp
```

The below is sample output

```
Global CDP information:
Sending CDP packets every 60 seconds
Sending a holdtime value of 180 seconds
Sending CDPv2 advertisements is enabled
```

12. On R1, type the command to adjust the amount 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 hold on to the CDP information in the update. It is also a global parameter.

13. On R1, type the command to adjust the holdtime timer to 60 seconds.

```
R1#conf t
R1(config)#cdp holdtime 60
```

14. On R1, type the command to verify the changes made.

```
R1#show cdp
```

The below is 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#
```

If there are no other directly connected Cisco devices on the network, or to simply conserve bandwidth, CDP can be disabled.

15. On R1, type the command to disable CDP for the ENTIRE router.

```
R1#conf t
R1(config)#no cdp run
```

16. On R1, type the command to turn CDP back on for the ENTIRE router.

```
R1#conf t
R1(config)#cdp run
```

At times you may wish to disable CDP for a specific interface, for example a very low-bandwidth interface, or security reasons.

17. On R1, Disable CDP for only the specific interface Ethernet 0.

```
R1(config)#interface Ethernet 0
R1(config-if)#no cdp enable
```

18. On R1, type the command to verify that Ethernet 0 is no longer sending CDP updates (You can verify that the interface Ethernet 0 is not sending cdp updates because it does not show up as an entry in the output).

```
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
```

Basic Lab Summary

This lab is designed to introduce the Cisco Discovery Protocol (CDP) and some of its available commands.

The Cisco Discovery Protocol (CDP) Discovery Protocol

CDP allows devices to share basic configuration information without even configuring any protocol specific information and is enabled by default on all interfaces.

CDP is a Datalink Protocol occurring at Layer 2 of the OSI model.
This is important to understand because CDP is not routable and can only traverse to directly connected devices.

CDP allows you to view information such Operating System Version, Protocol Information, and much more.
This can be very handy for troubleshooting a variety of problems.

CDP Configuration by default it is enabled on the router and all interfaces.
The commands are simple:

Global Configuration Commands:

no cdp run	turn off CDP for the entire router
cdp run	(default) turn it on for the entire router
cdp timer 120	would change CDP to advertise every 120 seconds

Interface Configuration Commands:

cdp enable	(default) turn it on for the interface
no cdp enable	turn it off for interface

Show Commands:

show cdp interface	view interface settings,
show cdp neighbor	view directly connected neighbors
show cdp neighbor detail	view detailed information about neighbors
show cdp	general information

Lab 5: Extended Basics

Objective: To be able to view and configure some basic areas of the router.
Lab Equipment: We will be using "Router 1". To select "Router 1" click on the button labeled Router 1 at the top of the screen.

1. Get to the router prompt.

```
Router>
```

2. We wish to view the list of all available commands to us at this prompt. Enter the command that is used to view all the available commands.

```
Router>?
```

3. Now enter Privilege mode. This is the mode that lets you have total control of the router.

```
Router>enable
Router#
```

4. View the available commands in Privilege mode.

```
Router#?
```

5. We would like to configure the router. What command do we use to get into configuration mode?

```
Router#config terminal
Router(config)#
```

6. The Router's *Host Name* is used for local identification. When you log into the router you see the *Host Name* in front of the prompt (either the > or the #). This can be used to identify the location or function of the router. Set your Router's hostname to "**Krang**". What command do you use to configure the hostname?

```
Router(config)#hostname Krang
Krang(config)#
```

7. The enable password controls access to privilege mode. This is a VERY important password because in privilege mode you can make configuration changes. Set your enable password to "**boson**". What command will accomplish this?

```
Krang(config)#enable password boson
```

8. Let's test this password. Exit out of the router and try to enter privilege mode. Notice what password got you into privilege mode. Now type: conf term and proceed with the lab instructions in the next step.

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 show someone this file so that they can help you troubleshoot a problem you may inadvertently compromise the security of your systems by reveling the passwords. What command will create an encrypted password? Set the enable secret password to "**cisco**". What command did you use?

```
Krang(config)#enable secret cisco
```

10. You can now test this password by logging out of the router and then typing enable. The enable secret is an additional password over and above the enable password, in fact, it overrides the enable password. If you have set both passwords, the enable SECRET is the password you use to enter into privilege mode. The enable PASSWORD is still present but is now deactivated.

Extended Basic Commands Review

This review will require the use of the simulator to help with your responses.

1. You want to connect to Router 1 and view all the available commands. What command did you use to do this? _____________

2. To enter Privilege mode, what command would you use? ______________

3. You want to view all the available commands for Privilege mode. What command would you use? _________________

4. What command will get you into configuration mode? ______________

5. Set the Router's hostname to "**Krang**". What command would you use to do this? _______________

6. You will need to set the privilege mode password to "**boson**". What command will do this? _________________

7. Test the password by logging out of the router and then trying to enter enable mode.

8. To set the secret password to "**cisco**", what command would you use to do this? ________________

9. Logout of the router again and enter privilege mode. What password does the router require? _______________

Basic Lab Summary

This lab will introduce the Cisco Internetwork Operating System (IOS) command line interface (CLI). You will need to logon to a router and become familiar with the different levels of access on the router. You will also become familiar with the commands available to you in each mode (user or privileged) and the router help facility, history, and editing features.

User vs. Privileged Mode

User mode is indicated with the '>' next to the router name. You can look at settings but can not make changes from user mode. In Privilege mode (indicated by the '#', you can do anything. To get into privilege mode the keyword is enable**.**

```
Router>
Router>enable
Password:
Router#
```

HELP

To view all commands available from this mode type: ? This will give you the list of all available commands for the router in your current mode. You can also use the question mark after you have started typing a command. For example if you want to use a show command but you do not remember which one, typing in show ? will output all of the commands that you can use with the show command.

```
Router#show ?
access-expression List access expression
access-lists List access lists
backup Backup status
cdp CDP information
clock Display the system clock
cls DLC user information
compress Show compression statistics
configuration Contents of Non-Volatile memory
--More--
```

Configuration Mode

From privilege mode you can enter configuration mode by typing config term you can exit configuration mode type type **end** or pressing **<CTL>+z** (Press the 'Control' key and the letter 'Z' at the same time)

```
Router#config t
Router(config)#end
Router#
```

The Host Name

The Router's *Host Name* is used for local identification. When you log into the router you see the *Host Name*. This is also visible via LAT and CDP. However this is NOT used for TCP/IP address resolution.

```
Router(config)#hostname Krang
Krang(config)#
```

The Enable Password

The *enable password* controls access to privilege mode. This is a VERY important password because in privilege mode you can make configuration changes.

```
Krang(config)#enable password frodo
```

You can securely encrypt the *enable password*, by using the enable secret command.

```
Krang(config)#enable secret hobbits
```

If you have both passwords, the enable secret is the password used.

Additional Information

enable secret *password* Enable Secret Command

This command defines the enable secret password used to protect access to privileged exec commands. The password is case sensitive and can be defined on the router two different ways. A password set with the "enable password" command is stored as clear text, whereas a password set with "enable secret" is encrypted. For security, configuring the router with an enable secret is preferred. The enable secret always takes precedence if both enable secret and enable password are set.

Note: The unencrypted form of the password "cisco" is shown in the sample configurations. In an actual configuration, the password would appear in an encrypted form: (i.e. enable secret 7 13061E010803 -- where 7 denotes the encryption type and 13061E010803 is an encrypted form of the password cisco.) When entering or making changes to the enable secret, always type the password in its unencrypted form. Do not enter the encryption type (7); it is set automatically.

Lab 6: Setting the Banner MOTD (Message of the Day)

Objective: The goal of this lab is to setup a banner MOTD (Message of the Day). The MOTD is displayed when someone logs into the router. The banner can also be used to display information about the router itself or to display a security message.
Lab Equipment: We will be using Router 1. To select Router 1 click on the button labeled "Router 1" at the top of the screen.

1. Connect to Router 1 and enter Privileged mode.

```
Router>
Router>enable
Router#
```

2. Next enter configuration mode.

```
Router#config t
Router(config)#
```

3. Give the command to enter the banner message and hit return. After we type banner motd we need to enter a delimiting character. The delimiting character will be entered at the end of the banner text so the router will know when we are finished entering text for the banner. The easiest one to use is the "**z**".

```
Router(config)#banner motd z
Enter the text followed by the 'z' to finish
```

4. Now all text that we type, until we type the letter "**z**", will be stored as our banner. Enter the text "**You do not have permission to be here. This router eats hackers for lunch z**" and hit return. This will set our banner.

```
You do not have permission to be here. This router eats hackers for lunch! z
```

5. To view the banner exit out of configuration mode and then exit out of the system. Press return and you will get to see your 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: In this lab we will become familiar with the Router Configuration as well as be introduced to the Copy Commands Available to us in the Cisco IOS.
Lab Equipment: We will be using Router 1. To select Router 1 click on the button "Router" located at the top of the screen.

1. Get to the router prompt.

```
Router>
```

2. Enter Privilege Mode.

```
Router>enable
Router#
```

3. Show the active configuration in memory. The currently active configuration script running on the router is referred to as the *running-config* on the routers command-line interface. Note that privileged mode is required. 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
```

4. Try and show the configuration stored in NVRAM, this is your *startup-config*. We have not saved the configuration so there is not one to show.

```
Router#show startup-config
```

5. Copy the current active configuration to NVRAM. The current active configuration is in RAM and we would like to save it so that in the event the of a power outage the router will still boot up with our configuration.

```
Router#copy running-config startup-config
```

6. Now show the configuration stored in NVRAM.

```
Router#show startup-config
```

7. If we decided that we would like to start configuring the router from scratch we could erase the *startup-config* and reload the router. This will enable us to completely delete **ALL** configurations on the router so that we can start from scratch. What command will delete your configuration file in NVRAM.

```
Router#erase startup-config
```

8. Now that we have deleted our configuration lets reload the router. The router notices that you have a configuration and asks you if you would like to save it before you reload. We do not want to save it so we are going to select no.

```
Router#reload
```

9. After the router is done rebooting lets look at the startup configuration file again. Because we did not save it before we reloaded there is nothing there.

```
Router#show startup-config
```

10. Now lets change the hostname of your router to **Boson**. What command will do this?

```
Router#config terminal
Router(config)#hostname Boson
Boson(config)#exit
Boson#
```

11. After we changed the hostname we will now reload the router and when the router asks we will save our configurations.

```
Boson#reload
```

12. After reloading the router, the hostname of Boson appears in the prompt. If you do a show startup-config, nothing appears.

Basic Copy Commands Review

This review will require the use of the simulator to help with your responses.

1. Login to the Router and get to the Privileged Mode Prompt(#).

2. View your running configuration. ______________________

3. Show your configuration stored in NVRAM. Did you see anything?
____

4. Now copy your current active configuration into NVRAM. What command will do this? ____________________

5. Now again show your configuration stored in NVRAM. ______________________

6. Erase your configuration stored in NVRAM. ______________________

7. Reload the router and do not save your changes. What command did you use?

____

8. Now again show your configuration stored in NVRAM. ______________________

9. Change your routers hostname to **Boson**. What command did this?

____

10. Reload the router again but this time save your changes.

11. Notice that your hostname was not deleted. This is because we saved our configuration.

Copy Summary

Objective: Saving your configurations using the Copy command

Running Configuration

The currently active configuration script running on the router is referred to as the *running-config* on the routers command-line interface. Note the privilege mode required. 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>
Router>enable
Router#show running-config
Building configuration...

Current configuration:
!
version 12.0
!
hostname Router
!
interface Serial0
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
interface Ethernet0
no ip address
shutdown
!
```

```
line con 0
line aux 0
line vty 0 4
!
end

Router#
```

If you decide you would like to start configuring a router from scratch you will need to reload the router making sure you have deleted your *startup-config* file that is stored in NVRAM. To do this you will need to first erase the configuration file you have in NVRAM using the command erase startup-config. Next you will need to reload the router and do not save the configurations when asked.

```
Router#erase startup-config
Erasing the nvram filesystem will remove all files! Continue? [confirm]
[OK]
Erase of nvram: complete

Router#reload

System configuration has been modified. Save? [yes/no]: n
Proceed with reload? [confirm]
```

Lab 8: Introduction to interface configuration.

Objective: To understand how to enable interfaces on a router and what it takes for the interface to be UP.
Lab Equipment: We will be using Router 1 & Router 2. To select Router 1 click on the button labeled "Router1" located at the top of the screen.

1. On Router 1, enter global configuration mode

```
Router>enable
Router#conf t
Router(config)#
Router(config)#hostname Router1
```

2. We now wish to configure the Ethernet Interface. To do so, we must enter interface configuration mode. Type the command to enter interface configuration mode for Ethernet 0.

```
Router1(config)#interface Ethernet 0
Router1(config-if)#
```

3. What can you do to view all the commands available to you in interface configuration user mode?

Just type a '?' by itself.
this will show you all the available commands for that mode.

```
Router1(config-if)#?
```

4. Which command listed, looks like it would disable or turn off the interface?

```
shutdown        Shutdown the selected interface
```

5. We can often do the opposite of a command, by typing no in front of it. What command might enable this interface? Execute this command on Router 1 Ethernet 0 to enable the interface.

```
Router1(config-if)#no shutdown
```

6. Now add a description for this interface.

```
Router1(config-if)#description Ethernet interface on Router 1
```

7. To view your interface description exit back to privilege mode and do a show interface command. You should see your description under Ethernet 0.

```
Router1(config-if)#end
Router1#show interface
```

8. Now connect to Router 2 and enter the Ethernet 0 interface.

```
Router#conf t
Router(config)#hostname Router2
Router2(config)#interface Ethernet 0
```

9. Now enable the interface.

```
Router2(config-if)#no shutdown
```

10. Now that the interfaces on both sides of our Ethernet connection are enabled you should be able to see one another by CDP. Use the command show cdp neighbors to view all directly connected Cisco Routers.

```
Router2(config-if)#end
Router2#show cdp neighbors
```

Configuring and Examining Interfaces Summary

Examining the Interfaces

Routers can have many types of interfaces, such as token ring, FDDI, Ethernet, Serial, ISDN etc. We often want to view the status and settings. There are a few important commands that we need to know. Show interfaces is one of the more important commands.

```
Router#show interfaces
Ethernet0 is administratively down, line protocol is down
Hardware is Lance, address is 0060.5cc4.f445 (bia 0060.5cc4.f445)
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
```

[OUTPUT OMITTED]

This command will produce output about each interface. In this case we see that Ethernet 0 is administratively down. That means that it is turned off with the shutdown command.

Ethernet 0 is	Line protocol is	Meaning
administratively down	down	The interface is turned off with the shutdown command
up	down	Cable is connected but keep alives are not being received.
down	down	Cabling problem or no clock rate set on DCE. Or other router interface is shutdown.
up	up	connected and receiving keep alives. This is what we want!!!

You can view particular interfaces with the command: show interface serial 0**.** Or any other interface. A handy command is show ip interface brief**.**

```
Router#show ip int brief
Interface IP-Address OK? Method Status Protocol
Ethernet0 unassigned YES not set administratively down down
PCbus0 unassigned YES not set administratively down down
Serial0 unassigned YES not set up down
Router#
```

This allows you to rapidly see the status of all the interfaces.

Examining the Controllers

Controllers are the part of the interface that makes the physical connection. The most important to us is what kind of cable is attached to a Serial interface.

A **DTE** (data terminating equipment) cable is the normal cable you should use. Being DTE means you expect the other end to providing **clocking.**

A **DCE** (data circuit-terminating equipment) means that this device must provide the clocking on the wire.

The show controllers command will allow you to see if you are DCE or DTE.

```
Router#show controllers serial 0

HD unit 0, idb = 0xA2B58, driver structure at 0xA7020
buffer size 1524 HD unit 0, V.35 DCE cable
cpb = 0x42, eda = 0x2140, cda = 0x2000
```

Configuring the Interfaces

If an interface is administratively down. You must enter configuration mode, then enter interface configuration mode, and lastly, issue the command no shutdown.

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#no shutdown
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
%LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Router(config-if)#end
Router#
```

If your interface is the DCE, you must provide clocking using the clock rate command.

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0
Router(config-if)#clock rate 56000
Router(config-if)#end
Router#
```

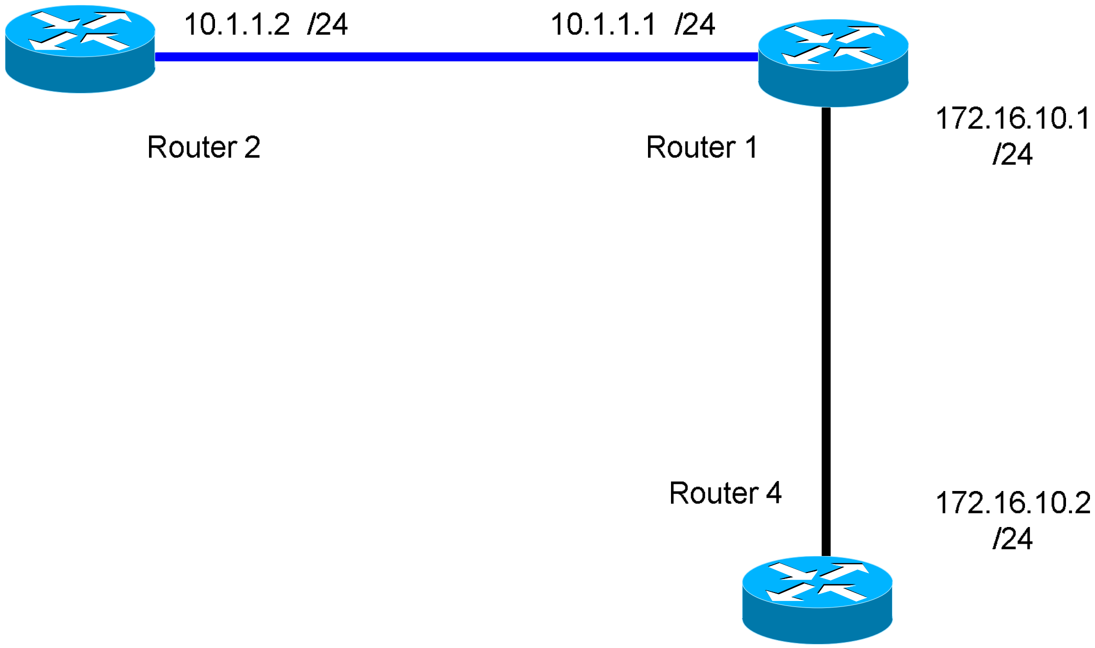
It is often useful to put a description of what the interface is used for using the description command.

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#description My Connection to the Engineering Hub
Router(config-if)#end
Router#
```

You can view your changes using show running-config or show interfaces or show controllers

Lab 9: Introduction to IP (Internet Protocol)

Objective**:** We will configure Routers 1, 2, and 4 with IP addresses and ping between them to test connectivity.



Configuring IP Addresses:

1. First you want to connect to router one and assign a hostname of Router1

```
Router>enable
Router#conf t
Router(config)#hostname Router1
Router1(config)#
```

2. What mode must you be in to set the IP address on an interface?

```
Router1(config)#interface ethernet 0
Router1(config-if)#
```

3. What command will set the IP address on the Ethernet 0 interface to 10.1.1.1 255.255.255.0?

```
Router1(config-if)#ip address 10.1.1.1 255.255.255.0
```

4. Now you need to enable the interface. What command did you use?

```
Router1(config-if)#no shutdown
```

5. Now set the IP address on the S0 interface of Router1 to 172.16.10.1 255.255.255.0

```
Router1(config)#Interface Serial 0
Router1(config-if)#ip address 172.16.10.1 255.255.255.0
Router1(config-if)#no shut
```

6. Next click on the button Router2 at the top of your screen.

7. Assign a hostname of Router2.

```
Router>enable
Router#conf t
Router(config)#hostname Router2
Router2(config)#
```

8. Set the IP address for the Ethernet 0 interface to 10.1.1.2 255.255.255.0

```
Router2(config)#interface Ethernet 0
Router2(config-if)#ip address 10.1.1.2 255.255.255.0
```

9. Enable the interface.

```
Router2(config-if)#no shutdown
```

10. Now click on the button Router4 at the top of your screen.

11. Assign an Hostname of Router4 and an IP address of 172.16.10.2 255.255.255.0 on the serial 0 interface.

```
Router>enable
Router#conf t
Router(config)#hostname Router4
Router4(config)#interface serial 0
Router4(config-if)#ip address 172.16.10.2 255.255.255.0
```

12. Make sure you enable the interface.

```
Router4(config-if)#no shutdown
```

13. Connect back to Router1

14. Try and ping Router2's Ethernet interface

```
Router1#ping 10.1.1.2
```

15. Try and ping Router4's Serial 0 interface.

```
Router1#ping 172.16.10.2
```

16. What command will let you verify that your interfaces line state and protocol state are up?

```
Router1#show ip interface brief
```

17. View your running configuration and verify that the IP addresses appear.

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

18. View detailed IP information about each interface.

```
Router1#show ip interface
```

IP Addresses Review

This review will require the use of the simulator to help with your responses.

1. First you want to connect to router one assign a hostname of Router1. What command did you use?________________________________

2. What mode must you be in to set the IP address on an interface?__________________

3. What command will set the IP address on the Ethernet 0 interface to 10.1.1.1 255.255.255.0? ____________________________

4. Now you need to enable the interface. What command did you use?
____

5. Now set the IP address on the S0 interface of Router1 to 172.16.10.1 255.255.255.0? What command will do this? _____________________________

6. Next click on the button Router2 at the top of your screen.

7. Assign a hostname of Router2. What command did you use?___________________

8. Set the IP address for the Ethernet 0 interface to 10.1.1.2 255.255.255.0. What command did you use?_________________________

9. Enable the interface. What command does this? ___________________

10. Now click on the button Router4 at the top of your screen.

11. Assign an IP address of 172.16.10.2 255.255.255.0 on the serial 0 interface. What command did you use?_________________________

12. Make sure you enable the interface. What command did you use?_________________________

13. Connect back to Router1

14. Try and ping Router2's Ethernet interface. What command allows you to ping?

____

15. Try and ping Router4's Serial 0 interface. What address is the Serial 0 interface?

____

16. What command will let you verify that your interfaces are up and up?

____

17. View your running configuration and verify that the IP addresses appear. What command allows you to view your running-configuration?

____

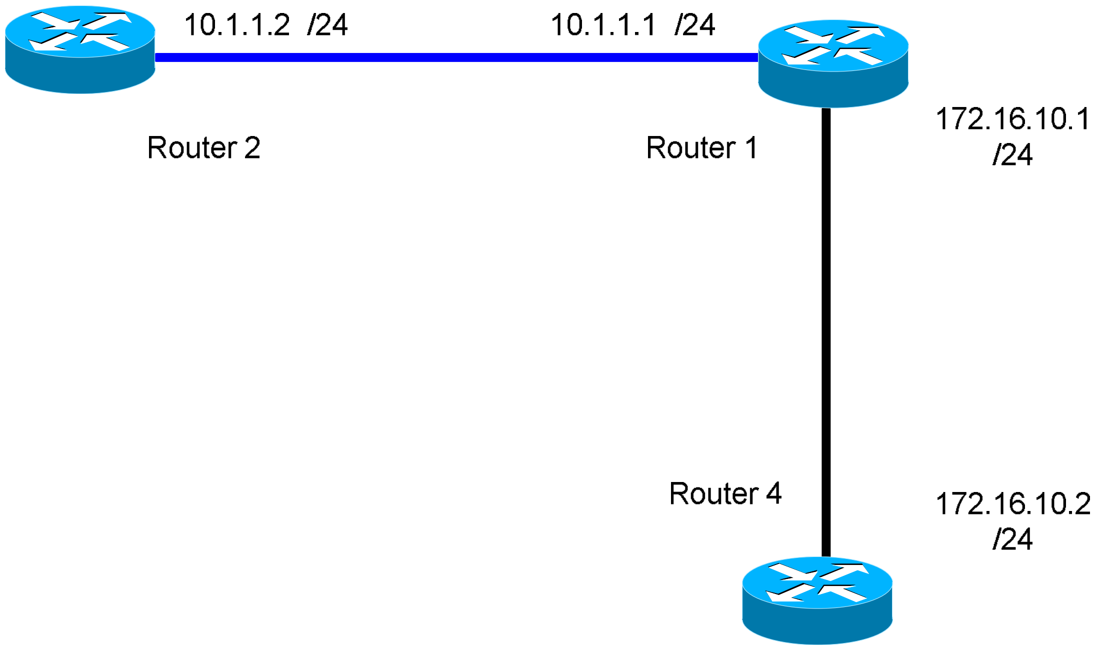
18. View detailed IP information about each interface. What command will do this?

____

Basic IP Configuration and Verification Summary

IP addressing is very easy to configure on a Cisco router. Although the calculation of IP addresses, subnet masks and host can be rather difficult.

The syntax to place an IP address on the interface is:
ip address *ip-address mask*
Given the routers below, we wish to configure IP addresses on Router1 and Router2.



Remember the /24 means 255.255.255.0.   For your convenience here is a handy table:

Slash	Dotted Decimal	Slash	Dotted Decimal	Slash	Dotted Decimal
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0
/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254

Let's start configuring Router 1

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#ip address 10.1.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#int s0
Router(config-if)#ip address 172.16.10.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#end
%SYS-5-CONFIG_I: Configured from console by console
Router#
```

We can view the IP addresses on the interface:

```
Router#sh ip interface brief
Interface      IP-Address     OK?   Method     Status                   Protocol
BRI0           unassigned     YES   manual     administratively down    down
Ethernet0      10.1.1.1       YES   manual     administratively down    down
Serial0        172.16.10.1    YES   manual     administratively down    down



Router#
```

We have assigned an IP address to each interface but the interface is still administratively down because we have not executed a `no shutdown` command on each interface.

Now you should go to each of the interfaces and type `no shutdown`, this should turn the interfaces to up.

Connect to Router 2.   We would also like to add IP addresses to the Ethernet 0 interface.

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#ip address 10.1.1.2 255.255.255.0
Router(config-if)#no shut
%SYS-5-CONFIG_I: Configured from console by console
Router(config)#exit
Router#exit
```

Now that we have an IP address on both sides of this Ethernet connection, we can now jump into ping.

PING

PING, the Packet Inter Net Groper, allows a user to test basic connectivity. The syntax is:

ping *ip-address*

The router will send out five echo requests to the destination IP address, if it receives a reply, it will note it with an '!', if no reply is received it will note it with a '.'.

A successful ping:

```
Router#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/37/44 ms


Router#
```

A failed ping:

```
Router#ping 2.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#
```

Ping is one of the most commonly used test tools. PING uses the Internet Control Message Protocol (ICMP) to communicate with other routers.

When pinging devices for the first time, ping may fail on the first try. This is because the router has not completed its ARP resolution.

You can also view your IP addresses using the command show running-config or show ip interface.

Lab 10: ARP

View your ARP Table:

1. First you want to connect to Router 1 and view your ARP table. What command will do this?

```
Router>enable
Router#show arp
```

2. Next you need to assign an IP address of 10.1.1.1 /24 to the Ethernet 0 interface.

```
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. Now view your ARP table again. How many entries do you have now?

```
Router(config)#exit
Router#show arp
```

4. Select Router2 from the button menu.

5. Set Router 2's Ethernet 0 interface IP address to 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
```

6. You should now have a connection between your Router 1 and Router 2 Ethernet interfaces. To ensure that the connection is functional ping your Router 1 Ethernet 0 IP address.

```
Router(config-if)#exit
Router(config)#exit
Router# ping 10.1.1.1
```

7. Now view your ARP table and notice the entry. What address is it and how was it learned?

```
Router#show arp
```

8. Now that you have built an ARP table, go ahead and clear it.

```
Router#clear arp
```

9. View your ARP table one last time and notice what entries are there.

```
Router#show arp
```

ARP Table Review

This review will require the use of the simulator to help with your responses.

1. First you want to connect to Router 1 and view your ARP table what command will do this?____________________

2. Next you need to assign an IP address of 10.1.1.1 /24 to the Ethernet 0 interface. What command will set the IP address on your Ethernet interface?
____

3. Now view your ARP table again. How many entries do you have now? __________

4. Select Router2 from the button menu.

5. Set Router 2's Ethernet 0 interface IP address to 10.1.1.2 /24

6. You should now have a connection between your Router 1 and Router 2 Ethernet interfaces. To ensure that the connection is functional ping your Router1 Ethernet 0 IP address. What address did you ping? ____________________________

7. Now view your ARP table and notice the entry. What address is it and how was it learned? ____________________________

8. Now that you have built an ARP table, go ahead and clear it. What command will clear the ARP table? ____________________

9. View your ARP table one last time and notice what entries are there. How many entries do you have? _____________________

ARP Summary

In this lab you will view the ARP table that is stored in the router. You will also learn how to clear the router's table as a troubleshooting technique. The router stores detailed information obtained from other devices on the network. The information it collects is what is referred to as a MAC address and IP address. This information can get corrupted from time to time which will cause the router to have packet-delivery problems. When this happens we must clear the ARP table and have it rebuilt.

You can view the ARP table using the command show arp. This displays detailed information about interfaces that are learning MAC addresses. Looking at the table below we can see that we learn the IP address and MAC (hardware) address of each Ethernet interface. The age is how long we have had the information and the interface is what interface we learned this information from.
Notice the age time on the 1.1.1.4 address, this is because it is the IP address of the Ethernet port that is connected to the router.

```
Router#show arp
Protocol Address     Age (min)   Hardware          Addr     Type Interface
Internet 1.1.1.2        207         0000.0c32.f57d    ARPA     Ethernet0
Internet 1.1.1.4          -         0060.7062.e040    ARPA     Ethernet0
Router#
```

You must be in privileged mode to clear the ARP table. After entering privileged mode the command to clear the ARP table is clear arp. After you have cleared the ARP table you can view it again using the show arp command. Notice all the entries have disappeared with the exception of the directly connected interfaces of the router.

```
Router#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 1.1.1.2 - 0060.7062.e040 ARPA Ethernet0
Router#
```

Lab 11: Creating a Host table

Objective: Our goal of this lab is to become familiarized with the Routers Host Table. We can use a host table to set names for commonly used IP addresses. This will help us with troubleshooting or make our lives easier if we are testing a lot with pings.
Lab Equipment: We will be using Router 1. To select Router 1 click on the button labeled "Router 1" at the top of the screen.

1. Connect to Router 1 and enter Privilege mode

```
Router>
Router>enable
Router#
```

2. Enter configuration mode and set the hostname to **California**.

```
Router#config t
Router(config)#
Router(config)#hostname California
California(config)#
```

3. Set an IP address of 195.42.36.10 255.255.255.240 on Ethernet 0’s interface, make sure to enable the interface.

```
California(config)#interface ethernet 0
California(config-if)#
California(config-if)#ip address 195.42.36.10 255.255.255.240
California(config-if)#no shutdown
```

4. Now that we have an IP address, connect to Router 2 and enter privilege mode.

```
Router>enable
Router#
```

5. Enter Configuration mode and set the hostname to **Tampa**.

```
Router#config t
Router(config)#
Router(config)#hostname Tampa
Tampa(config)#
```

6. Set an IP address of 195.42.36.12 255.255.255.240 on Ethernet 0’s interface, make sure to enable the interface.

```
Tampa(config)#interface ethernet 0
Tampa(config-if)#
Tampa(config-if)#ip address 195.42.36.12 255.255.255.240
Tampa(config-if)#no shutdown
```

7. Exit out of interface mode and now we are going to make a host table entry. We do not want to have to type California's Ethernet 0 IP address every time we try to ping it so we are going to set a host table entry for California using the IP address 195.42.36.10.

```
Tampa(config-if)#exit
Tampa(config)#
Tampa(config)#ip host California 195.42.36.10
```

8. After we have done this we should be able to ping California's Ethernet 0 IP address just by typing ping California.

```
Tampa(config#)exit
Tampa#ping California
```

9. You can now verify that the entry is made in the router by using the command show hosts.

```
Tampa#show hosts
```

Lab 12: Static Routes

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

Goals:

1) Set our hostname and get our interfaces up.
2) Ping our directly connected interfaces
3) Configure static routes for our topology
4) View our routing table
5) Verify that we can ping all routers

IP Addresses: Please set these IP addresses on the interfaces of your routers.

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	12.5.10.1 255.255.255.0		12.5.10.2 255.255.255.0

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

2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP addressing setup correctly lets get into configuring our static routes on each router. First connect to Router 1. Let's think about what we are doing. We will need to establish static routes to any location that is not directly connected. Router 1 is directly connected to both Router 2 and Router 4 so it will not need any static routes. Next connect to Router 4.

4. Now enter configuration mode and think about what the static route command will be. We know we currently cannot get to Router 2 because it is not directly connected. So now let's think about Router 4 and the network it is connected to. Off of its Serial interface is network 12.5.10.0 that is connected to Router 1. Router 1 is also connected to network 10.1.1.0 that we would also like to access. In this case we will need a static route for network 10.1.1.0. On Router 4 what command would you use to establish a static route to network 10.1.1.0?

```
Router4(config)#ip route 10.1.1.0 255.255.255.0 12.5.10.1
```

We established a route to network 10.1.1.0 on our router. Now whenever a packet of information is destined for network 10.1.1.0 it will be sent to the router with IP address 12.5.10.1 which in this case is Router 1.

5. Lets see what we accomplished. When we are on Router 4 and we know we can ping Router1's Serial interface but we could not ping Router1's Ethernet interface. We should now have established a route to network 10.1.1.0. To make sure we understand what our route did for us, try and ping Router 1’s Serial 0, Router 1’s Ethernet 0 and Router 2’s Ethernet 0.

```
Router4#ping 12.5.10.1
Router4#ping 10.1.1.1
Router4#ping 10.1.1.2
```

6. Why couldn't we ping? If you think about a packet going through the network it leaves Router 4's S0 interface destined for 10.1.1.2. The packet knows this is on the 10.1.1.0 network so it is going to first go to 12.5.10.1 because of our static route. When it gets to 10.1.1.1 (Router 1) the Router looks at its Routing table and knows that it is directly connected to network 10.1.1.0 off its Ethernet interface, so it sends the packet out there. Router 2 picks up that packet and wants to respond back with a "Hey, You found me". It looks at the Source IP address and it is 12.5.10.2 (Router 4's Serial 0 interface). When it goes to send the packet it does not have a route to network 12.5.10.0 so it drops the packet. This is why you did not get the !!!!! as a successful response.

7. Just to make sure our static route worked view your routing table to see if it is in there.

```
Router4#show ip route
```

8. To get the static routes to work we need to connect to Router 2 and give a static route back to Router 4's network. What command will set a static route on Router 2 for the network 12.5.10.0?

```
Router2(config)#ip route 12.5.10.0 255.255.255.0 10.1.1.1
```

That means anything we are sending to network 12.5.10.0 will go to 10.1.1.1 first.

9. Connect back to Router 4 and make sure you can ping all of the interfaces we have working.

Router1 Ethernet 0 : 10.1.1.1
Router1 Serial 0 : 12.5.10.1
Router2 Ethernet 0: 10.1.1.2

10. Lets look at the routing table on Router 2 and talk about its entry.

```
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
```

Looking at the second line we see the "S" denoting the Static route. Next we see the destination network and its subnet info. The [1/0] is showing the administrative distance (by default "1") and the metric (in this case hop count) which is 0. Via just says the address to go to which in this case is 10.1.1.1.

Lab 13: RIP

Objective: We will configure Routers 1, 2, and 4 with IP addresses and RIP Routing Protocol

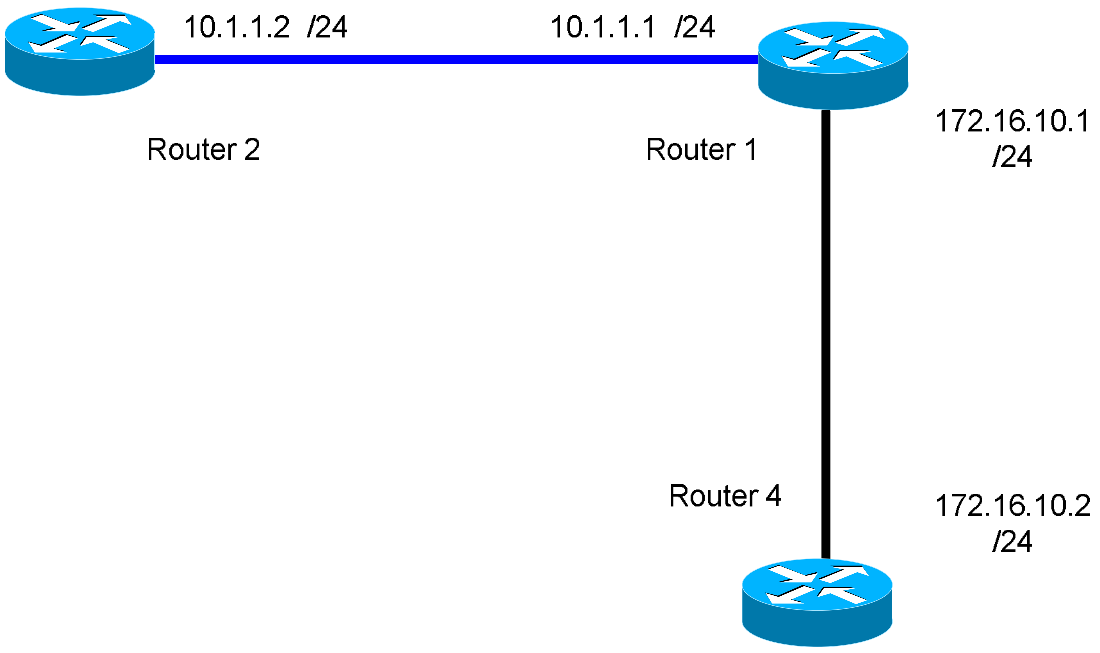
Goals:

1) Set our hostname and get our interfaces up.
2) Configure RIP routing protocol
3) Select the directly connected networks
4) View our routing table
5) View the RIP protocol information

IP Addresses: Please set these IP addresses on the interfaces of your routers.

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.255.0		172.16.10.2 255.255.255.0

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



2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP addressing setup correctly lets get into configuring RIP as our routing protocol. This is very easy to do; first we need to get into router configuration mode on Router 1. What command does this?

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

4. Now enter the command to configure the router for RIP.

```
Router1(config)#router rip
Router1(config-router)#
```

5. Add the network(s) that Router 1 is directly connected to. What statements will do this?

```
Router1(config-router)#network 10.0.0.0
Router1(config-router)#network 172.16.0.0
```

6. Now you have Router 1 configured for RIP, connect to Router 2 and enter configuration mode.

```
Router2#
Router2#config Terminal
Router2(config)#
```

7. Add RIP routing protocol to the router. What command does this?

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

8. Add the network(s) that Router 2 is directly connected to. What statements will do this?

```
Router2(config-router)#network 10.0.0.0
```

9. Now you have Router 2 configured for RIP, connect to Router 4 and enter configuration mode.

```
Router4#
Router4#config Terminal
Router4(config)#
```

10. Add RIP routing protocol to Router 4. What command does this?

```
Router4(config)#router rip
Router4(config-router)#
```

11. Add the network(s) that Router 4 is directly connected to. What statements will do this?

```
Router4(config-router)#network 172.16.0.0
```

12. Now we should have RIP running on all three of our routers. Type <ctrl> Z to exit to privileged mode and let's see if we can ping non-directly connected routers. From Router 2 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Let's try it!

```
Router2#ping 172.16.10.2
```

13. Next let's connect to Router 4 and ping Router 2's Ethernet 0 interface with IP address 10.1.1.2

```
Router4#ping 10.1.1.2
```

14. If you can ping both devices, CONGRATULATIONS you are routing. If you were not successful, trace yourself back through the steps. Now let's view our routing table on our Router 4. What command will do that?

```
Router1#show ip route
```

15. Lets view the specific IP routing protocol information on our router. What command will do this?

```
Router1#show ip protocols
```

RIP Review

This review will require the use of the simulator to help with your responses.

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

2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP address setup correctly let's get into configuring RIP as our routing protocol. This is very easy to do; first we need to get into router configuration mode on Router 1. What command does this?

4. Now enter the command to configure the router for RIP.

5. Add the networks that Router 1 is directly connected to. What statements will do this?
____

6. Now that you have Router 1 configured for RIP, connect to Router 2 and enter configuration mode. ___________________________

7. Add RIP routing protocol to the router. What command does this?
____

8. Add the networks that Router 2 is directly connected to. What statements will do this?
____

9. Now that you have Router 2 configured for RIP, connect to Router 4 and enter configuration mode. ___________________________

10. Add Rip routing protocol to the router. What command does this?
____

11. Add the networks that Router 4 is directly connected to. What statements will do this?
____

12. Now we should have RIP running on all three of our routers. Type <ctrl> Z to exit to privileged mode and lets see if we can ping non directly connected routers. From Router 1 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Lets try it!

13. Next lets connect to Router 4 and ping Router 1's Ethernet 0 interface with IP address 10.1.1.2

14. If you can ping both devices, CONGRATULATIONS you are routing. If you were not successful, trace yourself back through the steps. Now lets view our routing table on our Router 4. What command will do that? ___________________________ How many RIP Routes do you see? ______

15. Lets view the specific routing protocol information on our router. What command will do this? ___________________________

What version of RIP are you sending? ___________________________

How often are the updates being sent? ___________________________

What networks are you routing for? ___________________________

What is the default administrative distance? ___________________________

RIP Summary

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is a standards-based, distance-vector, interior gateway protocol (IGP) used by routers to exchange routing information. RIP uses hop count to determine the best path between two locations. Hop count is the number of routers the packet must go through till it reaches the destination network. The maximum allowable number of hops a packet can traverse in an IP network implementing RIP is 15 hops. In a RIP network, each router broadcasts its entire RIP table to its neighboring routers every 30 seconds. When a router receives a neighbor's RIP table, it uses the information provided to update its own routing table and then sends the updated table to its neighbors. This procedure is repeated by each router and results in a state referred to as network convergence, in which all routers have an identical view of the internetwork topology.

The rest of this lab is a walk through lab that you can complete on the program or just follow through the steps. The output displayed below demonstrate how to complete a Rip Configuration. Some time is taken to explain in detail what each command does.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router1
eRoute1r(config)#int e0
Router1(config-if)#ip address 10.1.1.1 255.255.255.0
Router1(config-if)#no shut
%LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Router1(config-if)#exit

Router1(config)#int s0
Router1(config-if)#ip address 172.16.10.1 255.255.0.0
Router1(config-if)#no shut
%LINK-3-UPDOWN: Interface Serial0, changed state to up
%LINK-3-UPDOWN: Interface Serial0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
Router1(config-if)#exit
Router1(config)#
```

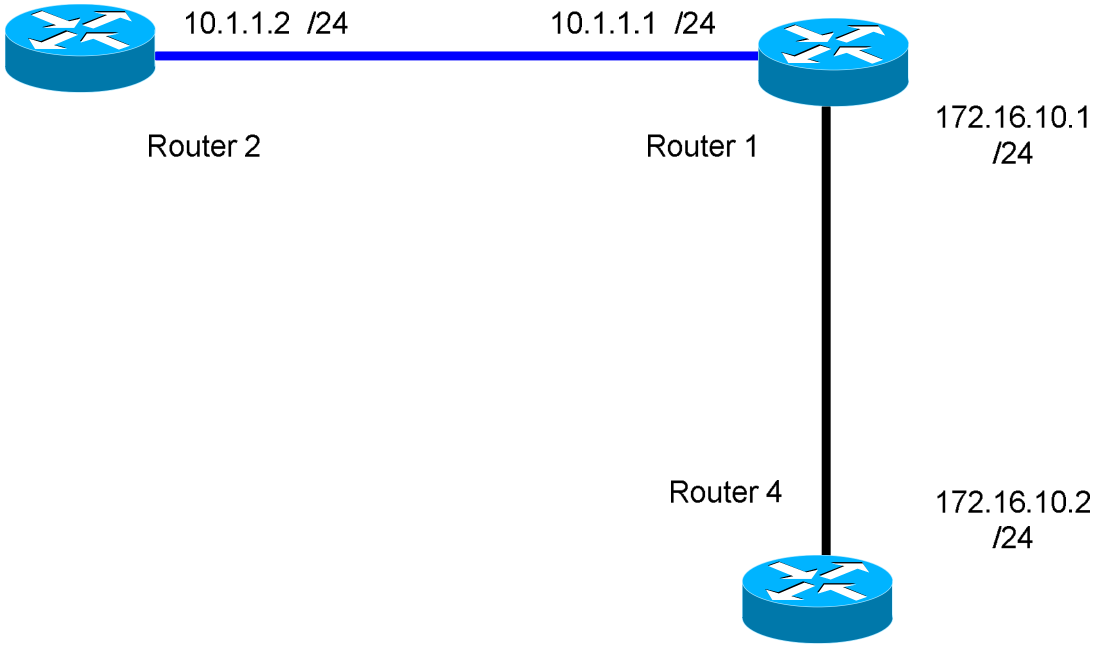
RIP version 1 is **classful**, meaning it does not include the subnet mask in its routing table updates. RIP version 2 is classless and includes the subnet information.

We first want to configure Router1 for RIP. To enable RIP as the routing protocol we only need to type: router rip, from there we can see this below in the router output. Notice the new mode we have entered Router1(config-router)# that tells us we are configuring the router.

```
Router1(config)#router rip
Router1(config-router)#
```

Network Statements / Network Numbers

Now that we have RIP running on our Router we need to tell the router which networks it is connected to. We do this by using the network statement. What this means is every interface of our router that is directly connected to an active network needs a network number. We will have some networks using the same ip addressing schemes with different subnets, and some are using entirely different addressing schemes. Look at the diagram below. In this diagram we have three different kinds of addressing schemes. Let's look at these in more detail. On Router 1 we have an IP address of 10.1.1.1 with a /24 subnet mask. Since RIP is classful you are only required to enter the class part of the address for the network statement. For example on Router1 we have already issued the command router rip, we then need to specify the directly connected networks to Router1 so the router can advertise these routes in its routing table. To do this we would only need to type: network 10.0.0.0 now we have not told the router about the network on its serial interface, to do this we would type: network 172.16.0.0 Lets look at Router 2, what network statements would you need to use on this router ____(see the answer below the diagram.)



The answer is network 10.0.0.0. The network statement for the serial link is the same for Router1 and Router2. For the network statement for the Ethernet link you had to remember that a 172 address was a class B address, for this network statement you used the classful portion of the address 172.16.0.0.

Now that we understand the network command lets enter it on our Router1.

```
Router1(config-router)#network 172.16.0.0
Router1(config-router)#network 10.0.0.0
Router1(config-router)#
```

If you notice we only entered 10.0.0.0 for our network statement, this is because 10.0.0.0 is a Class A address and RIP only uses the classful portion of the address. Now we have configured Router1 for RIP lets connect to Router2 and get it setup.

We need to connect to Router2 and follow the same instructions. Let’s select Router2 from the Window pull down menu. When we connect we are going to set the hostname to Router2, then set the IP addresses to the table above and configure RIP.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router2
Router2(config)#int e0
Router2(config-if)#ip address 10.1.1.2 255.255.255.0
Router2(config-if)#no shut
00:17:25: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Router2(config-if)#exit
Router2(config)#
```

Now add the RIP stuff!

```
Router2(config)#router rip
Router2(config-router)#network 10.0.0.0
Router2(config-router)#exit
Router2(config)#exit
Router2#
```

We should now have RIP running on our network between Router1 and Router2. Now we need to get Router4 setup.

We need to connect to Router4 and follow the same instructions. Let’s select Router4 from the Window pull down menu. When we connect we are going to set the hostname to Router4, then set the IP addresses to the table above and configure RIP.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router4
Router4(config)#int s0
Router4(config-if)#ip address 172.16.10.2 255.255.0.0
Router4(config-if)#no shut
00:20:35: %LINK-3-UPDOWN: Interface Serial0, changed state to up
Router4(config-if)#exit
Router4(config)#
```

Now add the RIP stuff!

```
Router4(config)#router rip
Router4(config-router)#network 172.16.0.0
Router4(config-router)#exit
Router4(config)#exit
Router4#
```

Show Commands

Now that we have RIP running on our entire network lets verify that it is receiving routes. To do this we will be using some show commands. The most common one is show ip route. This displays all entries in the routing table. If we do this on our Router 4 we will see the route to our directly connected Router1, we will also see routes to the other routers we have setup on the network. Let’s take a look at our routing table, to do this type: show ip route from the privilege mode.

Lets look at the first entry R 10.1.1.0/24 [120/1] via 172.16.10.2, 00:00:21, Serial0. It starts off with R this says it is a RIP route, it then says the destination network with subnet mask in this case it is 10.1.1.0 with a /24 (255.255.255.0) subnet mask. Next it gives 120/1, the 120 is the administrative distance, RIP's default administrative distance is 120. Administrative distance is considered the trustworthiness of the route. If you have two routing protocols with the same route the router will pick the route with the lower number. The 1 is the hops required to get to the destination network. The next piece of information is via 172.16.10.1, which is the next hop address it must go to. The last item is that this information was learned via Serial0.

Another great command is show ip protocols. This displays information about the IP routing protocols you have enabled. Let’s type the command: show ip protocols and see what we get.

```
Router4#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 12 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: rip
Default version control: send version 1, receive any version
Interface Send Recv Key-chain
Ethernet0    1 1 2
Serial0      1 1 2
Routing for Networks:
172.16.0.0
Routing Information Sources:
Gateway Distance Last Update
172.16.10.2 120 00:00:09
Distance: (default is 120)

Router4#
```

Looking at the output in detail we see we are sending updates every 30 seconds. We know RIP is a distance vector routing protocol so it exchanges its entire routing table every 30 seconds. We also see our network statements are working by noticing the networks are both under the Routing for Networks area. The last area to notice is the Distance which we said was administrative distance. This tells us the default is 120 and that is what we are using.

Lab 14: Troubleshooting RIP

Objectives: We will set up an ip addresses on Routers 1, 2 and 4 with RIP as the routing protocol. We will then Observe Routing Activity using *debug ip rip*. The routes will be examined using *show ip route* command.

1. Establish the configurations outlined in the table below before continuing.

Device	Router 1	Router 2	Router 4
Hostname	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. Configure Rip routing protocol on all routers using the proper network statements.

3. Check to make sure you receive the routes on all routers with the show ip route command.

4. Once you have received the routes execute the command debug ip rip from privilege mode on Router 1.

5. Observe the output on the routers terminal screen.   (The output could take up to 60 seconds to appear.)

6. To turn off the debug command us the no keyword in front of the command (ie no debug ip rip) or to turn off all debugging use “undebug all” or “u all”

7. View the routing table entries on Routers 2 and 4. Notice the Administrative distance and metrics for these routes.

8. Make sure you can ping all devices in the network. If you cannot ping then you will need to troubleshoot the router configurations to ensure you set everything up correctly.

Lab 15: IGRP

Objective: We will configure Routers 1, 2, and 4 with IP addresses and IGRP Routing Protocol

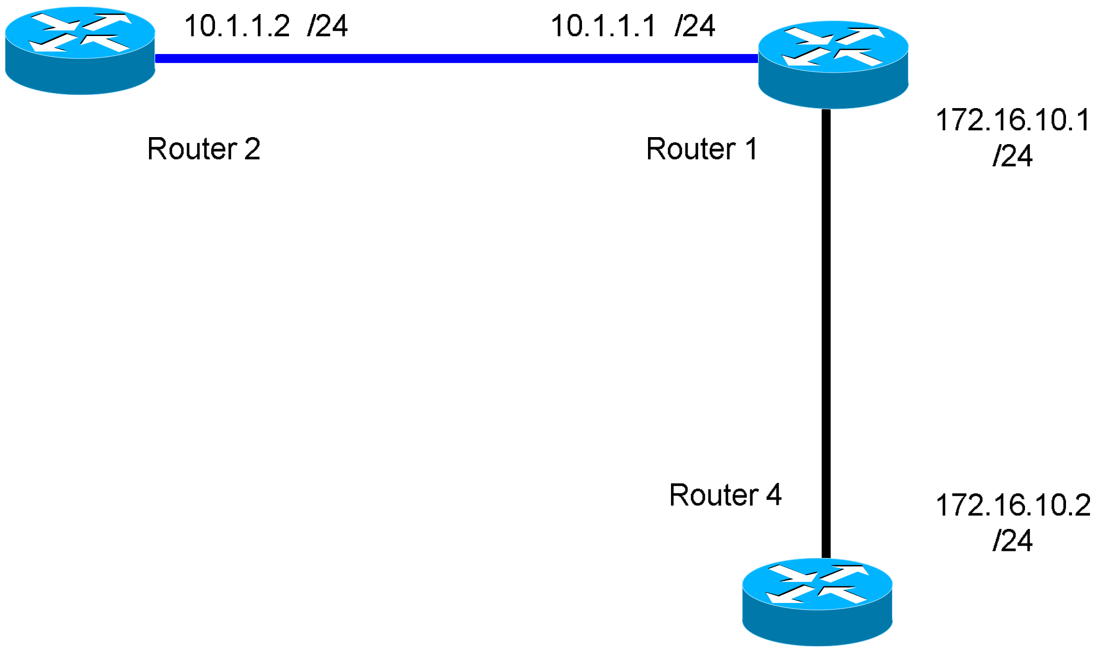
Goals:

1) Set our hostname and get our interfaces up
2) Configure IGRP routing protocol
3) Select the directly connected networks
4) View our routing table
5) View the IGRP protocol information

IP Addresses: Please set these IP addresses on the interfaces of your routers.

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.255.0		172.16.10.2 255.255.255.0

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



2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP addressing setup correctly lets get into configuring IGRP as our routing protocol. This is very easy to do; first we need to get into router configuration mode on Router 1. What command does this?

```
Router1#
Router1#config Terminal
Router1(config)#
```

4. Now enter the command to configure the router for IGRP use the autonomous system number **100**.

```
Router1(config)#router igrp 100
Router1(config-router)#
```

5. Add the network(s) that Router 1 is directly connected to. What statements will do this?

```
Router1(config-router)#network 10.0.0.0
Router1(config-router)#network 172.16.0.0
```

6. Now that you have Router 1 configured for IGRP, connect to Router 2 and enter configuration mode.

```
Router2#
Router2#config Terminal
Router2(config)#
```

7. Add IGRP routing protocol to the router remember to use the same Autonomous System #. What command does this?

```
Router2(config)#router igrp 100
Router2(config-router)#
```

8. Add the network(s) that Router 2 is directly connected to. What statements will do this?

```
Router2(config-router)#network 10.0.0.0
```

9. Now that you have Router 2 configured for IGRP, connect to Router 4 and enter configuration mode.

```
Router4#
Router4#config Terminal
Router4(config)#
```

10. Add IGRP routing protocol to the router remember to use the same Autonomous System #. What command does this?

```
Router4(config)#router igrp 100
Router4(config-router)#
```

11. Add the network(s) that Router 4 is directly connected to. What statements will do this?

```
Router4(config-router)#network 172.16.0.0
```

12. Now we should have IGRP running on all three of our routers. Type <ctrl> Z to exit to privileged mode and let's see if we can ping non-directly connected routers. From Router 2 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Let's try it!

```
Router2#ping 172.16.10.2
```

13. Next let's connect to Router 4 and ping Router 2's Ethernet 0 interface with IP address 10.1.1.2

```
Router4#ping 10.1.1.2
```

14. If you can ping both devices, CONGRATULATIONS you are routing. If you are not successful, trace yourself back through the steps. Now let's view our routing table on our Router 4. What command will do that?

```
Router1#show ip route
```

15. Lets view the specific IP routing protocol information on our router. What command will do this?

```
Router1#show ip protocols
```

IGRP Review

This review will require the use of the simulator to help with your responses. This lab is exactly the same as the lab you just completed except it does not include any commands to lead you in the right direction.

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

2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP address setup correctly lets get into configuring IGRP as our routing protocol. This is very easy to do, first we need to get into router configuration mode on Router 1. What command does this?

4. Now enter the command to configure the router for IGRP with the Autonomous System # 100.

5. Add the networks that Router 1 is directly connected to. What statements will do this?
____

6. Now you have Router 1 configured for IGRP connect to Router 2 and enter configuration mode. ___________________________

7. Add IGRP routing protocol to the router with the Autonomous System # 100. What command does this? ___________________________

8. Add the networks that Router 2 is directly connected to. What statements will do this?
____

9. Now that you have Router 2 configured for IGRP connect to Router 4 and enter configuration mode. ___________________________

10. Add IGRP routing protocol to the router with the Autonomous System # 100. What command does this? ___________________________

11. Add the networks that Router 4 is directly connected to. What statements will do this?
____

12. Now we should have IGRP running on all three of our routers. Type <ctrl> Z to exit to privileged mode and let's see if we can ping non-directly connected routers. From Router 1 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Let's try it!

13. Next let's connect to Router 4 and ping Router 1's Ethernet 0 interface with IP address 10.1.1.2

14. If you can ping both devices, CONGRATULATIONS you are routing. If you were not successful, trace yourself back through the steps. Now let's view our routing table on our Router 4. What command will do that? ___________________________ How many IGRP Routes do you see? ______

15. Lets view the specific routing protocol information on our router. What command will do this? ___________________________

How often are the updates being sent? ___________________________

What networks are you routing for? __________________________

What is the default administrative distance? __________________________

IGRP Summary

Interior Gateway Routing Protocol (IGRP) is a standards-based, distance-vector, interior gateway protocol (IGP) used by routers to exchange routing information. IGRP uses a composite metric of bandwidth and delay to determine the best path between two locations. The metric can also be administratively configured to factor in the Maximum Transmission Unit (MTU), Reliability, and load for the link. In a IGRP network, each router broadcasts its entire IGRP table to its neighboring routers every 90 seconds. When a router receives a neighbor's IGRP table, it uses the information provided to update its own routing table and then sends the updated table to its neighbors. This procedure is repeated by each router and results in a state referred to as network convergence, in which all routers have an identical view of the Internetwork topology.

The rest of this lab is a walk through lab that you can complete on the program or just follow through the steps. The output displayed below demonstrates how to complete an IGRP Configuration. Some time is taken to explain in detail what each command does.

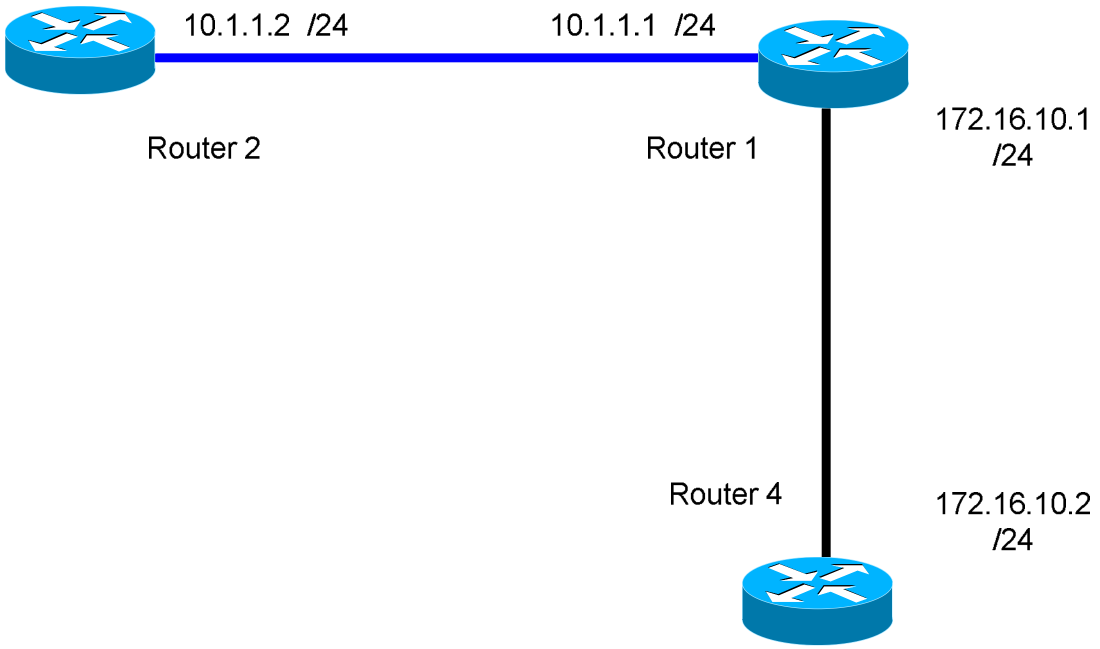
```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#ip address 10.1.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
00:35:15: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Router(config)#hostname Router1
Router1(config)#int s0
Router1(config-if)#ip address 172.16.10.1 255.255.0.0
Router1(config-if)#no shut
00:35:16: %LINK-3-UPDOWN: Interface Serial0, changed state to up
Router1(config-if)#exit
00:35:16: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
```

IGRP is classful, meaning it does not include the subnet mask in its routing table updates. So now let's go ahead and start the lab.

We first want to configure Router1 for IGRP. To enable IGRP as the routing protocol we only need to type: router IGRP *AS*. The AS stands for a Autonomous System number. An Autonomous System is defined as a network under a common administration with a common routing policy. You will need to use the **SAME** autonomous system number on every router that you would like to share its routing table with. We can see this below in the router output. Notice the new mode we have entered Router1(config-router)# that tells us we are configuring the router.

```
Router1(config)#router IGRP 100
Router1(config-router)#
```

Now that we have IGRP running on our Router we need to tell the router which networks it is connected to. We do this by using the network statement. What this means is every interface of our router that is directly connected to an active network needs a network number. We will have some networks using the same ip addressing schemes with different subnets, and some are using entirely different addressing schemes. Look at the diagram below. In this diagram we have three different kinds of addressing schemes. Let's look at these in more detail. On Router 1 we have an IP address of 10.1.1.1 with a /24 subnet mask. Since IGRP is classful you are only required to enter the class part of the address for the network statement. For example on Router1 we have already issued the command router IGRP, we then need to specify the directly connected networks to Router1 so the router can advertise these routes in its routing table. To do this we would only need to type: network 10.0.0.0. We have not told the router about the network on its serial interface, to do this we would type: network 172.16.0.0. Let's look at Router 2. What network statement would we need to use on this router?
____ (see the answer below the diagram.)



The answer is network 10.0.0.0. The network statement for the ethernet link is the same for Router1 and Router2. On Router1 what network statement would you need for the serial link? For this network statement you used the classful portion of the address 172.16.10.1 which would be just network 172.16.0.0.

Now that we understand the network command lets enter it on our Router1.

```
Router1(config-router)#network 172.16.0.0
Router1(config-router)#network 10.0.0.0
Router1(config-router)#
```

If you notice we only needed to enter 10.0.0.0 for our network statement, this is because 10.0.0.0 is a Class A address and IGRP only uses the classful portion of the address. Now we have configured Router1 for IGRP lets connect to Router2 and get it setup.

We need to connect to Router2 and follow the same instructions. Let's select Router2 from the Window pull down menu. When we connect we are going to set a hostname to Router2, then set the IP addresses to the table above and configure IGRP.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router2
Router2(config)#int e0
Router2(config-if)#ip address 10.1.1.2 255.255.255.0
Router2(config-if)#no shut
Router2(config-if)#exit
01:23:17: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
01:23:18: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
Router2(config)#
```

Now add the IGRP stuff!

```
Router2(config)#router IGRP 100
Router2(config-router)#network 10.0.0.0
Router2(config-router)#exit
Router2(config)#exit
Router2#
```

We should now have IGRP running on our network between Router1 and Router2. We need to get Router4 setup.

We need to connect to Router4 and follow the same instructions. Let's select Router4 from the Window pull down menu. When we connect we are going to set a hostname to Router4, then set the ip addresses to the table above and configure IGRP.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router4
Router4(config)#int s0
Router4(config-if)#ip address 172.16.10.2 255.255.0.0
Router4(config-if)#no shut
Router4(config-if)#exit
01:23:17: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
01:23:18: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
Router4(config)#
```

Now add the IGRP stuff!

```
Router4(config)#router IGRP 100
Router4(config-router)#network 172.16.0.0
Router4(config-router)#exit
Router4(config)#exit
Router4#
```

Now that we have IGRP running on our entire network lets verify that it is receiving routes. To do this we will be using some show commands. The most common one is show ip route. This displays all entries in the routing table. If we do this on our Router 4 we will see the route to our directly connected Router1. Let's take a look at our routing table, to do this type: show ip route from the privilege mode.

Lets look at the first entry I 10.1.1.0/24 [100/651] via 172.16.10.1, 00:00:21, Serial0. It starts off saying it is an IGRP route, it then says the destination network with subnet mask in this case is 10.1.1.0 with a /24 (255.255.255.0) subnet mask. Next it gives 100/651 the 100 is the administrative distance, IGRP's default administrative distance is 100. Administrative distance is considered the trustworthiness of the route. If you have two routing protocols with the same route the router will pick the route with the lower number. The 651 is the calculated metric, which is based on bandwidth delay. The next piece of information is via 172.16.10.1, which is the next hop address it must go to. The last item is that this information was learned via Serial0.

Another great command is show ip protocols. This displays information about the ip routing protocols you have enabled. Let's type the command: show ip protocols and see what we get.

```
Router4#show ip protocols
Routing Protocol is igrp 100
Sending updates every 90 seconds, next due in 12 seconds
Invalid after 270 seconds, hold down 280, flushed after 630
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
IGRP maximum hopcount 100
IGRP maximum metric variance 1
Redistributing: igrp 100
Routing for Networks:
172.16.0.0
Routing Information Sources:
Gateway Distance Last Update
172.16.10.2 100 00:00:09
Distance: (default is 100)

Router4#
```

Looking at the output in detail we see that we are sending updates every 90 seconds. We know IGRP is a distance vector routing protocol so it exchanges its entire routing table every 90 seconds. We also see our network statements are working by noticing the networks are both under the Routing for Networks area. The last area to notice is the Distance which we said was administrative distance. This tells us the default is 100 and that is what we are using.

Lab 16: PPP with CHAP Authentication

Objective: To understand how PPP encapsulation works and how to secure the connection with CHAP Authentication.
Lab Equipment: We will be using Router 1 & Router 4. To select Router 1 click on the button Router1 at the top of your screen. Connect to Router 1 now.

1. Enter global configuration mode on Router 1

```
Router>enable
Router#conf t
Router(config)#
```

2. On Router 1 change the hostname to **R1**

```
Router(config)#hostname R1
R1(config)#
```

3. The enable secret passwords will be used along with the hostname to pass over to the other router. You must use the same password on both sides to get CHAP to work. Set R1's Enable secret password to **sameone**

```
R1(config)#enable secret sameone
```

4. On R1, set a username for R4 with password **myboson**

```
R1(config)#username R4 password myboson
```

5. On R1 assign an IP address of 10.1.1.1 255.255.255.0 to the Serial 0 interface.

```
R1(config)#interface Serial 0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
```

6. On R1 set the encapsulation for the serial interface to PPP

```
R1(config-if)#encapsulation ppp
```

7. On R1 set PPP authentication to CHAP on the serial interface

```
R1(config-if)#ppp authentication chap
```

8. Now make sure the Serial 0 interface is enabled

```
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

9. Select Router 4 from the buttons and enter privilege mode, then change the hostname to **R4**

```
Router#config t
Router(config)#hostname R4
R4(config)#
```

10. Now on R4 we will need to set a secret password of **sameone**

```
R4(config)#enable secret myboson
```

11. Now we need to add a username for R1 with password **sameone**. What command will accomplish this?

```
R4(config)#username R1 password sameone
```

12. Now enable the Serial 0 interface on R4 and assign an IP address of 10.1.1.2 255.255.255.0.

```
R4(config)#interface Serial 0
R4(config-if)#ip address 10.1.1.2 255.255.255.0
R4(config-if)#no shutdown
```

13. We need to set the Serial 0 PPP authentication to CHAP on R4

```
R4(config-if)#ppp authentication chap
```

14. The last thing to do is enable PPP Encapsulation on the Serial 0 interface of R4 now watch the State change to up!

```
R4(config-if)#encapsulation ppp
R4(config-if)#exit
```

15. To verify the configuration is correct ping Router 1 Serial 0 from Router 4.

```
R4(config-if)#exit
R4(config)#exit
R4#ping 10.1.1.1
```

PPP with CHAP Authentication Summary

PPP (Point-to-Point Protocol) is a protocol for communication between two computers using a serial interface, typically a personal computer connected by phone line to a server. For example, your Internet server provider may provide you with a PPP connection so that the provider's server can respond to your requests, pass them on to the Internet, and forward your requested Internet responses back to you. PPP uses the Internet protocol (IP) (and is designed to handle others). It is sometimes considered a member of the TCP/IP suite of protocols. Relative to the Open Systems Interconnection (OSI) reference model, PPP provides layer 2 (data-link layer) service. Essentially, it packages your computer's TCP/IP packets and forwards them to the server where they can actually be put on the Internet.

PPP is a full-duplex protocol that can be used on various physical media, including twisted pair or fiber optic lines or satellite transmission. It uses a variation of High Speed Data Link Control (HDLC) for packet encapsulation.

PPP is usually preferred over the earlier de facto standard Serial Line Internet Protocol (SLIP) because it can handle synchronous as well as asynchronous communication. PPP can share a line with other users and it has error detection that SLIP lacks. Where a choice is possible, PPP is preferred.

CHAP (Challenge-Handshake Authentication Protocol) is a more secure procedure for connecting to a system than the Password Authentication Procedure (PAP). Here's how CHAP works:

1. After the link is made, the server sends a challenge message to the connection requestor. The requestor responds with a value obtained by using a one-way hash function.
2. The server checks the response by comparing its own calculation of the expected hash value.
3. If the values match, the authentication is acknowledged; otherwise the connection is usually terminated.

At any time, the server can request that a new challenge message be sent by the connected party. Because CHAP identifiers are changed frequently and because authentication can be requested by the server at any time, CHAP provides more security than PAP. RFC1334 defines both CHAP and PAP.

Configuring PPP w/CHAP on a Cisco Router

The interface command to enable PPP is:

```
encapsulation ppp
```

Place this on both ends and that is it. However, to enable authentication, we need to add the interface command:

```
ppp authentication chap
```

To both routers, the routers will now require authentication over the link. They will attempt to log in with their HOSTNAME as their USERNAME and their ENABLE password as their CHAP PASSWORD.   We must create an entry in the router that matches the remote routers username and password (global config):

```
username Other_Router password Other_enable_pass
```

That is all there is to basic PPP.

Our Samples:

(R1)s0----------s0(R2)

PPP Without CHAP

Router 1:

```
hostname R1
interface serial 0
   encapsulation PPP
   no shutdown
```

Router 2:

```
hostname R2
interface serial 0
   encapsulation PPP
   no shutdown
```

PPP With CHAP default names and password

Router 1:

```
hostname R1
enable secret toast1
username R2 password cool2
interface serial 0
   encapsulation PPP
   ppp authentication chap
   no shutdown
```

Router 2:

```
hostname R2
enable secret cool2
username R1 password toast1
interface serial 0
   encapsulation PPP
   ppp authentication chap
   no shutdown
```

A Good link for covering the PPP/CHAP authentication is found here:
http://www.cisco.com/warp/public/471/understanding_ppp_chap.html

Lab 17: Connectivity tests with Traceroute

Purpose: The purpose of this lab is to give you experience with the trace route command. This command is used to map the list of devices IP addresses it goes through to go from one device to another.

1. Establish the configurations outlined in the table below before continuing.

Device	Router 1	Router 2	Router 4
Hostname	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 established the proper ip addresses, enable rip routing across all three routers and make sure you use the proper network statements.

3. Now that we have RIP routing enabled across all three routers test by issuing some ping commands.

4. From Router 1 ping your directly connected Routers and their interfaces, which are Router 2 Ethernet 0 and Router 3 Ethernet 0

```
Router1#ping 192.168.1.2
Router2#ping 192.168.2.2
```

5. Since we have Rip routing enabled we should be able to ping non-directly connected routers. Connect to Router 2 and ping Router 4's Serial 0 interface.

```
Router2# ping 192.168.2.2
```

6. The goal behind the trace route command is to help you troubleshoot and determine the path a packet is taking to get to a destination device. In this example we have three routers and only one path to the destination. View the output of the command by tracing the route from Router 2 to Router 4's Serial 0 interface.

```
Router2#traceroute 192.168.2.2
```

Observe the output from the trace route command. It lists Router 1's Ethernet 0 and then the destination ip address we are tracing to. This means that it first leaves Router 2's Ethernet 0 goes through Router 1's Ethernet 0 before reaching Router 4's Serial 0.

Lab 18: Saving Router Configurations.

Objective: This lab will teach you how to backup your router's configuration in case you accidentally delete it or your routers die.
Lab Equipment: We will be using Router 4 & eStation 1.

1. Connect to "Router 4"and enter configuration mode.

```
Router>enable
Router#conf t
Router(config)#
```

2. Assign the hostname **Tampa** to Router 4.

```
Router(config)#hostname Tampa
Tampa(config)#
```

3. Enter the Ethernet 0 interface.

```
Tampa(config)#interface ethernet 0
Tampa(config-if)#
```

4. Assign the IP address of 24.37.2.1 255.255.255.0.

```
Tampa(config-if)#ip address 24.37.2.1 255.255.255.0
```

5. Enable the interface.

```
Tampa(config-if)#no shutdown
```

6. Connect to PC1 1 by selecting it from the list of additional devices. Type the command 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 Router 4's Ethernet 0 IP address (24.37.2.1).

```
C:> winipcfg
```

7. Ping the connection to make sure you have correct connectivity.

```
c:> ping 24.37.2.1
```

8. Connect back to Router 4 and exit interface mode. Copy the running-configuration to the tftp server on PC 1.

```
Tampa(config-if)#exit
Tampa(config)#exit
Tampa#
Tampa# copy running-config tftp
```

9. Enter the IP address of the tftp server and the name of the configuration file that we will store on the tftp server.

```
24.37.2.252
Tampa_config
After you press return, the router will take a minute
to establish the connection, then you will see it
copy the configuration file and tell you how long it took.
```

10. Next connect back to PC 1 and type the command that will show the configurations that are stored on the tftp server. (Note: This command does not work on normal PC's.)

```
C:>show tftp-configs
```

If you see the configuration in the list you are finished.
This completes the lab.

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

Saving Router Configurations Review

This review will require the use of the simulator to help with your responses.

1. Connect to Router 1 and enter enable mode. Next enter configuration mode and assign the router a hostname of "Tampa".

What command did you use to change the router name? ___________________

2. Next enter the Ethernet 0 interface.

Command _________________

3. Assign an IP address of 24.37.2.1 255.255.255.0 to router Tampa and enable the interface.

4. From eStation 1, type the command to configure it's IP address and default gateway.

eStation 1 Command_________________________________

5. Set the IP address of eStation 1 to 24.37.2.252 255.255.255.0 with a default gateway of Router Tampa's Ethernet 0 IP address (24.37.2.1). Then ping the connection.

6. Connect back to Router Tampa and enter privileged mode.

Commands_________________________________

7. Copy the running configuration to the tftp server on eStation 1.

What command did you use? ____

8. Connect back to eStation 1 and enter the command used to view the configuration files that are stored on the tftp server.

Command____

Copy Summary

Objective: Saving your configurations using the Copy command

Running Configuration

The currently active configuration script running on the router is referred to as the *running-config* on the routers command-line interface. Note the privilege mode required. 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>
Router>enable
Router#show running-config
Building configuration...

Current configuration:
!
version 12.0
!
hostname Router
!
interface Serial0
no ip address
shutdown
!
interface BRI0
no ip address
shutdown
!
interface Ethernet0
no ip address
shutdown
!
line con 0
line aux 0
line vty 0 4
!
end

Router#
```

Lab 19: Loading Router Configurations

Objective: To become familiar with loading router configurations.
Lab Equipment: We will be using Router 4. To select Router 4 click on the button Router 4 located at the top of the screen.

NOTE: You must have just completed the Saving Router Configurations lab (Lab 18) to complete this lab. If you have not completed the saving router configurations lab, then please do it now so that you can continue with this lab.

1. Now that you have your config stored on the tftp server we want to change the hostname of our **Router**. This will prove to us that the config was copied from the TFTP server. Log onto Tampa and enter configuration mode.

```
Tampa#config t
Tampa(config)#
```

2. Set the hostname to **Bad_Router**.

```
Tampa(config)#hostname Bad_Router
Bad_Router(config)#
```

3. Now we want to copy the config we stored on the TFTP server into our running configuration.

```
Bad_Router(config)#exit
Bad_Router#copy tftp running-config
```

4. When the router prompts us for the IP address enter the IP address of the TFTP server.

```
Address or name of remote host []?24.37.2.252
```

5. Next you will need to enter the name of the config file to grab from the TFTP server.

```
Source filename []?Tampa_config
```

6. Now the router will download the config and load it into the running-config. Afterwards the hostname is restored to what it was when we saved the configuration.

```
Tampa#
```

Lab 20: Copying and Pasting Configurations

Goal: The goal of this lab is to get you introduced to saving, reloading and pasting modified configurations from within the simulator.

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 get that command working. The simulator works the same way. When you are working with the simulator you can easily switch between devices using the buttons across the top of the main window. The simualtor offers some built in saving and loading options.

1. Set the hostname of Router 1 to Router1

2. Select the File->Save Single Device Config option. The program will ask for a filename use Router1 and click save. Save the files somewhere you will remember.

3. After you have saved the files exit the simulator program and then reload it again.

4. Select the File -> Load Single Device Config (overwrite) option. This will ask for the filename that you previously saved. Select the file 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 you now have your hostname back.

6. There are two other options under the File menu that offer similar functionality. The File-> Save Multi Device Config option and File-> Load Multi Device Config option. These two options will save and load the configs for all the devices you have loaded into the simulator. These options will take a complete snapshot of all devices and save them to files.

7. Another neat feature about our saved files is the capabilities to edit them easily. Minimize the program and double click on one of the .rtr files 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 Notepad.

8. Notepad will launch with the show run displayed of Router 1. You will see the hostname command a couple lines down. Change this line from hostname Router1 to hostname Miami. Save your changes.

9. Now repeat step 4 and observe the hostname change.

10. There is one other feature the simulator offers. If you have a configuration that you have created and you would like to “Paste” the configuration into the routers, the program offers a tool to allow you to do this.

11. First make sure you are working on Router 1. Select the File->Paste Real Router Configs menu option. This will bring up a window that will allow you to paste configuration files you would like to have executed on Router 1. 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 out the commands above click the “OK” button. You will quickly see the router execute the commands. Notice the hostname of the router will change back to Router 1.

13. Execute the show ip interface brief command on Router 1 to see that the IP address has been set for Ethernet 0.

Lab 21: ISDN: Integrated Services Digital Network

Objective: To understand how to setup ISDN on Cisco Routers.
Lab Equipment: We will be using Router 1 & Router 2. To select Router 1 click on the button Router1 at the top of your screen.

1. Setup the connection between Router 1 (hostname Router1), and Router 2 (hostname Router2) using the BRI ports. Router 1 will have an IP address of **42.34.10.1** with a **255.255.255.0** subnet mask .

```
Router>enable
Router#conf t
Router(config)#hostname Router1
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)#end
```

Now connect to Router 2 by clicking on the Router 2 button and give it an IP address of **42.34.10.121** with a **255.255.255.0** subnet mask.

```
Router>enable
Router#conf t
Router(config)#hostname Router2
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)#end
```

2. Lets go back to Router 1 and start to setup ISDN. The first thing we need to do is specify the ISDN switch we will be using. If you used the simulator defaults, the switch-type is basic-ni. There are two different places you can tell the router what ISDN switch-type you are using, you can specify the command globally for all BRI interfaces on the router, or you can make the switch-type interface specific. We are going to make it global, so now enter the switch-type globally on your router.

```
Router1#conf t
Router1(config)#isdn switch-type basic-ni
```

3. We will need to give some specific information to this BRI interface. The first thing we will give it is the ISDN SPID (Service Profile Identifier). The SPID needs to be set on your BRI interface with 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 used the defaults for the ISDN Switch your SPID for Router 1 will be 32177820010100.

```
Router1(config)#interface Bri 0
Router1(config-if)#isdn spid1 32177820010100
```

4. Now that you have setup the switch-type and SPID you should have layer 1 connectivity. Layer 1 connectivity is the establishment of connectivity between the ISDN switch and the router. To verify you have layer 1 connectivity use the show isdn status command and make sure you have Multi-Frame Established in layer 2.

```
Router1(config-if)#exit
Router1(config)#exit
Router1#show isdn status
```

5. Now we want to setup the number we will need to dial on the ISDN switch to establish a layer 3 connection, this is called the dialer string. Set the dialer string on your Router 1 BRI 0, if you are using the default configuration use "**7782001**"

```
Router1#config t
Router1(config)#interface bri 0
Router1(config-if)#dialer string 7782001
```

6. Because ISDN costs money when the connection is up we only want to have an active connection when we are using it. There are multiple ways of doing this, we will be using dialer groups and dialer lists. A dialer list is a list either permitting or denying traffic. This means we will specify a dialer-list of *protocol ip permit*. What that means is any IP traffic will be permitted. To setup a dialer list use the command dialer-list in the global config mode.

```
Router1(config-if)#exit
Router1(config)#dialer-list 1 protocol ip permit
```

7. Now we have a dialer-list but we have not associated that list with any interfaces. We will need to add that list to our ISDN BRI interface using the dialer-group command.

```
Router1(config)#interface bri 0
Router1(config-if)#dialer-group 1
```

Let’s go over what we have done again:

1. First we told the router what type of ISDN switch it would be connecting to.

2. Second we specified the SPID we will be using to communicate with the ISDN switch.

3. Third we told the router what number it will be dialing to connect with the other router.

4. Fourth we created a global dialer list to allow our IP traffic

5. Then we associated the dialer list with our ISDN BRI 0 interface using the Dialer-group command.

8. Now we have setup ISDN on Router 1 we need to do the same steps for Router 2, but with some slight modifications. Use the steps below to try and configure Router 2 for ISDN. If you would prefer to be guided step by step then skip to step 9..

1. Specify the ISDN switch we would be connecting to (Basic-ni).

2. Specify the SPID we will be using to communicate with the ISDN switch (32177820020100).

3. Configure the router with the number it dials in order to connect with the other router through the ISDN Switch (7782002).

4. Create a global dialer list to allow our IP traffic

5. Associate the dialer list with our ISDN BRI 0 interface using the Dialer-group command.

If you got it to work, CONGRATULATIONS jump down to step 15!

9. Now connect to Router 2 using the button at the top labeled Router 2. The first thing we need to do is specify the ISDN switch we will be using. If you used the simulator defaults, the switch-type is basic-ni. Enter the switch-type globally on your router.

```
Router#conf t
Router(config)#hostname Router2
Router2(config)#isdn switch-type basic-ni
```

10. Next we will need to give the SPID for this interface. If you used the simulator defaults for the ISDN Switch your SPID for Router 2 will be 32177820020100.

```
Router2(config)#interface Bri 0
Router2(config-if)#isdn spid1 32177820020100
```

11. Now that you have setup the switch-type and SPID you should have layer 1 connectivity. To verify that you have layer 1 connectivity use the show isdn status command and make sure you have Multi-Frame Established in layer 2.

```
Router2(config-if)#exit
Router2(config)#exit
Router2#show isdn status
```

12. Now we want to setup the number we will need to dial on the ISDN switch to establish a layer 3 connection, this is called the dialer string. Set the dialer string on your Router 2 BRI 0. If you are using the default configuration use "7782002"

```
Router2#config t
Router2(config)#interface bri 0
Router2(config-if)#dialer string 7782002
```

13. Setup a dialer-list on your Router 2 to permit all IP traffic. To setup a dialer list use the command dialer-list in the Global Config mode.

```
Router2(config-if)#exit
Router2(config)#dialer-list 1 protocol ip permit
```

14. Now we have a dialer-list but we have not associated that list with any interfaces. We will need to add that list to our ISDN BRI interface using the dialer-group command.

```
Router2(config)#interface bri 0
Router2(config-if)#dialer-group 1
```

15. If we have both Routers configured for ISDN lets see if we can ping the other side. From Router 2 you want to ping Router 1's BRI 0 interface (IP address 42.34.10.1)

```
Router2(config-if)#exit
Router2(config)#exit
Router2#ping 42.34.10.1
```

16. If you have a successful ping, Congratulations you have ISDN working. Verify this with the show isdn status command.

```
Router2#show isdn status
```

Look at your layer 3 settings, it should say you have 1 Active layer 3 Call, you can also see in layer 2 that your SPID is valid. This is good for troubleshooting.

17. You can also view your settings in your *running-config*.

```
Router2#show running-config
```

Lab 22: IPX

1. First you want to connect to Router1 and enable IPX as the routing protocol.

```
Router>enable
Router#confg t
Router(config)#ipx routing
```

2. Next you need to assign an IPX network of **B201** to your Ethernet 0 interface.

```
Router(config)#interface Ethernet 0
Router(config-if)#ipx network B201
Router(config-if)# no shutdown
Router(config-if)#exit
```

3. Now assign an IPX network of **AAA** to your Ethernet 1 interface.

```
Router(config)#interface Ethernet 1
Router(config-if)#ipx network AAA
Router(config-if)# no shutdown
Router(config-if)#exit
```

4. Select Router2 from the button menu

5. Enable IPX routing on Router 2.

```
Router>enable
Router#confg t
Router(config)#ipx routing
```

6. Set Router 2's Ethernet 0 interface to match Router 1's Ethernet 0 IPX network of B201

```
Router#conf terminal
Router(config)#interface Ethernet 0
Router(config-if)#ipx network B201
Router(config-if)# no shutdown
Router(config-if)#exit
```

7. Now view your IPX routing table on Router 2. Notice you have a route for network AAA.

```
Router#show ipx route
```

8. Select Router 3 from the button menu.

9. Enable IPX routing on Router 3.

```
Router>enable
Router#confg t
Router(config)#ipx routing
```

10. Set Router 3's Ethernet 0 interface to match Router 1's Ethernet 1 IPX network of AAA

```
Router#conf terminal
Router(config)#interface Ethernet 0
Router(config-if)#ipx network AAA
Router(config-if)# no shutdown
Router(config-if)#exit
```

11. Now when you view your IPX routing table you should see a route to IPX network B201

```
Router#show ipx route
```

12. If we want to verify the link is up and running we should try to ping it. From Router 3 we want to ping Router 2's Ethernet 0 interface. Since it does not have an IP address we need to find out its IPX address. To do this, connect to Router 2 and view the IPX interfaces. We use this command to get the IPX address of the Ethernet 0 interface.

```
Router#show ipx interface
```

Look at Ethernet 0's IPX address and write it down so we can ping it from Router 3.

13. Connect back to Router 3 and ping using ping ipx followed by Router 1's Ethernet 0 IPX address obtained in the last step

```
Router#ping ipx "Router 2's Ethernet 0 IPX address"
```

14. If you were able to Ping, congratulations you have IPX routing working.

15. Now use the other IPX show commands.

```
Router#show ipx interface
Router#show ipx traffic
Router#show ipx interface brief
```

IPX Summary

Background

Internetwork Packet eXchange (IPX) was developed by Novell in the mid 1980's and is based on the XNS protocol developed by Xerox. The IPX protocol was the dominate LAN protocol in the late 80's and early 90's.

IPX Addressing

IPX has an 80-bit address. The first 32-bits represent the network address and the last 48-bits represent the node address. The network address is selected by the administrator. The node address is borrowed from the interfaces MAC address. For a serial interface, the node address is borrowed from the first LAN interface.
An IPX address is hexadecimal.
The format is: network.node.node.node

Examples:

5c.0000.0ca1.4567 (network portion is 5c)
1ace99.0000.0c23.1231 (network portion is 1ace99)
4001231.0002.0D12.1241 (network portion is 4001231)

IPX Routing

The three primary IPX routing protocols are IPX RIP, IPX EIGRP, and NLSP (Novell Link-Services Protocol). IPX RIP is a distance vector routing protocol that advertises the entire routing table every 60 seconds. The metric has two parts: ticks then hops. Ticks are about 1/18 of a second and measure the delay. A serial link is 6 ticks, and an Ethernet link is 1 tick. The path with the lowest number of ticks is used. In the event of a tie, hop count is the tie-breaker. IPX RIP is automatically enabled with the ipx routing command and simply places a network address on an interface.

IPX Load Balancing

If a Cisco router has two equivalent paths to a destination, by default it will only select ONE of these paths. This differs from IP which load balances by default. To enable IPX load balancing so the router will balance the number of packets sent across equal cost paths use the **ipx maximum-paths** command to specify how many paths it will load balance across.

IPX Service Advertising Protocol (SAP)

Netware compatible devices use SAPs to broadcast the name and type of resources they have to share every 60 seconds. A single Netware server can have multiple SAPs, for example one for file-sharing (type 4) and one for printer-sharing (type 7). Cisco routers can be configured to generate SAPs as well.

IPX Encapsulation or Frame types

IPX is a layer-3 protocol that is wrapped in a layer 2 frame before being released onto the network. The type of layer 2 frame or encapsulation can vary. The default encapsulation type on serial links is HDLC. It is possible to have more than one encapsulation type on an interface. This chart shows shows the Ethernet Encapsulation types.

Encapsulation Type Cisco Name	**Encapsulation Type Novell Name**	**When is it used?**
novell-ether***	802.3	Old Novell. Version 3.11 or earlier
sap	802.2	New Novell. Version 3.12 or later
arpa	ethernet_II	rarely
snap	snap	IPX for Macintosh
*** default on Ethernet		

Sub-interfaces

There may be times when you have more than one encapsulation type running on a segment.   This can happen when you have an old Netware server and a newer one. To enable the router to talk to both of these encapsulation types, you must create a sub-interface for each. A sub-interface is a logical interface. You can have multiple sub-interfaces for each physical interface. To create or configure a sub-interface, follow the physical interface name with a period and the sub-interface number like interface ethernet 0.2.

IPX Configuration

Basic IPX is very easy to configure. Remember, you do not need to configure the node address because the router borrows the MAC address for that.

Basic IPX Config

1. Enable IPX Routing on the router

```
Router(config)#ipx routing
```

2. Enable IPX on the Interface by assigning the network number and, optionally, then encapsulation type (on same line or different line)

```
Router(config)#interface ethernet 0
Router(config-if)#ipx network 6F encapsulation arpa
or
Router(config)#interface ethernet 0
Router(config-if)#ipx network 6F
Router(config-if)#ipx encapsulation arpa
```

Viewing your IPX information:

1. View the interfaces information

```
Router#show ipx interface
Router#show ipx interface brief
```

2. View the IPX routing information

```
Router#show ipx route
```

Lab 23: Introduction to the Switch

Objective: To be able to view some basic areas of a 1900 switch
Lab Equipment: We will be using Switch 1. To select Switch 1 click on the button labeled "Switch 1" at the top of the screen.

1. You should start out with a basic prompt.

```
>
```

2. Enter the command to show the IOS version of the switch.

```
>show version
```

What version of the IOS are you running? ________________________

What is the model number of the switch? ________________________

What is the Base Ethernet Address of the switch? _________________

3. Show the interfaces of the switch.

```
>show interfaces
```

How many of those interfaces are 10 Mbps? ______________________

How many ports are 100 Mbps Fast Ethernet? ___________________

4. Enter the command to view your MAC address table.

```
>show mac-address-table
```

How many dynamic entries have you learned? ____

```
>
```

5. Show your running configuration.

```
>show running-config
```

The Basic Switch: Summary

A switch works at Layer 2 of the OSI model (the Data Link Layer) and functions to concentrate the point of attachment for workstations, servers, routers, hubs and switches. A switch provides a dedicated point-to-point connection between two networking devices, so there are no collisions.

Switch Components

A switch includes all the hardware components a PC has including a CPU, RAM, and an operating system (IOS). A switch can be managed the same way a router can, you can console into its console port, telnet to its IP address and even change IOS's through the use of TFTP.

Switches use some of the same commands that Routers use. To check info about the interfaces you can use the show interfaces command. To show the ip info for the interfaces use the Show ip interfaces command. If you wanted to find information relating to the model number or IOS version you would still use the show version command. If you wanted to view the running configuration file the command is still show running-config**.**

There are some commands that we are not familiar with yet. One of those is show mac-address-table this command will show the mac table for the switch. The mac table is the table that matches all the ports on the switch with the mac addresses it has learned.

Lab 24: Introduction to Basic Switch Commands

Objective: Basic configuration of the 1912 switch
Lab Equipment: We will be using Switch 1. To select Switch 1 click on the button labeled "Switch 1" at the top of the screen.

1. You should start out with a basic prompt.

```
>
```

2. We wish to view the list of all available commands to us at this prompt. Enter the command that is used to view all the available commands.

```
>?
```

3. Now enter Privilege mode. This is the mode that lets you have total control of the switch.

```
>enable
#
```

4. View the available commands in Privilege mode.

```
#?
```

5. We would like to configure the switch. What command do we use to get into configuration mode?

```
#config terminal
(config)#
```

6. The Host Name is used for local identification. When you log into the switch you see the *Host Name* in front of the prompt (either the > or the #). This can be used to identify the location or function of the switch. Set your Switch's hostname to **Boson**. What command do you use to configure the hostname?

```
(config)#hostname Boson
Boson(config)#
```

7. The enable password controls access to privilege mode. This is a VERY important password because in privilege mode you can make configuration changes. There is a difference in syntax when setting the router and the switches password. On the 1900 series switch there are levels that you need to set when you declare a password. These are used so you can have different sets of people who can enter different commands on the switch. Set your enable password to **Krang**. What command will accomplish this?

```
Boson(config)#enable password level 15 Krang
```

8. Let's test this password. Exit out of the switch and try to enter privilege mode. Notice what password got you into privilege mode. Now type: conf term and proceed with the lab instructions in the next step.

```
Boson(config)#exit
Boson#exit
Boson>enable
Password:
Boson#
```

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 show someone this file so that they can help you troubleshoot a problem you may inadvertently compromise the security of your systems by revealing the passwords. What command will create an encrypted password? Set the enable secret password to **cisco**. Don't forget to use the level commands. What command did you use?

```
Boson(config)#enable secret level 15 cisco
```

10. You can now test this password by logging out of the switch and then typing enable. The enable secret is an additional password over and above the enable password, in fact, it overrides the enable password. If you have set both passwords, the enable SECRET is the password you use to enter into privilege mode. The enable PASSWORD is still present but is now deactivated.

```
Boson(config)#exit
Boson#exit
Boson>enable
Password:
Boson#
```

Basic Switch Commands Review

This is equivalent to the lab above with the exception that it does not give you the answers as you complete each step.

1. Connect to Switch 1 and view all the available commands. What command did you use to do this? ______________

2. Now enter Privilege mode. What command did you use? _______________

3. View all available commands for Privilege mode. What command did you use?
____

4. What command will get you into configuration mode? _______________

5. Now set the switch's hostname to Boson. What command did you use to do this?

____

6. Now we will set the enable password to **Krang**. What command will do this?

____

7. Test the password by logging out of the switch and then trying to enter enable mode.

8. Set the secret password to **cisco**. What command did you use to do this?

____

9. Logout of the switch again and enter privilege mode. What password was required?

____

Basic Switch Commands Summary

This lab will introduce the Cisco Internetwork Operating System (IOS) command line interface (CLI) for the 1900 series switch. You will need to logon to a switch and become familiar with the different levels of access on the switch. You will also become familiar with the commands available to you in each mode (user or privileged) and the switch help facility, history, and editing features.

User vs. Privileged Mode

User mode is indicated with the > next to the switch name. You can look at settings but can not make changes from user mode. In Privilege mode, indicated by the #, you can do anything. To get into privilege mode the keyword is enable**.**

```
Switch>
Switch>enable
Password:
Switch#
```

HELP

To view all commands available from this mode type: ?   This will give you the list of all available commands for the switch in your current mode. You can also use the question mark after you have started typing a command. For example if you want to use a show command but you do not remember which one it is, use the ? as this will output all commands that you can use with the show command.

```
r1#show ?
access-expression List access expression
```

```
access-lists List access lists
backup Backup status
cdp CDP information
clock Display the system clock
cls DLC user information
compress Show compression statistics
configuration Contents of Non-Volatile memory
--More--
```

Configuration Mode

From privilege mode you can enter configuration mode by typing **config** term command you can exit configuration mode type type end or <CTL>+z

```
Switch#config t
Switch(config)#end.
```

Lab 25: Introduction to Frame Relay

Objective: To understand how to establish a Frame Relay connection.
Lab Equipment: We will be using Router 1 & Router 2. To select Router 1 click on the button Router1 at the top of your screen.

1. On Router 1, enter global configuration mode on Router 1

```
Router>enable
Router#conf t
Router(config)#
```

2. On Router 1 change the hostname to R1

```
Router(config)#hostname R1
R1(config)#
```

3. On R1 assign an IP address of 10.1.1.1 255.255.255.0 to the Serial 0 interface.

```
R1(config)#interface Serial 0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
```

4. On R1 make sure you enable the Serial 0 interface.

```
R1(config-if)#no shut
```

5. Now connect to Router 2 and change the hostname to R2

```
Router>en
Router#config t
Router(config)#hostname R2
R2(config)#
```

6. On R2 assign an IP address of 10.1.1.2 255.255.255.0 to the Serial 0 interface.

```
R2(config)#interface Serial 0
R2(config-if)#ip address 10.1.1.2 255.255.255.0
```

7. On R2 make sure you enable the Serial 0 interface.

```
R2(config-if)#no shut
```

8. On R1 set the encapsulation for the Serial interface to Frame-Relay. Notice how both interfaces are still down. We need to setup the frame relay information on both interfaces so that the interfaces come up.

```
R1(config-if)#encapsulation frame-relay
```

9. Next set the frame relay interface DLCI for the connection from Router 1 to Router 2. Since we are using the default frame network the DLCI will be **102**.

```
R1(config-if)#frame-relay interface-dlci 102
```

10. On R2 set the encapsulation for the Serial interface to Frame-Relay. Notice how both interfaces are still down. We need to setup the Frame-Relay interface DLCI for this interface so the connection can go up.

```
R2(config-if)#encapsulation frame-relay
```

11. Now set the Frame-Relay interface DLCI for the connection from Router 2 to Router 1. Since we are using the default frame network the DLCI will be **201**.

```
R2(config-if)#frame-relay interface-dlci 201
```

12. You should have seen the output from the router saying that the DLCI changed to state ACTIVE. This means we have established a connection from our Router 1 through the Frame Relay switch to Router 2. Verify that you have your configuration setup correctly by first trying to ping the serial 0 ip address on Router 1. Next we will be using the frame relay show commands for the proof that our connection is active. The first command we will look at is show frame-relay lmi. This command shows the LMI traffic that has been exchanged from our Router and the Frame-Relay Switch.

```
R2(config-if)#exit
R2(config)#exit
R2#ping 10.1.1.1
R2#show frame-relay lmi
```

13. The next command is show frame-relay traffic. This command shows the global frame-relay statistics since the last reload of the router.

```
R2#show frame-relay traffic
```

14. The command show frame-relay map will show the mappings of the layer 2 DLCI to the Layer 3 IP address. Later on we will see that you can add your own mappings into this table using the frame-relay map command.

```
R2#Show frame-relay map
```

15. The command show frame-relay PVC will display 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
```

Frame Relay Summary

Frame Relay

Frame Relay is a network access protocol similar in principle to X.25 (a protocol is a set of procedures or rules that govern the transfer of information between devices). The main difference between Frame Relay and X.25 is data integrity (error detection) and network error flow control (error correction).

X.25 does all of its data checking and correcting at the network level. That means the network devices correct the corrupt data or ask for the data to be retransmitted. The cost of such checking and retransmission is network delay.

Frame Relay performs error detection only, not error correction. It thereby leaves the task of error correction to the protocols used by intelligent devices at each end of the network. These intelligent devices will provide end-to-end data integrity. Since Frame Relay relies on the device at the end to perform retransmission and error recovery, there is significantly less processing required for the network and less overall delay.

description *descriptive-string*

A description can be added to an interface to help keep track of PVCs (e.g. Frame Relay to Boston)

encapsulation frame-relay [cisco | ietf**]**

This command specifies frame relay encapsulation.

frame-relay interface-dlci *dlci* **[**broadcast**]**

This command assigns a Data Link Connection Identifier (DLCI) number to the corresponding frame-relay sub interface. A DLCI is assigned by the local frame relay provider for every Permanent Virtual Circuit (PVC) connected to the router. DLCI numbers are NOT exchanged between routers. DLCI numbering at one frame relay site is mutually exclusive from DLCI numbering at another site. This concept is illustrated in the examples. DLCI numbers for Atlanta (16 and 17) need not match DLCI numbers for Boston (16) and Chicago (16). By the same token, it is OK for Boston and Chicago to both use DLCI 16.

The broadcast keyword is optional and should only be included if broadcast packets (e.g. IP RIP or IPX RIP/SAP updates) need to be forwarded out of the sub interface. In static routing examples, routing updates are not required and the keyword is omitted.

frame-relay lmi-type {ansi | cisco | q933a**}**

This command configures the router with which frame-relay Local Management Interface (LMI) type to expect from the Frame Relay provider. LMI is a frame relay control protocol sent to the router from the frame relay switch at the service provider and is not exchanged between routers. The LMI type at one location does NOT have to match the LMI type at other locations. To illustrate this point, the examples have Boston (Cisco LMI) using a different LMI type than Atlanta (ANSI Annex D LMI).

Supported LMI Types	
cisco	default
ansi	ANSI Annex D
q933a	CCITT Q933a

frame-relay map ip *ip-address dlci* **[**broadcast**]**

This command is used in multipoint frame-relay examples and defines a static mapping between a protocol address and a frame-relay Data Link Connection Identifier (DLCI). A DLCI is assigned by the local frame relay provider for every Permanent Virtual Circuit (PVC) connected to the router. DLCI numbers are NOT exchanged between routers. DLCI numbering at one frame relay site is mutually exclusive from DLCI numbering at another site. This concept is illustrated in the examples. DLCI numbers for Atlanta (16 and 17) need not match DLCI numbers for Boston (16) and Chicago (16). By the same token, it is OK for Boston and Chicago to both use DLCI 16.

The broadcast keyword is optional and should only be included if broadcast packets (e.g. IP RIP or IPX RIP/SAP updates) need to be forwarded out of the sub interface. In static routing examples, routing updates are not required and the keyword is omitted.

In multipoint frame-relay examples, Atlanta uses DLCI 16 to reach Boston (IP address 172.16.1.2). Therefore, Atlanta defines a static frame relay map with the command "frame-relay map IP 172.16.1.2 16". Also, Boston contains static frame-relay maps to use DLCI 16 for both Atlanta and Chicago because traffic destined to Chicago must first be sent over the PVC to Atlanta. Atlanta will then redirect the packet out its PVC to Chicago.

interface Serial0.*subinterface#* **[**point-to-point | multipoint**]**

This command creates a logical frame-relay sub interface and defines it as a point-to-point or multipoint connection. A sub interface is treated as if it where a separate interface dedicated for a PVC to a remote site. "Serial0" indicates that the sub interface belongs to the physical serial0 interface and "16" is the unique sub interface ID number.

The sub interface ID number can be any unique value between zero and 4,294,967,295 and does not have to be in any particular order (i.e. it is not necessary to begin with 1 and sequentially progress with 2,3,…etc.). In fact, to reduce confusion, it is good practice to identify a sub interface with the same number as the DLCI used on that sub interface.

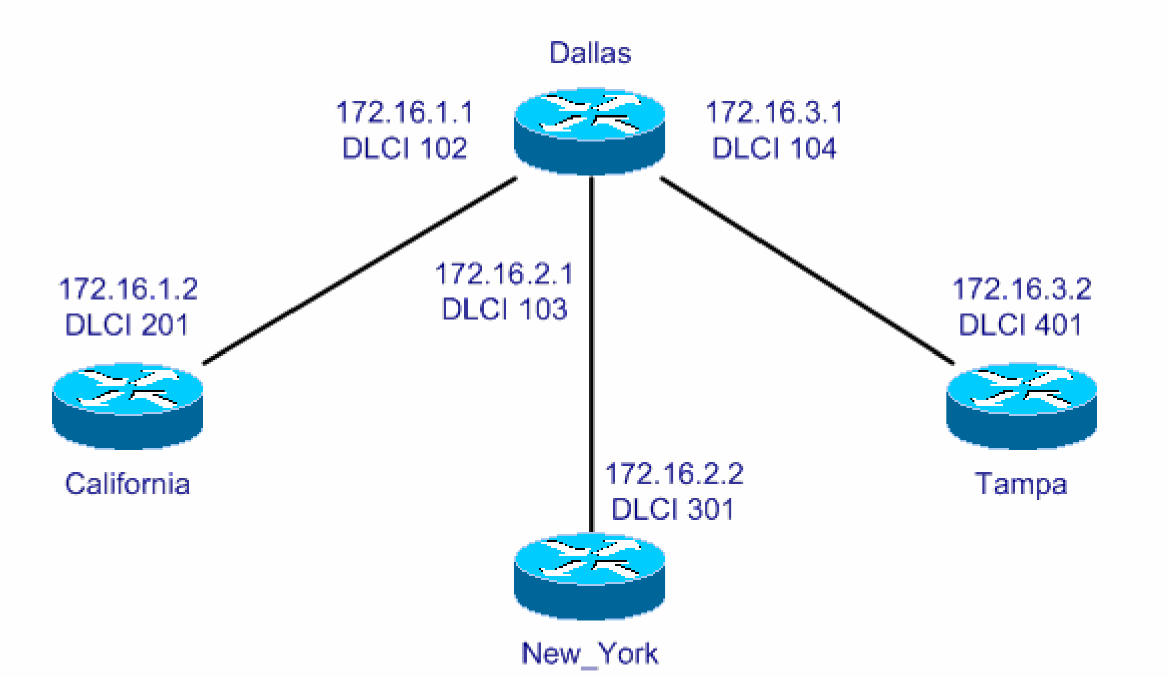
ip address *ip-address subnet-mask*

This command configures an interface with an IP address and subnet mask. In IP routing examples, 10.1.1.1 is the IP address of the Ethernet interface in Atlanta and 255.0.0.0 is the corresponding subnet mask. For examples in which IP is bridged, all interfaces on the router are configured with the same IP address because the router is reduced to a simple node on an IP network with only one IP address.

Lab 26: Frame Relay Hub and Spoke Topology

Objective: This lab will allow you to configure a Hub and Spoke topology. A hub and spoke topology can be used when you have a corporate office and smaller sales offices in other cities. All of the sales offices connect to the corporate office to send all of its data. This is also how they can communicate between sales offices.

Here is the example of a full mesh frame relay topology.



1. Enter Global Configuration Mode on Router 1.

```
Router>enable
Router#conf t
Router(config)#
```

2. Enter a hostname of **Dallas**.

```
Router(config)#hostname Dallas
Dallas(config)#
```

3. Configure the Serial 0 interface.

```
Dallas(config)#interface Serial 0
Dallas(config-if)#
```

4. Set the encapsulation type to frame relay. Don't forget to enable the interface.

```
Dallas(config-if)#encapsulation frame-relay
Dallas(config-if)#no shutdown
```

5. Now we will create a subinterface for each of the connections to the Sales offices.

```
Dallas(config-if)#exit
Dallas(config)#
Dallas(config)#interface serial 0.100 point-to-point
Dallas(config-subif)#
```

6. Next we will need to assign the DLCI number for our connection between Router 1 and Router2 (Dallas and California) and the IP address of 172.16.1.1 255.255.255.0.

```
Dallas(config-subif)#frame-relay interface-dlci 102
Dallas(config-subif)#ip address 172.16.1.1 255.255.255.0
```

7. Next create a sub interface for the connection to the Sales office in New York.

```
Dallas(config-subif)#exit
Dallas(config)#interface serial 0.200 point-to-point
Dallas(config-subif)#
```

8. Now add the DLCI for this connection and the IP address for our connection to New York.

```
Dallas(config-subif)#frame-relay interface-dlci 103
Dallas(config-subif)#ip address 172.16.2.1 255.255.255.0
```

9. Next create the sub interface for the connection to the Sales office in Tampa.

```
Dallas(config-subif)#exit
Dallas(config)#interface serial 0.300 point-to-point
Dallas(config-subif)#
```

10. Add the DLCI for this connection and the IP address for our connection to Tampa.

```
Dallas(config-subif)#frame-relay interface-dlci 104
Dallas(config-subif)#ip address 172.16.3.1 255.255.255.0
```

11. Next connect to Router 2 (California) enter configuration mode and set the hostname to California.

```
Router>enable
Router#config t
Router(config)#hostname California
California(config)#
```

12. Get into the Serial interface and set the encapsulation to frame relay and enable the interface.

```
California(config)#interface serial 0
California(config-if)#encapsulation frame-relay
California(config-if)#no shutdown
```

13. Now since we only have one connection to worry about, we don't need to use sub interfaces, just add the DLCI value we are going to use here.

```
California(config-if)#frame-relay interface-dlci 201
```

14. Next set the IP address for this interface.

```
California(config-if)#ip address 172.16.1.2 255.255.255.0
```

15. Next connect to Router 3 (New York) enter configuration mode and set the hostname to **New_York**.

```
Router>enable
Router#config t
Router(config)#hostname New_York
New_York(config)#
```

16. Get into the Serial interface and set the encapsulation to frame relay.

```
New_York(config)#interface serial 0
New_York(config-if)#encapsulation frame-relay
```

17. Now since we only have one connection to worry about we don't need to use sub interfaces, just add the DLCI value we are going to use here.

```
New_York(config-if)#frame-relay interface-dlci 301
```

18. Next 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
```

19. Next connect to Router 4 (Tampa) enter configuration mode and set the hostname to Tampa.

```
Router>enable
Router#config t
Router(config)#hostname Tampa
Tampa(config)#
```

20. Get into the Serial interface and set the encapsulation to frame relay.

```
Tampa(config)#interface serial 0
Tampa(config-if)#encapsulation frame-relay
```

21. Now since we only have one connection to worry about we don't need to use subinterfaces, just add the DLCI value we are going to use here.

```
Tampa(config-if)#frame-relay interface-dlci 401
```

22. Next set 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
```

23. Now we should have all interfaces up and up. To test, connect to Router1 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
```

If you are successful congratulations!! Now for a more advanced lab get RIP running on all routers so that you can see routes to Tampa and New York from California!!!

Lab 27: Frame Relay Full Mesh Topology

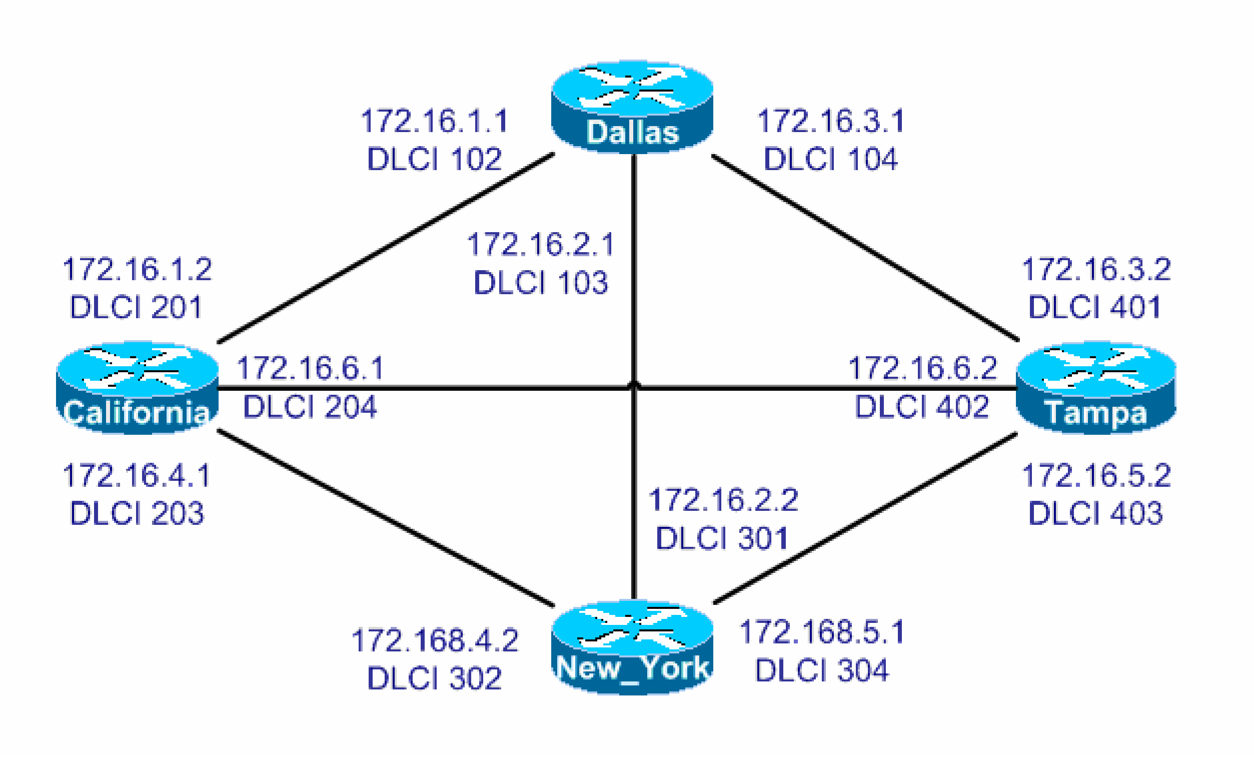
Objective: This lab will allow you to configure a Full Mesh topology.

Full Mesh Topology

The difference between the frame relay hub and spoke topology and the full mesh topology is that now every sales office will have a direct connection to every other sales office along with the corporate office. This is a very redundant topology so if one of the connections goes down data can still be transferred to every site by using a different path.

Let's go back to our example of having a corporate office in one city and smaller sales offices in other cities. Each sales office will have a connection to each of the other sales offices and the corporate office will have a link to each of the sales offices as well. The theory is that we have our corporate headquarters in Dallas and we want our sales offices in California, New York and Tampa to be able to access all of our company resources. We will have a point-to-point frame Relay connection to each sales office and from each sales office to every other sales office. This will start off with the same configuration as we used for the hub and spoke for the first router but all of the sales offices will have more configurations.

Here is an example of a full mesh frame relay topology.



1. Enter global configuration mode on Router 1.

```
Router>enable
Router#conf t
Router(config)#
```

2. Enter a hostname of Dallas.

```
Router(config)#hostname Dallas
Dallas(config)#
```

3. Configure the serial 0 interface.

```
Dallas(config)#interface Serial 0
Dallas(config-if)#
```

4. Set the encapsulation type to frame relay. Don't forget to enable the interface.

```
Dallas(config-if)#encapsulation frame-relay
Dallas(config-if)#no shutdown
```

5. Now we will create a sub interface for each of the connections to the Sales offices.

```
Dallas(config-if)#exit
Dallas(config)#
Dallas(config)#interface serial 0.100 point-to-point
Dallas(config-subif)#
```

6. Next we will need to assign the DLCI number for our connection between Router 1 and Router2 (Dallas and California) and the IP address of 172.16.1.1 255.255.255.0.

```
Dallas(config-subif)#frame-relay interface-dlci 102
Dallas(config-subif)#ip address 172.16.1.1 255.255.255.0
```

7. Next create a sub interface for the connection to the Sales office in New York.

```
Dallas(cofig-subif)#exit
Dallas(config)#interface serial 0.200 point-to-point
Dallas(config-subif)#
```

8. Now add the DLCI for this connection and the IP address for our connection to New York.

```
Dallas(config-subif)#frame-relay interface-dlci 103
Dallas(config-subif)#ip address 172.16.2.1 255.255.255.0
```

9. Next create the subinterface for the connection to the Sales office in Tampa.

```
Dallas(config-subif)#exit
Dallas(config)#interface serial 0.300 point-to-point
Dallas(config-subif)#
```

10. Add the DLCI for this connection and the IP address for our connection to Tampa.

```
Dallas(config-subif)#frame-relay interface-dlci 104
Dallas(config-subif)#ip address 172.16.3.1 255.255.255.0
```

11. Next connect to Router 2 (California) enter configuration mode and set the hostname to California.

```
Router>enable
Router#config t
Router(config)#hostname California
California(config)#
```

12. Get into the Serial interface and set the encapsulation to frame relay.

```
California(config)#interface serial 0
California(config-if)#encapsulation frame-relay
California(config-if)#no shutdown
```

13. Lets create our first sub interface for our connection to the corporate office.

```
California(config-if)#interface serial 0.100 point-to-point
California(config-subif)#
```

14. Add the correct DLCI value for this connection.

```
California(config-subif)#frame-relay interface-dlci 201
```

15. Next set the IP address for this interface.

```
California(config-subif)#ip address 172.16.1.2 255.255.255.0
```

16. Now create the subinterface for the connection to New York (Router 3).

```
California(config-subif)#exit
California(config)#interface serial 0.200 point-to-point
```

17. Add the correct DLCI value for this connection.

```
California(config-subif)#frame-relay interface-dlci 203
```

18. Next set the IP address for this interface and enable the interface.

```
California(config-subif)#ip address 172.16.4.1 255.255.255.0
```

19. Now create the subinterface for the connection to Tampa (Router 4).

```
California(config-subif)#exit
California(config)#interface serial 0.300 point-to-point
```

20. Add the correct DLCI value for this connection.

```
California(config-subif)#frame-relay interface-dlci 204
```

21. Next set the IP address for this interface.

```
California(config-subif)#ip address 172.16.6.1 255.255.255.0
```

22. Next connect to Router 3 (New York) enter configuration mode and set the hostname to New_York.

```
Router>enable
Router#config t
Router(config)#hostname New_York
New_York(config)#
```

23. Get into the Serial interface and 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
```

24. Lets create our first sub interface for our connection to the corporate office.

```
New_York(config-if)#exit
New_York(config)#interface serial 0.100 point-to-point
```

25. Add the correct DLCI value for this connection.

```
New_York(config-subif)#frame-relay interface-dlci 301
```

26. Next set the IP address for this interface and enable the interface.

```
New_York(config-subif)#ip address 172.16.2.2 255.255.255.0
```

27. Now create the subinterface for the connection to California (Router 2).

```
New_York(config-subif)#exit
New_York(config)#interface serial 0.200 point-to-point
```

28. Add the correct DLCI value for this connection.

```
New_York(config-subif)#frame-relay interface-dlci 302
```

29. Next set the IP address for this interface and enable the interface.

```
New_York(config-subif)#ip address 172.16.4.2 255.255.255.0
```

30. Now create the sub interface for the connection to Tampa (Router 4).

```
New_York(config-subif)#exit
New_York(config)#interface serial 0.300 point-to-point
```

31. Add the correct DLCI value for this connection.

```
New_York(config-subif)#frame-relay interface-dlci 304
```

32. Next set the IP address for this interface and enable the interface.

```
New_York(config-subif)#ip address 172.16.5.1 255.255.255.0
```

33. Next connect to Router 4 (Tampa) enter configuration mode and set the hostname to Tampa.

```
Router>enable
Router#config t
Router(config)#hostname Tampa
Tampa(config)#
```

34. Get into the Serial interface and set the encapsulation to frame relay.

```
Tampa(config)#interface serial 0
Tampa(config-if)#encapsulation frame-relay
Tampa(config-if)#no shutdown
```

35. Lets create our first sub interface for our connection to the corporate office (Dallas).

```
Tampa(config-subif)#exit
Tampa(config-if)#interface serial 0.100 point-to-point
```

36. Add the correct DLCI value for this connection.

```
Tampa(config-subif)#frame-relay interface-dlci 401
```

37. Next set the IP address for this interface and enable the interface.

```
Tampa(config-subif)#ip address 172.16.3.2 255.255.255.0
```

38. Now create the sub interface for the connection to California (Router 2).

```
Tampa(config-subif)#exit
Tampa(config)#interface seial 0.200 point-to-point
```

39. Add the correct DLCI value for this connection.

```
Tampa(config-if)#frame-relay interface-dlci 402
```

40. Next set the IP address for this interface.

```
Tampa(config-subif)#ip address 172.16.6.2 255.255.255.0
```

41. Now create the sub interface for the connection to New York (Router 3).

```
Tampa(config-subif)#exit
Tampa(config)#interface serial 0.300 point-to-point
```

42. Add the correct DLCI value for this connection.

```
Tampa(config-subif)#frame-relay interface-dlci 403
```

43. Next set the IP address for this interface and enable the interface.

```
Tampa(config-subif)#ip address 172.16.5.2 255.255.255.0
```

44. Now we should have all interfaces up and up. To test connect to Router1 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
```

45. Connect to Router 2 (California) and try to ping the other three offices.

```
California(config-subif)#exit
California(config)#exit
California#ping 172.16.1.1
California#ping 172.16.4.2
California#ping 172.16.6.2
```

Lab 28: Standard Access Lists

Objective: To understand and gain experience in configuring standard access lists.
Lab Equipment: We will be using Routers 1, 2, and 4 for this lab.

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

	Router1	Router2	Router4
Interface Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Interface Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240
RIP	On interfaces Ethernet 0 and Serial 0	On Ethernet 0	On Serial 0

1. Select Router1 from the menu bar 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** interface to **24.17.2.17 255.255.255.240**. Don't forget to enable both interfaces.

```
Router>
Router#
Router#config t
Router(config)#hostname Router1
Router1(config)#
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. Select Router 2 from the menu bar and set the IP address on the **Ethernet 0** interface to **24.17.2.2 255.255.255.240**. Don't forget to enable the interface.

```
Router>
Router>enable
Router#
Router#config t
Router(config)#hostname Router2
Router2(config)#
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. Ping the Router 1's Ethernet 0 interface to ensure you have a connection (24.17.2.1).

```
Router2#ping 24.17.2.1
```

4. Select Router 4 from the menu bar and set the IP address on the **Serial 0** interface to **24.17.2.18 255.255.255.240**, then ping Router 1's Serial 0 interface (24.17.2.17).

```
Router>
Router>enable
Router#
Router#config t
Router(config)#hostname Router4
Router4(config)#
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 we have established IP addresses on all the interfaces we need to have a routing protocol enabled to facilitate communication between Router 2 and Router 4. We will use RIP as our routing protocol. Enable RIP on Router 1 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 Router 2 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. Finally on Router 4 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. Now that we have finished all of our background configuration verify that you can ping Router 2’s Ethernet 0 (24.17.2.2) interface from Router 4.

```
Router4#ping 24.17.2.2
```

9. Our Standard Access List will block Router 4 from being able to ping Router 2. We will configure this access list on Router 2. Connect to Router 2 and enter configuration mode.

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

10. Create an access-list 1 that blocks only the single IP address 24.17.2.18 followed by the command access-list 1 permit any. We have listed 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
----- THEN ----
Router2(config)#access-list 1 permit any
```

11. After you have created the access-list you need to apply it to the Ethernet 0 interface. What direction would you apply the access-list in? "In" means packets coming in from the network and going to the router and "out" means packets going from the router out the interface to the network. For our example we would use the command IP access-group 1 in.

```
Router2(config)#interface ethernet0
Router2(config-if)#ip access-group 1 in
Router2(config-if)#exit
```

12. This completes the Standard Access List lab. Please continue on to the Verify Standard Access-list lab (Lab 29) without accessing the Lab Navigator.

Access-List Overview

As the name implies, **access-lists** are sequential listings of guidelines, which are used to provide or prevent the flow of packets within a network based on information provided within the list. Standard IP access-lists are very straightforward in the fact that the only criteria used to determine if packets should be ‘permitted’ or ‘denied’ are based solely on the source address of any given packet.

Access-lists may be used for a variety of reasons, including; controlling the propagation and reception of routing updates traffic shaping, definition of traffic that will allow dial backup connectivity, and security. The primary implementation, and the main topic of this lesson, will be to implement the access-list as a security mechanism.

Why implement restricted access?

You may choose to implement security policies for a variety of reasons, which includes, but is certainly not limited to, the prevention of outside attacks on company devices, isolation of interdepartmental traffic, or load distribution. Without the use of access-lists all packets within a network are allowed without restriction to all parts of the network.

When using access-lists as a “firewall”, routers can limit or restrict access to your internal network from an outside network, for example the Internet.   This type of access-list would typically be placed at the point of connection between the two networks. When using access-lists for interdepartmental isolation, the access-list would typically be placed at strategic locations within the internal network.

The Basics of Standard IP Access-Lists

The basic format of the Standard IP Access-List is:

```
access-list [#] [permit | deny] [source-address | keyword any] [source mask]
```

As mentioned earlier, an access-list is a sequential listing of guidelines that are used to provide or prevent the flow of packets. In other words an access-list may contain multiple lines, each following the format as listed above. The access-list may contain multiple lines, specifying multiple source addresses to be evaluated. Each line entry of the access-list must maintain the same access-list number identifier so the router will know that the entities listed will be grouped into the same access-list. Always remember that access-lists are processed "top down", which means that the first line of the access-list will be check, then the second, etc. The router will immediately break out of processing the access list with the first “match”. Therefore the most general statements should be placed at the beginning of the list to avoid extra processing. This topic will be discussed in greater detail at a later point.

Various access-lists can be defined by different protocols within a router. The router will know the type of access-list based on the access-list number that is assigned. The numbering range for Standard IP Access-Lists is from 1 to 99. All Standard IP Access-Lists must be numbered within this range.

After a number in the appropriate range has been selected for your access-list, the list must know if the packets to be evaluated will be ‘permitted’ (allowed to pass) or ‘denied’ (dropped and not allowed to pass). This is accomplished by placing either a **permit** or **deny** keyword within the line of the access-list command. The usage of the keyword instructs the router to allow the packet to pass or not based on the next specified parameter, the source address contained within the evaluated packet.

As briefly discussed earlier, the only criteria used by Standard IP Access-Lists to determine if a packet should be ‘permitted’ or ‘denied’ is based solely on the source address of any given packet. This brings us to the point where we specify exactly which host (or hosts) will be permitted or denied by our access-list. This parameter is quite simply, the source IP address of the host that you wish the access-list to take action upon. You may optionally replace the address with the keyword **any** which will cause the router to act upon “any” IP address.

As found with most IP addressing schemes, the standard IP access-list allows for a source-mask to be applied to the source IP address. Although similar to the subnet mask that is applied to IP addresses, the source-mask is somewhat different. When using a source-mask with IP access-lists, a bit set to 0 means “match exactly” and a bit set to 1 means “don’t care”. For example, if you would like to include all hosts in the class C network 192.1.1.0, the source address, source mask combination would be: 192.1.1.0 0.0.0.255. This statement says: In the first, second, and third octet of this address (192.1.1), all bits must “match exactly” (0.0.0, or all 0’s in the source-mask for the first, second, and third octet), but we “don’t care” what bits are sent in the fourth octet (255, or all 1’s in the source-mask for the fourth octet). By using this source address / source mask combination a single line in our access list includes all hosts in the 192.1.1.0 network. The keyword **any**, was briefly mentioned earlier. This keyword is the same as using a source address / source mask combination of 0.0.0.0 255.255.255.255. The 255.255.255.255 source mask indicates we “don’t care” what bits are set in any of the four octets. The use of the source mask parameter is optional. If the omitted from the configuration line, the router by default will use a source mask of 0.0.0.0, or “match exactly” the address entered.

We now have the basic building blocks to begin building our first standard IP access-list. There is one more note that is critical to the successful completion of building an access-list. After an access-list has been created, the Cisco router will assume that any source ip addresses that are not explicitly mentioned in the list will be ***DENIED*.** In other words, at the end of the access-list, the router will implicitly deny all remaining traffic. If your access-list has been configured to permit only a single source-address of 1.1.1.1, ALL OTHER SOURCE ADDRESSES WILL BE IMPLICITLY DENIED.

Creating a simple Standard IP Access-List

Now the time has come to create our first Standard IP Access-list. We will use the format as discussed:

```
access-list [#] [permit | deny] [source-address | keyword any] [source mask]
```

Access-lists are created in global configuration mode of the router. Remember that all standard IP access-lists must be numbered in the range of 1-99, for our example we will use #1. We have decided that we want to permit traffic from address 1.1.1.1, and deny all other traffic.   The procedure will be as follows:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
eRouter(config)#access-list 1 permit 1.1.1.1
eRouter(config)#^Z
Router#
```

This configuration creates a permit statement for host address 1.1.1.1. Since the source-mask was not specified, the router uses a default of 0.0.0.0 (match exactly).   Don't forget the implicit "deny any" at the end of the access-list, this automatically denies everything we did not permit.

Applying the Access-list to an interface

Before the access-list actually does any work it has to be applied to an interface. The interface configuration command for applying the standard IP access-list to an interface is:

```
ip access-group [access-list-number] [in | out]
```

Access-lists may be applied as either outbound or inbound on the router interfaces. When you apply the access-list as an inbound list, the router will receive an inbound packet; check the source address of the packet against the access list. "Permit" the packet to be routed to the destination interface if the packet matches a "permit" statement in the access-list, or discard the packet if the packet matches a "deny" statement in the access-list.

When you apply the access-list as an outbound list, the router will receive a packet on an interface, route the packet to the appropriate outbound interface, and then check the source address of the packet against the access-list. At this point the router will either "permit" the packet to exit the interface if the packet matches a "permit" statement in the access-list. If the packet matches a "deny" statement in the access-list it will be discarded.

To apply the access-list we created above to interface Ethernet 0 as an inbound access-list:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
eRouter(config)#int Ethernet 0

eRouter(config-if)#ip access-group 1 in
eRouter(config-if)#^Z
Router#
```

To apply the access-list we created above to interface Ethernet 0 as an outbound access-list:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
eRouter(config)#int Ethernet 0

eRouter(config-if)#ip access-group 1 out
eRouter(config-if)#^Z
Router#
```

Creating a more advanced Standard IP Access-List

Now let’s create a more advanced access list. In this exercise we will create access-list #2, with the following criteria.

Permit all packets originating from network 10.1.1.0 255.255.255.128, but deny all packets originating from network 10.1.1.128 255.255.255.128. We also want to deny all packets originating from network 15.1.1.0 except for packets from a single host of 15.1.1.5. The final criterion is to permit all other traffic not previously mentioned. The procedure will be as follows:

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
eRouter(config)#access-list 2 deny 10.1.1.128 0.0.0.127
eRouter(config)#access-list 2 permit 15.1.1.5
eRouter(config)#access-list 2 deny 15.1.1.0 0.0.0.255
eRouter(config)#access-list 2 permit any
eRouter(config)#^Z
Router#
```

One of the first things you may notice about our configuration is that there is no permit statement for the network 10.1.1.0, which our criteria specified we must permit. The last line of the access-list (access-list 2 permit any) will take care of this criteria. Let’s review our criteria and verify we have completed our tasks:

- Permit all packets originating from network 10.1.1.0 255.255.255.128

  The last line of our access list accomplishes this criterion. It was not necessary to explicitly permit this network in our access-list since there were no statements in our access-list matching this network except for the final line of “permit any”.

- Deny all packets originating from network 10.1.1.128 255.255.255.128.

  The first line of our access list accomplishes this criterion. It is very important to note that it was necessary to use a source-mask of 0.0.0.127 for this network. This mask says we “don’t care” about the final seven bits of the fourth octet, which are the bits that have been assigned for host addressing on this network. The subnet mask specified for the network was 255.255.255.128 which says the first bit of the fourth octet has been assigned to the "subnet" the last seven bits have been assigned for host addressing.

- Deny all packets originating from network 15.1.1.0 except for packets from a single host of 15.1.1.5.

  This has been accomplished with lines 2 and three of our access-list. It is very important to note that the access-list did not accomplish this in the same order as the criteria specified. It is imperative to remember that access-lists are processed top down, and that upon the first match processing stops and action is taken. Our criteria specified to deny packets from network 15.1.1.0 and secondly permit packets from host 15.1.1.5. If lines two and three had been swapped, and the entire network 15.1.1.0 was denied prior to permitting host 15.1.1.5, packets with a source address of 15.1.1.5 would match the more general criteria of "deny 15.1.1.0" first, thus the host would have been denied before it could have been permitted.

- The final criterion is to permit all other traffic not previously mentioned.

  The last line of our access list accomplishes this by permitting "any" packets that were not matched in the first three lines of the list.

Bringing it all together

In general the process for creating and implementing Standard IP access-lists are:

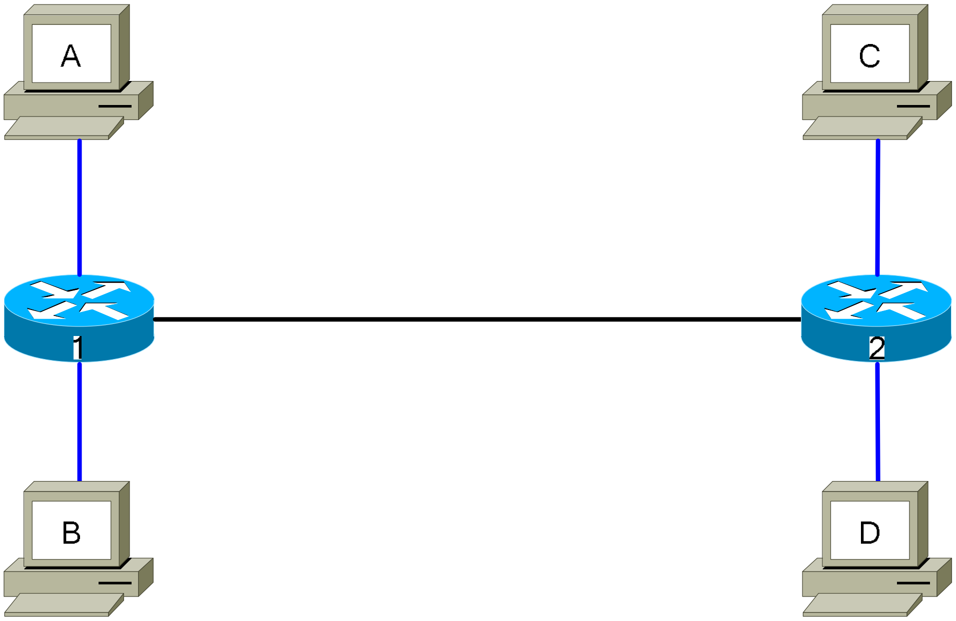
1. Define the rules for which to design the access-list
2. Create the access-list with a number in the range of 1-99
3. Apply the access-list, either inbound or outbound, to the appropriate interface

Items 1 and 2 above have been fairly well covered in this lesson. Lastly, the placement of the access-list needs to be discussed. In general Standard IP access-lists should be placed nearest the destination and not the source. However, this is not an absolute rule, certain exceptions exist. Due to the fact Standard IP access-lists only operate on the source address, detailed granularity is not always possible. Care must be taken to avoid implementing undesirable policies. If a standard access-list is placed near the source it is very possible that access to devices other than those desired will be impeded.

For example, if access-list 2, which we created in this lesson, were implemented as an inbound access-list on the Ethernet interface of a router directly connected to the 15.1.1.0 network, the only workstation that would be allowed off the local segment would be

15.1.1.5. This access-list would most likely be implemented as an outbound access-list on the remote end of the connection, where the filtering of packets is truly desired.

Viewing the figure below, let’s assume that workstation C is device 15.1.1.5, and Workstation D is device 10.1.1.133. Our desire is to implement a policy for Workstation A that only allows Workstation C access from remote Ethernet C. We also wish to implement a policy that will deny any access from remote Ethernet D. Placement is critical for this accomplishment. If access-list 2 from above is implemented as an outbound access-list on Router 2’s serial interface we will accomplish the desired task, BUT we will also deny traffic from Ethernet D to Ethernet B, which is undesired. The same scenario holds true if the access-list is implemented as an inbound access-list on Router 1’s serial interface. If we place this access-list as an outbound access-list on Router 1’s Ethernet A interface, our policy is intact, without any unwanted policy implementations.



Access List Cheat Sheet

With Access Lists you will have a variety of uses for the wild card masks, but typically you will want to do only the following:

1. Match a specific host,
2. Match an entire subnet,
3. Match an IP range, or
4. Match Everyone and anyone

Here are some simple examples to get these requirements done.

<u>Match a specific host</u>

All wildcard mask bits are zeros.
For a Standard Access-List to permit the host 192.168.0.58 you would use the following command:

```
access-list 101 permit 192.168.0.58 0.0.0.0
```

However, standard access lists assume a 0.0.0.0 mask, you could rewrite the command to be:

```
access-list 101 permit 192.168.0.58
```

For an Extended Access-List to permit the same host of 192.168.0.58 the command used would be:

```
access-list 101 permit ip 192.168.0.58 0.0.0.0 any or
access-list 101 permit ip host 192.168.0.58 any
```

<u>Match an entire subnet</u>

The key to matching an entire subnet is to use the following formula for the wildcard mask. It goes as follows:

Wildcard mask = 255.255.255.255 – subnet
So for example if my current subnet was 255.255.255.0, the mask would be 0.0.0.255. This can be seen here.

```
255.255.255.255
   255.255.255.0       -
        0.0.0.255
```

In this equation, you are subtracting each octet separately, since an IP Address is not a whole number.

To permit access to the network of 200.0.18.0 with a subnet of 255.255.255.0 you would use the following:

Using a Standard Access-List:
access-list 101 permit 200.0.18.0 0.0.0.255

Using an Extended Access-List:
access-list 101 permit ip 200.0.18.0 0.0.0.255 any

To permit access to the network of 10.4.0.0 with a subnet of 255.255.0.0 you would use the following:

Using a Standard Access-List:
access-list 100 permit 10.4.0.0 0.0.255.255

Using an Extended Access-List:
access-list 100 permit ip 10.4.0.0 0.0.255.255 any

Match an IP range

Take the range if 10.3.16.0 – 10.3.31.255. In order to find the mask, take the higher IP and subtract from it the lower IP.

10.3.31.255
10.3.16.0 -
0.0.15.255

In this case the wildcard mask for this range is 0.0.15.255.

To permit access to this range, you would use the following:

Using a Standard Access-List:
access-list 100 permit 10.3.16.0 0.0.15.255

Using an Extended Access-List:
access-list 100 permit ip 10.3.16.0 0.0.15.255 any

One thing to note is that each non-zero value in the mask must be one less than a power of 2, i.e. 0, 1, 3, 7, 15, 31, 63, 127, 255.

Match Everyone and Anyone

This is the easiest of Access-Lists to create, just use the following:

Using a Standard Access-List:
access-list 1 permit any or
access-list 1 permit 0.0.0.0 255.255.255.255

Using an Extended Access-List:
access-list 1 permit ip any any

Lab 29: Verify Standard Access Lists

Objective: To verify access-lists are configured correctly.
Prerequisite: Must have just completed the Standard Access List lab 28 .
Lab Equipment: We will be using Router2 and Router4. To start select Router 4, click on the button "Router 4" located at the top of the screen.

1. Our first step is to see if we can still ping Router 2 from Router 4. Connect to Router 4 and try to ping Router2's Ethernet 0 interface (24.17.2.2).

```
Router>enable
Router4#ping 24.17.2.2
```

2. If you get "UUUUU" then you have your access-list working correctly.

3. Now select Router 2 to verify that our access-lists are running on our interfaces, view the running configuration.

```
Router>enable
Router2#show running-config
```

4. You can also view what access-lists are applied to the interfaces by using the show ip interface command.

```
Router2#show ip interface
```

5. The command show access-lists will show you what access-lists you have created on the router. It will also tell you what lines have been used and how many packets they have either permitted or denied.

```
Router2#show access-lists
```

5. Continue on to Lab 30 without loading the Lab Navigator. This will save you the trouble of establishing the same ip Addresses again.

Lab 30: Extended Access Lists

Objective: To understand and gain experience in configuring extended access lists.
Lab Equipment: We will be using Routers 1, 2, and 4 for this lab.

Note: If you feel confident in configuring IP addresses and RIP, establish the configuration in the table below and then continue with step 9.

	Router1	Router2	Router4
Interface Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Interface Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240
RIP	On interfaces Ethernet 0 and Serial 0	On Ethernet 0	On Serial 0

IMPORTANT: If you have already done the standard access list lab then all you need to do is execute the command no ip access-group 1 on the E0 interface of Router2, and then start this lab at step 9.

```
Router2>enable
Router2#conf t
Router2(config)#interface ethernet0
Router2(config-if)#no ip access-group 1 in
```

1. Select Router1 from the menu bar 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 interface to 24.17.2.17 255.255.255.240. Don't forget 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
```

2. Select Router 2 from the menu bar and set the IP address on the Ethernet 0 interface to 24.17.2.2 255.255.255.240. Don't forget 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. Ping the Router 1's Ethernet 0 interface to ensure you have a connection (24.17.2.1).

```
Router2#ping 24.17.2.1
```

4. Select Router 4 from the menu bar and set the IP address on the Serial 0 interface to 24.17.2.18 255.255.255.240, then ping Router 1's Serial 0 interface (24.17.2.17).

```
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
```

5. Now that we have established IP addresses on all the interfaces we need to have a routing protocol enabled to facilitate communication between Router 2 and Router 4. We will use RIP as our routing protocol. Enable RIP on Router 1 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 Router 2 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. Finally on Router 4 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. Now that we have finished all of our background configuration verify that you can ping Router 2's Ethernet 0 (24.17.2.2) interface from Router 4.

```
Router4#ping 24.17.2.2
```

9. Our Extended Access List will do a couple of different things. First we will only allow telnet from the subnet off of Router1's Serial 0 to come into Router 1. Next we will allow anything from Router1's Ethernet 0 subnet to go anywhere. Connect to Router 1 and make sure that you are in configuration mode.

```
Router1#conf t
Router1(config)#
```

10. The first thing we are going to do is to allow only telnet from the subnet 24.17.2.16. This will be access-list 101 permit TCP 24.17.2.16 0.0.0.15 any eq telnet log. Notice I use the log command. This will show 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
```

11. Next we are going to permit anything from the subnet 24.17.2.0. Access-list 102 permit IP 24.17.2.0 0.0.0.15 any log

```
Router1(config)#access-list 102 permit ip 24.17.2.0 0.0.0.15 any log
```

12. We have established everything we would like to do; now we need to apply these access-lists to the interfaces. To take care of the subnet off Serial 0 we should apply access-list 101 to the Serial 0 inbound interface. Enter the interface configuration mode for Serial 0 and apply the access-list.

```
Router1(config)#interface serial0
Router1(config-if)#ip access-group 101 in
Router1(config-if)#exit
```

13. For Ethernet 0 we need to apply access-list 102 inbound.

```
Router1(config)#interface ethernet0
Router1(config-if)#ip access-group 102 in
Router1(config-if)#exit
```

14. Congratulations, you have completed the lab. Now continue on to the Verifying Extended Access-lists to make sure you have the access-lists setup correctly. You should do this without accessing the Lab Navigator.

Lab 31: Verify Extended Access Lists

Objective: To verify access-list are configured correctly.
Prerequisite: Must have completed the Extended Access-List lab.
Lab Equipment: We will be using Router 4. To select Router 4 click on the button "Router 4" located at the top of the screen.

1. We should now test and see if our access-lists are working properly. Connect to Router 4 and try to ping Router1's S0. You should not be able to ping the serial interface.

```
Router>enable
Router4#ping 24.17.2.17
```

2. Now that we verified the access-list is blocking ping we need to verify it will allow telnet. Connect to Router 1 and enable telnet access then set the password to boson.

```
Router>enable
Router#
Router#conf t
Router(config)#hostname Router1
Router1(config)#
Router1(config)#line vty 0 4
Router1(config-line)#login
Router1(config-line)#password boson
Router1(config-line)#exit
```

3. Now connect back to Router 4 and try to telnet into Router 1.

```
Router4#telnet 24.17.2.17
```

4. If you are given telnet access you should see the router prompt change to Router1. Now hold down the control-shift-6-x keys down all at once to change back to Router4. Then type `disconnect 1` to close your connection to Router 1. Congratulations one of your access-lists worked.

```
control+shift+6 followed by pressing x
Router4#disconnect 1
```

5. Now connect to Router 2 and see if you can ping Router 4's Serial 0 interface

```
Router>enable
Router2#ping 24.17.2.18
```

6. Why can't you ping the interface? Let's think about how the packet travels through the network. The packet starts at Router2, goes through Router1, and makes it to Router 4. Once it arrives at Router4 it is repackaged and sent back to Router 1. When Router4 repackages the packet, the packet's source ip becomes the destination ip and the destination IP becomes the source IP. When the packet encounters the access-list on

Router1's Serial 0 interface it is blocked because the packet's source IP is Router4's Serial 0 address.

7. Now see if you can ping Router1's Ethernet 0 interface (24.17.2.1)

```
Router2#ping 24.17.2.1
```

8. If you can, congratulations, see if you can further test it by telnetting to Router 1.

```
Router2#telnet 24.17.2.1
control+shift+6 followed by pressing x
Router2#disconnect 1
```

9. To verify that our access-lists are on our interfaces show the running configuration.

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

10. You can also view what access-lists are applied to the interfaces using the show IP interfaces command.

```
Router1#show ip interface
```

11. The command show access-lists will show you what access-lists you have created on the router. It will also tell you what lines have been used and how many packets they have either permitted or denied.

```
Router1#show access-lists
```

Lab 32: Named Access Control Lists

Objectives: Create a named ACL to deny all ping traffic from pc1 to Router 1 but enable access from Router 4 to Router 1 assuming we have to have the access list on Router 1.

1. Establish the configurations outlined in the table below before continuing. Use the 'winipcfg' command on PC1 to configure the IP address.

Device	Router 1	Router 4	PC1
Hostname	Router1	Router4	C:>
Ethernet 0/0		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 routing protocol on the two routers using the proper network statements.

3. Check to make sure you receive the routes on all routers with the show ip route command and verify you can ping Router 1 from PC1.

4. Prevent access to all ping traffic that originates from PC1 and is destined for Router 1. This access list can be located on Router 4 or Router 1. It makes more sense to have the access list located on the router closest to the source as possible; this helps keep unnecessary traffic off the backbone. For this example, however, we will have the access list located on Router 1 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
```

5. The first statement above defines the access list as extended. The second line denies any icmp traffic with a source address of only one host with ip 192.168.1.18 and destined for 192.168.1.1 with a wildcard of 0.0.0.0, which means to match the ip address exactly. Notice how we used the host command for the first part of the access list and the wildcard of 0.0.0.0 for the second part of the access list. The host command and wildcard of 0.0.0.0 both do the same thing. The log command is used for us to double check our work.

6. Next apply the access list to the Serial 0 interface on Router 1. The access list will have to be for the inbound traffic.

```
Router1(config-ext-acl)#exit
Router1(config)#interface Serial 0
Router1(config-if)#ip access-group deny_ping in
```

7. Now connect to PC1 and try to send a test ping to Router 1. Does the ping go through? Connect to Router 4 and try to send a test ping to Router 1 Serial 0.

8. Connect back to Router 1 and there should be two separate log messages. The first one is denying the ping from PC1 and the second is allowing the ping from Router 4.

Lab 33: Advanced Extended Access Lists

Objectives: Configure Extended Access Lists to filter out different types of network traffic:

- -Filter Network to Network Traffic
- -Filter Host to Host Traffic
- -Filter Network to Host Traffic

1. Establish the configurations outlined in the table below before continuing.

Device	Router 1	Router 2
Hostname	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 routing protocol on all routers using the proper network statements.

3. Check to make sure you receive the routes on all routers with the show ip route command and verify you can ping PC 1 from PC 2.

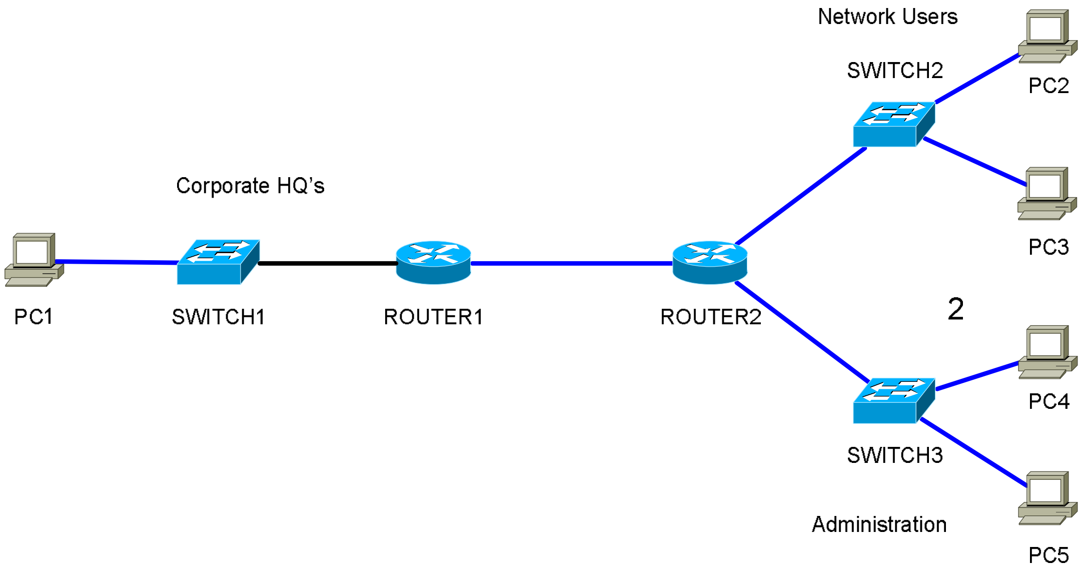
Network to Network Access List.

4. Looking at Figure 1 below. The first access list you are going to create will only allow traffic from the Administration network (PC4 – PC5) destined for the Corporate HQ network (PC1). To accomplish this you will use an extended access list. Since you are allowing all traffic you will be using IP as the protocol. Our access list should look something like this.

```
Access-list 100 permit ip 192.168.1.0 0.0.0.127 192.168.3.0 0.0.0.255 log
Access-list 100 permit ip 192.168.2.0 0.0.0.0 any
```

5. This access list is really simple because you are only allowing two things and denying all other traffic. Because there is an implicit deny at the end of each Access List you only need a permit statement for the pings and a permit statement for the Rip Broadcasts.

Figure 1



6. Now you need to apply the Access List to the interface. Since the traffic is coming from Router 2 and going to Router 1 you will have the access list on Router 1's Serial 0. The access list is checking all traffic coming inbound.

```
Router1#Conf t
Router1(config)#Interface Serial 0
Router1(config-if)#ip access-group 100 in
```

To test the access list try and ping PC 1 from PC2, PC3, PC4 and PC5. PC 2 and PC 3 should not be able to ping PC1 but PC 4 and PC 5 should. If this access list works congratulations continue on to the next step.

Host to Host Access List

7. The next part of the lab you will block access one individual workstation from accessing the central file server. Workstation 2 is a new employee and you do not want them to have access to your file server (PC5) for 30 days. To accomplish this you decide to implement an access list on Router 2 that will block access to PC 5 only for PC 2. For this section we are setting the access list manually. The administrator will be required to remove the list after the 30 days.

```
Access-list 101 deny ip host 192.168.1.130 192.168.1.3 0.0.0.0 log
Access-list 101 permit ip any any
```

8. For lab scenario purposes I always like to use the log command. This will show logging output on the screen 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 PC2.
9. Apply the access list to Router 2 FastEthernet 0/0 interface.

```
Router2#conf t
Router2(config)#interface FastEthernet 0/0
Router2(config-if)#ip access-group 101 in
```

10. Connect to PC2 and verify that you cannot ping PC5. Connect to PC3 and verify that you can ping PC5. Now connect to Router2 and verify the log statement is displayed matching your pings.

Network to Host Access List

11. Before you begin this access list remove the proceeding access lists from Router 1 and Router 2.

```
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
```

12. For the final scenario you will block all traffic to PC1 from the Network Users area in the topology. To accomplish this write an extended access list. The access list should look something like the following.

```
Access-list 102 deny ip 192.168.1.128 0.0.0.127 host 192.168.3.2 log
Access-list 102 permit ip any any
```

13. Apply this access list to Router 2 Serial 0 outbound.

```
Router2(config)#interface Serial 0
Router2(config-if)#ip access-group 102 out
```

To test this access list try and ping PC1 from PC2 or PC3. The pings should fail. You can also view the log file on Router 2.

Congratulations you have finished our Extended Access List labs!

Lab 34: Introduction to Telnet

Objective: To become familiar with establishing a telnet session between two routers.
Lab Equipment: We will be using Router 1 and Router 2 for this lab. To select "Router 1" click on the Router 1 button on the top of the screen.

Note: The simulator has limited telnet support outside of the commands shown within this lab.

1. Connect To Router 1. For this lab we will need to have the router accept telnet sessions and define the password we will use for these telnet sessions. On Router 1 get to Configuration mode and set the hostname to Router1. Then access the telnet lines. Each line in a router represents an active telnet session it can support. For our routers we support 5 lines so the command you need to use is line vty 0 4. This will cover all 5 lines.

```
Router>enable
Router#conf t
Router(config)#hostname Router1
Router1(config)#line vty 0 4
Router1(config-line)#
```

2. Now tell the router that we are going to require the use of a login password.

```
Router1(config-line)#login
```

3. Next define the password that will be used to establish a telnet session

```
Router1(config-line)#password boson
```

4. Now we have finished allowing telnet on Router 1. We need to setup an IP address for its Ethernet 0 interface. Get into the Ethernet 0 interface

```
Router1(config-line)#exit
Router1(config)#interface Ethernet 0
```

5. Set an IP address of **34.25.67.18 255.255.255.224** and enable the interface

```
Router1(config-if)#Ip address 34.25.67.18 255.255.255.224
Router1(config-if)#no shut
```

6. Next connect to Router 2 and set its hostname to Router2. Then enter its Ethernet 0 interface

```
Router>en
Router#conf t
Router(config)#hostname Router2
Router2(config)#interface Ethernet 0
Router2(config-if)#
```

7. Set the IP address for this interface to: **34.25.67.2 255.255.255.224**.

```
Router2(config-if)#Ip address 34.25.67.2 255.255.255.224
Router2(config-if)#no shutdown
```

8. Now exit back to Privilege mode.

```
Router2(config-if)#end
```

9. We will want to telnet to Router 1. The IP address we will use is the Ethernet 0's IP of **34.25.67.18**

```
Router2#telnet 34.25.67.18
```

10. The first thing you will notice is the screen will now ask you for a password. Enter in the password boson and press enter. The message box will tell you that we offer a limited support for telnet. You should notice the hostname of the router changed to "Router1" we have established a telnet session to Router1. Now hold down the **control-shift-6 keys** all at once for a second, then release and immediately press the **x** key**.** You should notice the hostname changed back to Router2. You are now back on Router2.

```
Password:
Router1#
<Control> + <Shift> + <6> followed by <x>
Router2#
```

11. Type the command show sessions. This will let you view all active telnet sessions you have. To resume the telnet session pick the number you would like to resume (in our case we only have 1) and type the command resume 1.

```
Router2#Show sessions
Router2#resume 1
Router1#
```

12. Now the hostname has changed back to Router1. Press the control-shift-6 combination followed by the x key and you will be back into Router2 again.

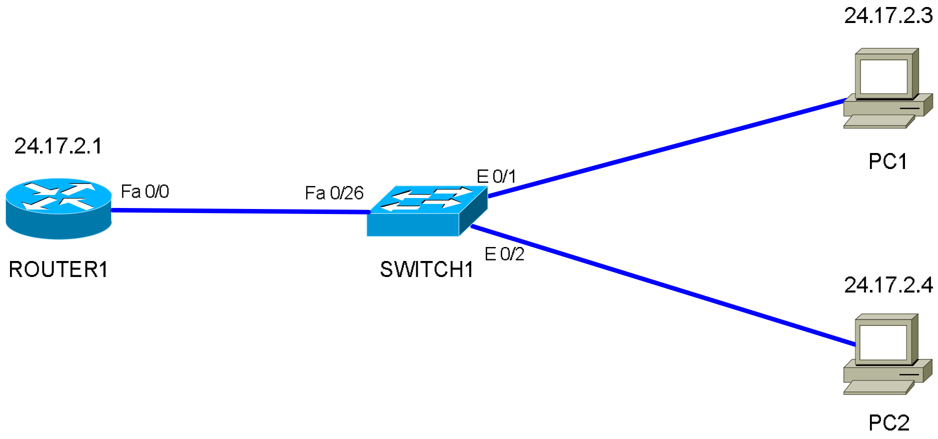
```
Router1#
<Control> + <Shift> + <6> followed by <x>
Router2#
```

13. Now to disconnect the session type disconnect 1.

```
Router2#disconnect 1
```

Lab 35: Introduction to VLANS

Objective: To become familiar with the benefits of Vlans on your LAN using a 1900 Series switch.
Lab Equipment: You will be using Router 1, Switch 1, eStation 1 and eStation 2



We are going to configure our router and switch to support VLANs. The goal of this lab is to setup your eStation's so that they can ping each other through the switch. We will then have you change the VLANs on the switch and observe that we cannot ping any longer. Once we notice we cannot ping the router we will change the configuration on the switch so that the eStation's are on the same VLAN and observe that we can ping each other again.

1. Lets start by configuring the ip address on Router1 Fast Ethernet 0/0. Connect to Router 1 enter interface **Ethernet 0** and set the IP address of **24.17.2.1 255.255.255.0**

```
Router>enable
Router#
Router#conf t
Router(config)#hostname Router1
Router1(config)#
Router1(config)#interface Fast0/0
Router1(config-if)#ip add 24.17.2.1 255.255.255.0
Router1(config-if)#no shut
```

2. Next connect to **eStation 1** and set the IP address of **24.17.2.3 255.255.255.0** with default gateway **24.17.2.1**.

```
C:>winipcfg
```

3. Now connect to **eStation 2** and set the ip address to **24.17.2.4 255.255.255.0** with default gateway **24.17.2.1**.

```
C:>winipcfg
```

4. You should now be able to ping Router 1 and eStation 1 from eStation2

```
C:>ping 24.17.2.1
C:>ping 24.17.2.3
```

5. Now connect to Switch 1 and let’s set the VLANs up. The switch automatically has VLAN 1 setup on all ports. In this case we will setup a separate VLAN for the eStation's. Start off by creating the vlan.

```
>en
#config t
(config)#vlan 22 name pcs
```

6. Now that we have created the VLANs we need to assign the ports to it. Lets start by assigning port 1 for eStation1 to the new VLAN.

```
(config)#int e0/1
(config-if)#vlan-membership static 22
```

7. Now that we have assigned the VLAN to the port lets see if we can ping from eStation2 to Router 1 and eStation1. Connect back to eStation2 and try to ping the other two devices.

```
C:>ping 24.17.2.1
C:>ping 24.17.2.3
```

Lets think about what just happened. We were able to ping from eStation2 to Router1 but we were not able to ping from eStation2 to eStation1. Why?
On the switch we set VLAN 22 to only cover port 1. That means ports 2-12 and the two fast Ethernet ports were still on VLAN 1. So when our ping packets came into the switch from eStation2 they were tagged with VLAN 1.This means they can only go out of ports that are tagged with VLAN 1. (We will find out later there are exceptions to this rule.) That means it could not go out port 1 (to eStation 1).

8. Lets connect back to our switch and setup port 2, which is where PC2 is connected, to be included in VLAN 22.

```
(config-if)#exit
(config)#int e0/2
(config-if)#vlan-membership static 22
```

9. Now connect back to eStation2 and repeat the pings again to Router1 and eStation1.

```
C:>ping 24.17.2.1
C:>ping 24.17.2.3
```

What did we notice that was different? Now we could ping eStation1 but not Router 1. When the packet came in it was tagged with VLAN 22. This meant it could only go out port1, which was eStation 1 for the purposes of this lab. This is what we wanted to accomplish.

10. Connect back to the switch and let's view our VLAN assignments using some new show commands. Show VLAN and show VLAN-membership are two different ways to view the VLAN port assignments for the switch.

```
(config-if)#end
#show vlan
#show vlan-membership
```

10. Finally connect back to the switch and assign fastethernet 0/26 to the vlan 22 we have created. This will allow us to ping all devices.

```
#conf t
(config)#interface FastEthernet 0/26
(config-if)#vlan-membership static 22
```

11. Verify your pings work by sending test pings from Router 1 to PC 1 and 2 and From PC 1 and 2 to Router 1.

Lab 36: Virtual Trunking Protocol (VTP)

Objectives:

- Configure Vlans on the Catalyst 2950 switches.
- Assign vlans to multiple ports.
- Configure VTP protocol 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 IP addresses and hostnames to both Switch 3 and Switch 4 according to the table below.

Device	Switch 3	Switch 4
Hostname	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#
```

2. Verify that the switches are connected together by pinging switch 3 from switch 4.

3. Add vlan 8 and vlan 14 to Switch 3 and assign ports 0/2-0/5 for vlan 8 and ports 6-10 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)#
```

4. Use the show vlan command to verify your configurations are correct.

```
Switch3(config)#exit
Switch3#show vlan
```

5. By default the Catalyst switch is configured as a VTP Server. We would like to have Switch 3 setup as a VTP Server and switch 4 setup 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
```

Connect to Switch 4 to establish the VTP configuration

```
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 carry over the vlan configurations from Switch 3 to Switch 4. To accomplish this enable trunking on the port that links between the two switches. Your encapsulation will only be 802.1q because that is the only supported encapsulation for the 2950.

```
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
Switch3(config-if)#end
```

7. After this configuration you should be able to view the vlans from Switch 3 on Switch 4. To verify the vlan configurations use the show vlan command on Switch 4. Also the show vtp status command will display some VTP specific information.

```
Switch4# show vlan
Switch4# show vtp status
```

Lab 37: OSPF Single Area Configuration and Testing

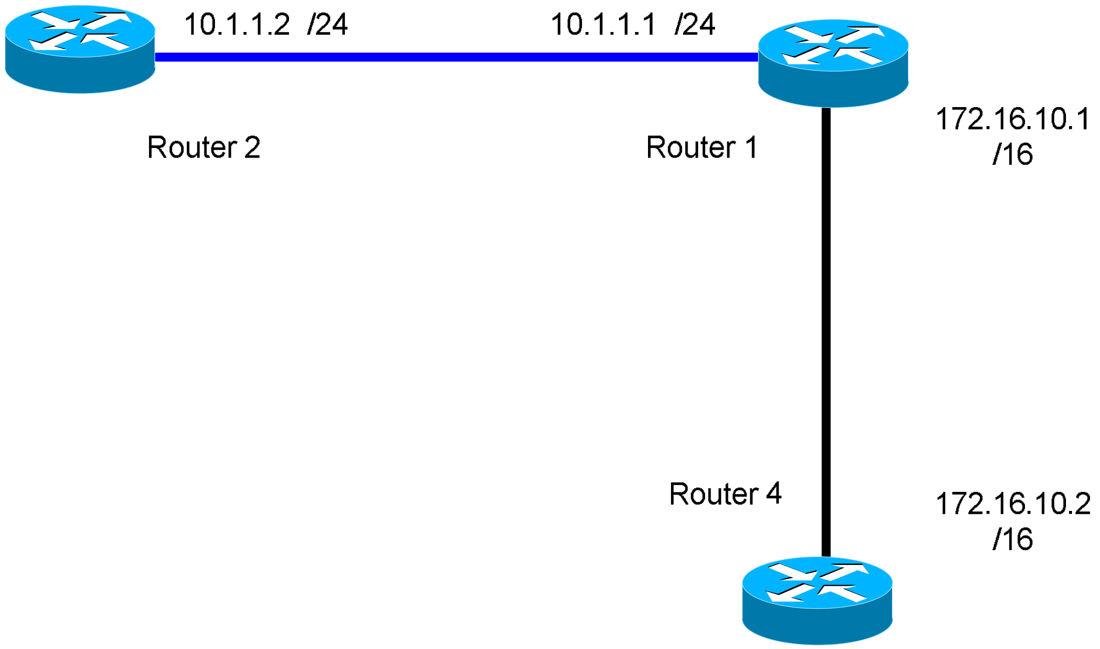
Objective: We will configure Routers 1, 2, and 4 with IP addresses and OSPF Routing Protocol

Goals:
1) Set our hostname and get our interfaces up.
2) Configure OSPF routing protocol
3) Select the directly connected networks
4) View our routing table
5) View the OSPF protocol information

IP Addresses: Please set these IP addresses on the interfaces of your routers.

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.0.0		172.16.10.2 255.255.0.0

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



2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP address setup correctly let’s get into configuring OSPF as our routing protocol. This is very easy to do; first we need to get into router configuration mode on Router 1. What command does this?

```
Router1#
Router1#config Terminal
Router1(config)#
```

4. Now enter the command to configure the router for OSPF use the Process ID number 100.

```
Router1(config)#router ospf 100
Router1(config-router)#
```

5. Add the network(s) that Router 1 is directly connected to. What statements will do this?

```
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
```

6. Now that you have Router 1 configured for OSPF connect to Router 2 and enter configuration mode.

```
Router2#
Router2#config Terminal
Router2(config)#
```

7. Add OSPF routing protocol to the router. What command does this?

```
Router2(config)#router ospf 100
Router2(config-router)#
```

8. Add the network(s) that Router 2 is directly connected to. What statements will do this?

```
Router2(config-router)#network 10.1.1.0 0.0.0.255 area 0
```

9. Now that you have Router 2 configured for OSPF connect to Router 4 and enter configuration mode.

```
Router4#
Router4#config Terminal
Router4(config)#
```

10. Add OSPF routing protocol to the router. What command does this?

```
Router4(config)#router ospf 100
Router4(config-router)#
```

11. Add the network(s) that Router 4 is directly connected to. What statements will do this?

```
Router4(config-router)#network 172.16.0.0 0.0.255.255 area 0
```

12. Now we should have OSPF running on all three of our routers. Type <ctrl> Z to exit to privileged mode and let's see if we can ping non-directly connected routers. From Router 2 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Let's try it!

```
Router2#ping 172.16.10.2
```

13. Next let's connect to Router 4 and ping Router 2's Ethernet 0 interface with IP address 10.1.1.2

```
Router4#ping 10.1.1.2
```

14. If you can ping both devices, CONGRATULATIONS you are routing. If you were not successful, trace yourself back through the steps. Now let's view our routing table on our Router 2. What command will do that?

```
Router2#show ip route
```

15. Lets view the specific IP routing protocol information on our router. What command will do this?

```
Router1#show ip protocols
```

16. What command will display the OSPF database?

```
Router1#show ip ospf database
```

17. What command will display all of the OSPF neighbors?

```
Router1#show ip ospf neighbor
```

18. What command will display all interfaces of the router that are running OSPF?

```
Router1#show ip ospf interface
```

OSPF Review

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

2. After you have configured your IP address on each interface verify that you can ping your directly connected neighbors. That means when you are on Router 1 you should be able to ping Router 2's Ethernet 0 interface and Router 4's Serial 0 interface.

3. Now that we have our IP address setup correctly let's get into configuring OSPF as our routing protocol. This is very easy to do; first we need to get into router configuration mode on Router 1. What command does this?

4. Now enter the command to configure the router for OSPF with the Process ID # 100.

5. Add the networks that Router 1 is directly connected to. What statements will do this?

____

6. Now you have Router 1 configured for OSPF connect to Router 2 and enter configuration mode. ___________________________

7. Add OSPF routing protocol to the router with the Process ID # 100. What command does this? ___________________________

8. Add the networks that Router 2 is directly connected to. What statements will do this?

____

9. Now that you have Router 2 configured for OSPF connect to Router 4 and enter configuration mode. ___________________________

10. Add OSPF routing protocol to the router with the Process ID # 100. What command does this? ___________________________

11. Add the networks that Router 4 is directly connected to. What statements will do this?

____

12. Now we should have OSPF running on all three of our routers. Type <ctrl> Z to exit to privileged mode and let's see if we can ping non directly connected routers. From Router 1 you should now be able to ping Router 4's Serial 0 interface with IP address 172.16.10.2. Let's try it!

13. Next let's connect to Router 4 and ping Router 1's Ethernet 0 interface with IP address 10.1.1.2

14. If you can ping both devices, CONGRATULATIONS you are routing. If you were not successful, trace yourself back through the steps. Now let's view our routing table on our Router 4. What command will do that? __________________________ How many OSPF Routes do you see? ______

15. Lets view the specific routing protocol information on our router. What command will do this? __________________________

What networks are you routing for? __________________________

What is the default administrative distance for OSPF routes?

____

16. What command will display the OSPF database for the router?

____

17. What command will display all of the OSPF neighbors?

____

How many neighbors do you have? _______

18. What command will display all interfaces of the router that are running OSPF?__________________________

OSPF Summary

Open Shortest Path First (OSPF) is a dynamic-link state, hierarchical IGP (Interior Gateway Protocol) routing protocol based on open standards. It was designed as a replacement to RIP and was derived from an early version of IS-IS protocol (Intermediate System to Intermediate System protocol). OSPF is a robust protocol whose features include: least-cost routing, multipath routing, and load balancing. The shortest path through the network is calculated by using the Dijkstra algorithm.

Cisco uses it own implementation of the OSPF standards with additional features that are important for interoperability.

Once we configure the router for OSPF it must start the process of learning its environment by going through a few phases of initialization. It goes through a process of using “hello” packets to determine its neighbors and developing adjacencies (relationship for exchanging routing updates) with them. The router will then start the ExStart phase; which is the initial database exchange. Next is the Exchange phase where the DR (Designated Router) sends the routing info and receives an ack (or acknowledgement) receipt from the new router. During the Loading phase the information is complied using a routing table. Once the router finishes its calculations, it will move into what is known as being in its full state and it is now an active member of the network.

The rest of this lab is a walk through lab that you can complete on the program or just follow through the steps. The output displayed below demonstrates how to complete an OSPF Configuration. Some time is taken to explain in detail what each command does.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#ip address 10.1.1.1 255.255.255.0
Router(config-if)#no shut
00:12:33: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
Router(config-if)#exit
Router(config)#hostname Router1
Router1(config)#int s0
Router1(config-if)#ip address 172.16.10.1 255.255.255.0
Router1(config-if)#no shut
00:15:30: %LINK-3-UPDOWN: Interface Serial0, changed state to up
00:15:35: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
Router1(config-if)#exit
```

We first want to configure Router1 for OSPF. To enable OSPF as the routing protocol we only need to type: router ospf 100. 100 is a process identification number (Integer from 1 to 65,535) that is used to initialize the protocol on the router. Unlike the EIGRP AS number, the process identification number does not have to be the same for all of the routers within the OSPF areas. The networks that are added to the OSPF session make up the area. We can see this below in the router output. Notice the new mode we have entered Router1(config-router)# that tells us we are configuring the router.

```
Router1(config)#router ospf 100
Router1(config-router)#
```

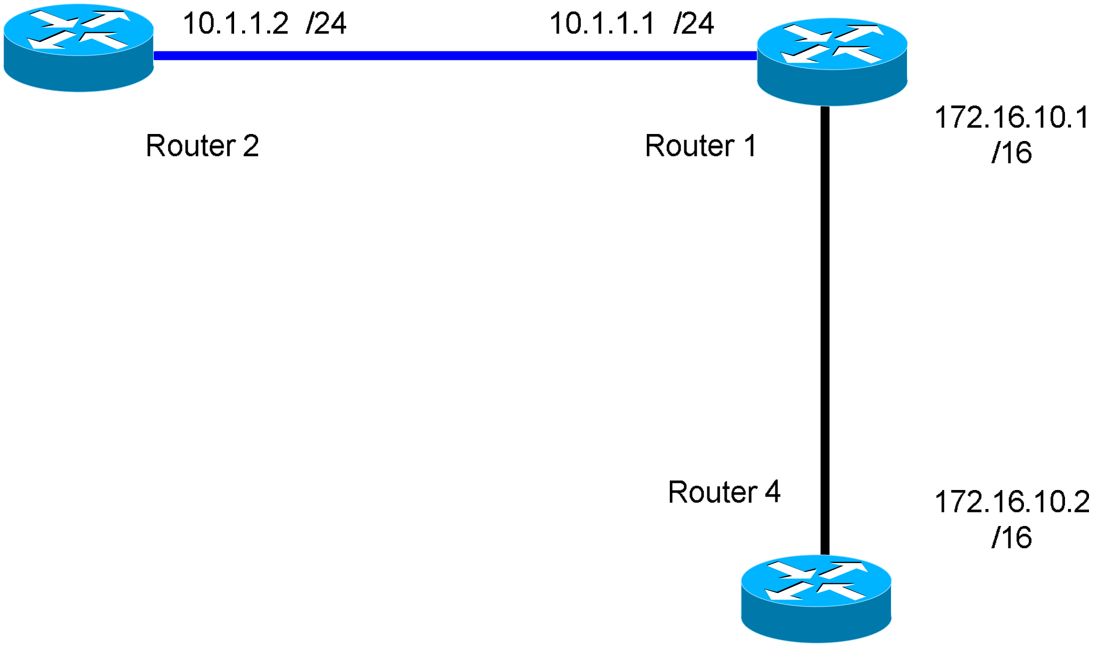
Now that we have OSPF running on Router1 we need to tell the router which networks it is connected to, its wildcard-mask and assigned OSPF area. We do this by using the network statement that contains the IP address, wildcard-mask and area-id. This means that every interface of our router that is directly connected to an active network needs a network number. We will have some networks using the same IP addressing schemes with different subnets, and some are using entirely different addressing schemes. Look at the diagram below. In this diagram we have three different kinds of addressing schemes. Let's look at these in more detail. On Router1 we have an IP address of 10.1.1.1 with a /24 subnet mask.

When entering the network statement we are required to enter the class part of the address, wildcard-mask and area-id. (Integer from 0-4,294,967,295) For example on Router1 we have already issued the command **router ospf 100**. We then need to specify the directly connected network, wildcard-mask and area-id to Router1 so the router can advertise these routes in its routing table. To do this we would need to type: network 10.0.0.0 0.0.0.255 area 0. Next we need to configure the Serial interface with the IP address, wildcard-mask and area-id, to do this we would type: network 172.16.0.0 0.0.0.255

area 0. The wildcard-mask is used for troubleshooting specific links by either adding or removing them.

What network statement would we need to use on Router2?

____(See the answer below the diagram.)



The answer is network 10.0.0.0 0.0.0.255 area 0. The network statement for the Ethernet interface is the same for Router1 and Router2. On Router1 what network statement would we need for the Serial interface? For this network statement we used the classful portion of the address 172.16.10.1, wildcard-mask and area-id, which would be network 172.16.0.0 0.0.0.255 area 0 Now that we understand the network command lets enter it on our Router1.

```
Router1(config-router)#network 10.0.0.0 0.0.0.255 area 0
Router1(config-router)#network 172.16.0.0 0.0.0.255 area 0
Router1(config-router)#
```

Now we will use the show running-config command to see how we did on the configuration of OSPF on Router1.

```
Router1(config-router)#exit
Router1(config)#exit
```

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

We should be able to confirm that the OSPF process ID was defined as 100 and two networks were added to OSPF area 0.

We need to connect to Router2 and configure the Ethernet interface, network IP address, subnet mask and router hostname. Let's select Router2 from the window pull down menu and then select Tile so we can see both routers 1 and 2. When we connect we are going to set a hostname to Router2, then set the IP address to the table above and configure OSPF.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router2
Router2(config)#int e0
Router2(config-if)#ip address 10.1.1.2 255.255.255.0
Router2(config-if)#no shut
Router2(config-if)#exit
00:21:23: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
00:21:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
Router2(config)#
```

Now add the OSPF process identification number and network statement.

```
Router2(config)#router ospf 100
Router2(config-router)#network 10.1.1.0 0.0.0.255 area 0
Router2(config-router)#exit
Router2(config)#exit
Router2#
```

We should now have OSPF running on our network between Router1 and Router2. We need to get Router4 setup.

We need to connect to Router4 and configure the Serial interface, network IP address, subnet mask and router hostname. Let's select Router4 from the window pull down menu and then select Tile so we can see all three routers 1, 2 and 4. When we connect we are going to set a hostname to Router4, then set the IP address to the table above and configure OSPF.

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router4
Router4(config)#int s0
Router4(config-if)#ip address 172.16.10.2 255.255.255.0
Router4(config-if)#no shut
Router4(config-if)#exit
```

Now add the OSPF process identification number and network statement.

```
Router4(config)#router ospf 100
Router4(config-router)#network 172.16.0.0 0.0.0.255 area 0
Router4(config-router)#exit
Router4(config)#exit
Router4#
```

Now we should have OSPF running on all three of our routers. We will use the ping command to test the connectivity between the routers. We need to connect to Router1. First we will ping Router4's Serial 0 interface with IP address 172.16.10.2 from Router1.

```
Router1#ping 172.16.10.2
```

Next we will ping Router2's Ethernet 0 interface with IP address 10.1.1.2 from Router1.

```
Router1#ping 10.1.1.2
```

Now let's see if Router2 can talk to Router1 and if Router4 can talk to Router1. Connect to Router2 and ping Router1.

```
Router2#ping 10.1.1.1
```

Connect to Router4 and ping Router1.

```
Router4#ping 172.16.10.1
```

If we can ping all devices, CONGRATULATIONS our routers are talking to each other in both directions and we are routing.

Now let's verify proper OSPF interface configuration with the show ip ospf interface command. This is an excellent command for learning all of the interface information. The data will include: interface IP address, area assignment, Process ID, Router ID, cost, priority, network type, timer intervals and adjacent neighbor information. We can also see the DR/BDR information when it is applied.

Connect to Router1.

```
Router1#show ip ospf interface
```

Our last command will show us all of the important information concerning neighbor and the adjacency state. This is also where the DR or BDR information is displayed if it is configured.

```
Router1#show ip ospf neighbor
```

Cisco Press, 800 East 96th Street, Indianapolis, Indiana, 46240, or toll-free 800-858-7674.

License

Boson Trademarks

Other Trademarks

Version: 060104a

ISBN: 1-58720-131-3

First Edition June 2004