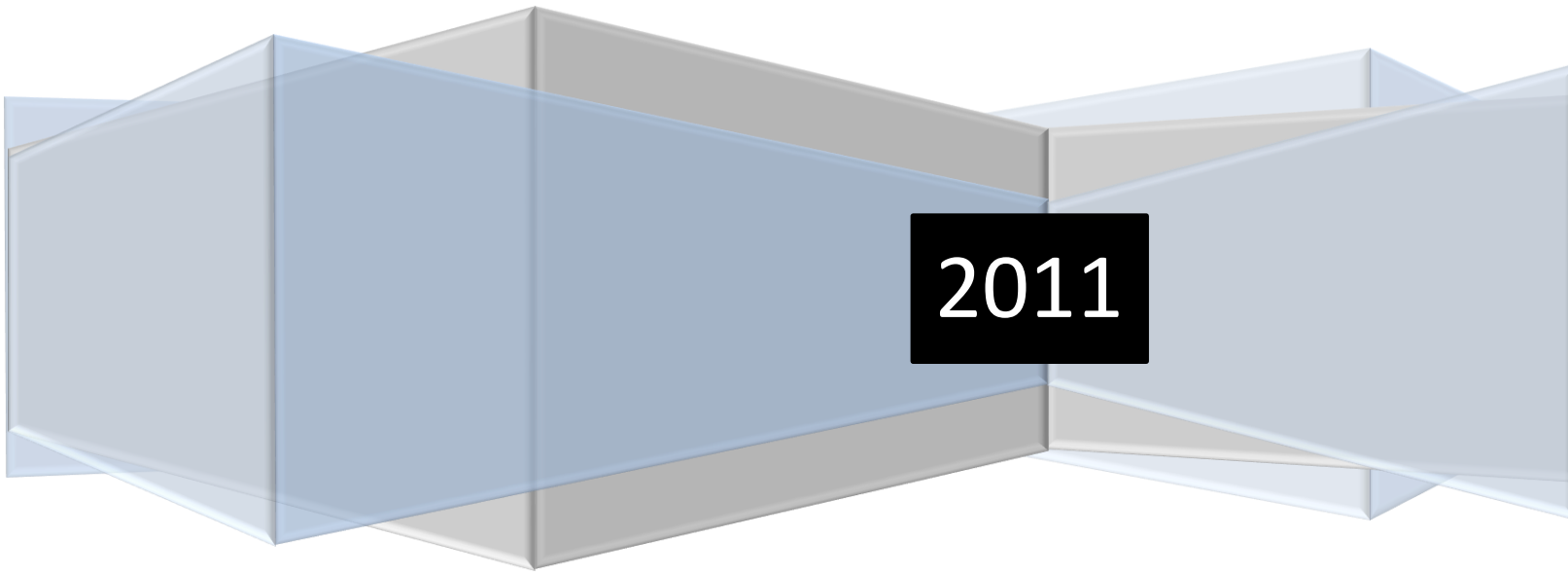
John Mason Institute
A Foundation For Your Future

CCNA LABS

GNS3 and Live Rack Labs

John Chislett
Donald Patch
Andy Brown
Version 1.2



Labs

Introduction Labs

Lab 1 Connecting to a router or switch

Lab 2 Logging in to the router, enable, configuration,

Lab 3 show command

Lab 4CDP Cisco Discovery Protocol

Lab 5 Password and secret password

Lab 6host name and motd

Lab 7Copy Command

Lab 8 configure ip address and show ip interface brief

Switch labs

Router labs

SDM Labs

Wireless Labs

Challenge Labs

Gns3 Set up

For most of these labs we'll be using the 3640 routers. On some labs there will also be PCs which will be configured into the labs. You can look at the top of each lab and there will be a diagram as well as a table with the description of the router, IOS and the interfaces and ip addresses of those interfaces.

The labs themselves will be configured in the files marked in the GNS3 Labs folder.

Lab 1

Description of lab: **Connecting to a router or switch**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640

In this lab we will introduce you to the basic commands and familiarize you with the GNS3 emulator.

First to start this lab double click on the "Lab 1.net" file in the GNS3 folder. You will use this file for Labs 1 - 6

This will open the lab with the router (3640) in the middle pane. You will have to hit 2 icons to get the lab started. The first is the green "start" arrow (this will turn on the router) and the second is the console window this will open a console port using putty to the router (and in the future routers) that you will be configuring.

After you open the console port you will see the starting output from the router which will eventually stop at the point shown directly below. Please type the letter "n" to indicate "no":

"--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: {**n"**

You will notice that the "n" at the end of that statement is bolded. That is an indication of a command you need to type. In the future all commands that you need to type will be bolded, this way you can tell the difference between output and typed commands

% Please answer 'yes' or 'no'.
Would you like to enter the initial configuration dialog? [yes/no]: **n**

Again you will notice that the "n***" is bolded. This indicates another command you will type.***

Press RETURN to get started!

After you hit the "enter" key you will see the "Router>***" prompt, which indicates that you are in "user mode". Type the command "***enable***" to put the router in privileged mode.***

Router>
Router>**enable**
Router#

You will now notice that there is a "#" character after the word router. This indicates that you are in privileged mode.

Next type "disable***" to put the router back to user mode and then "***exit***" to reset the router***

Router#**disable**
Router>
Router>**exit**

Lab 2 Description of lab: **Enable and Configure Terminal**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640

***Again you will open this lab, just as you did in Lab 1 (using the Lab1 file), unless you have already completed that lab, in which case you have the console port window open. In this lab we will enter both the privileged mode and then the configuration mode. The Configuration mode is sometimes called "*Global configuration mode".**

Router>
Router>**enable**
Router#
Router#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#
Router(config)#**end**
Router#
Router#
Router#**exit**

Press RETURN to get started.

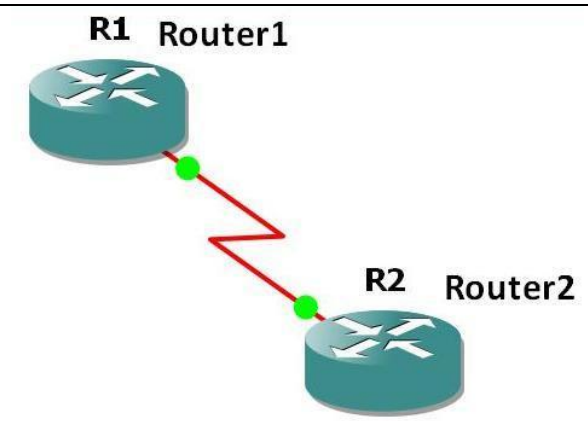
Lab 3 Description of lab: **Show command and "Do" command**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640
Hostname Name	

Start by opening the Lab1 file in the GNS3 folder and follow the steps to turning on the router and opening the console port.
In this lab we will be using some of the show commands in enable mode. These commands are most useful for troubleshooting and generally getting information on your router.

Router>
Router>**enable**
Router#**show ?**
access-lists List access lists (this gives you details of access lists and the permit and deny statements)
cdp CDP information(this gives you layer 2 info on devices directly connected)
Clock Display the system clock
flash:display information about flash: file system (the files in memory)
history Display the session command history (gives you a list of your recent commands)
host s IP domain-name, lookup style, nameservers, and host table
Interfaces Interface status and configuration (these are the Ethernet and serial ports)
ip IP information
protocols Active network routing protocols
running-config Current operating configuration
users Display information about terminal lines
version System hardware and software status

Lab 4 Description of lab: **CDP (Cisco Discovery Protocol)**

GNS3 file Lab 4.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Router1	Router2

CDP is a Data Link Protocol (Layer 2 of the OSI model). This means that CDP is non-routable and can only negotiate directly linked devices. It also means that you can find out basic information about directly connected devices before they have been configured, using CDP.

First enter the global configuration mode and give the hostname Router1 to R1 and enable serial 0/0

Router>**enable**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#
Router1(config)#**interface Serial 0/0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**no keepalive**

By default all interfaces are turned off so the "no shutdown***" command will enable the interface and the "***no keepalive***" will keep the interface from shutting off by protocol. Now let's configure R2 and we'll give it the hostname of Router2.***

```
Router>enable
Router#conf t
Router(config)#hostname Router2
Router2(config)#
Router2(config)#interface Serial 0/0
Router2(config-if)#no shutdown
Router2(config-if)#no keepalive
```

Now let's put Router1 into enable mode and type the "show cdp interface***" command.***

```
Router1(config-if)#exit
Router1(config)#exit
Router1#
Router1#show cdp interface
Serial0/0 is up, line protocol is up
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Router1#
```

Now that we can see that the interfaces are broadcasting CDP updates. Let's find out information on our directly connected devices. We'll type in the "show cdp neighbors***" command and we can see what is connected to Router1***

```
Router1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Router2          Ser 0/0           174          R S I     3640      Ser 0/0
Router1#
```

As you can see, Router2 appears in the output from this command. It also includes information about the device. If you want more details type the same command and add details to it "show cdp neighbors detail***"***

Router1#**show cdp neighbors detail**

Device ID: Router2
Entry address(es):
Platform: Cisco 3640, Capabilities: Router Switch IGMP
Interface: Serial0/0, Port ID (outgoing port): Serial0/0
Holdtime: 157 sec

Version:
Cisco IOS Software, 3600 Software (C3640-JK9S-M), Version 12.4(13a), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Tue 06-Mar-07 20:25 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''

Router1#

Since Router2 is the only device attached to Router1 the "show cdp neighbors detail" command only shows details about that router. If we had more devices attached the command would show each device one at a time. Let's compare the output from that command with the output from "show cdp entry Router2".

Router1#show cdp entry Router2

Device ID: Router2
Entry address(es):
Platform: Cisco 3640, Capabilities: Router Switch IGMP
Interface: Serial0/0, Port ID (outgoing port): Serial0/0
Holdtime : 172 sec

Version :
Cisco IOS Software, 3600 Software (C3640-JK9S-M), Version 12.4(13a), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Tue 06-Mar-07 20:25 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''

In this case (again since Router2 is the only device attached to Router1) the output is practically the same. This command is more useful when you have many devices attached to your router and you don't want to scroll through pages of information. This command is case sensitive so if you only typed in "router2" you wouldn't have gotten any information.

The "show cdp**" will show you how long between updates and holdtime.**

```
Router1#show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
Router1#
```

You can change those settings with the "cdp timer 4" and the "cdp holdtime".

```
Router1#conf t
Router1(config)#cdp timer 45
Router1(config)#cdp holdtime 60
Router1(config)#
```

Now compare the output with the first one.

```
Router1#show cdp
Global CDP information:
    Sending CDP packets every 45 seconds
    Sending a holdtime value of 60 seconds
    Sending CDPv2 advertisements is enabled
Router1#
```

You can disable CDP for the router with the command "no cdp run***" and then turn it back on with "***cdp run***".***

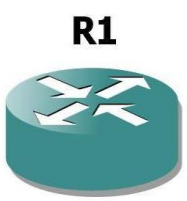
```
Router1#conf t
Router1(config)#no cdp run
Router1(config)#cdp run
```

If you want to turn it off only on a specific interface

```
Router1(config)#int s0/0
Router1(config-if)#no cdp enable
Router1(config-if)#end
Router1#show cdp interface
```

Now look at the output. Do you see the interface Serial0/0?

Lab 5 *Description of lab:* **Password and secret password**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640

Router>
Router>**enable**
Router#
Router#
Router#**configure terminal**
Router(config)#
Router(config)#**enable password johnmason**
Router(config)#
Router(config)#**end**
Router#
Router#**exit**
Router>**enable**
Password: (type in "johnmason")
Router#
Router#**conf t**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#**enable secret cisco**
Router(config)#end
Router#**exit**
Router>enable
Password: **cisco** (this time type in "cisco" the "johnmason" password has been superseded by the "secret" password)
Router#**conf t**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
TO REMOVE PASSWORD

Router>**enable**
Password:**cisco**
Router#**conf t**
Enter configuration commands, one per line. End with CNTL/Z.

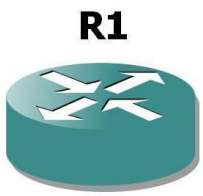
Router(config)#**no enable secret** (removes the "secret" password)
Router(config)#end
Router#

Press RETURN to get started.

Router>**enable**
Password:**johnmason** (now you don't type "cisco", but type "johnmason" as the "secret" password has been removed)
Router#**conf t**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#**no enable password*(removes the password)***
Router(config)#**end**
Router#ex
Press RETURN to get started.

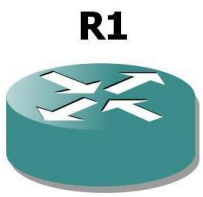
Router>**enable**
Router# **(this time no password was required as you removed the "enable password" command)**
Router#
Router#

Lab 6 Description of lab: **host name and motd**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640
Hostname Name	Router1
Ethernet 0/0	
Ethernet 0/1	
Serial 0/0	
Serial 0/1	

Router>
Router>**enable**
Router#conf
Router#configure t
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router1
Router1(config)#
Router1(config)#banner motd zJohn Mason Institutez
Router1(config)#end
Router1#
Router1#**exit**
Press RETURN to get started.
John Mason Institute
Router1>
Router1>

Lab 7 Description of lab: **Copy Command**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640
Hostname Name	
Ethernet 0/0	
Ethernet 0/1	
Serial 0/0	
Serial 0/1	

The output from this command will show you any and all of the active configuration information that is currently running on this router

Router>
Router>**enable**
Router#
Router#**show running-config**
Building configuration...
(output ad nauseam)
Next we will look at the information in NVRAM. Since we have not saved the configuration there is none to see.
Router#**show startup-config**
startup-config is not present

Now we will save the current running-config to the startup-config
Router#**copy running-config startup-config**
Destination filename [startup-config]?
Building configuration...

[OK]

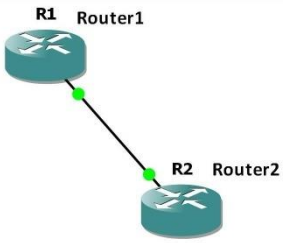
Now when we look at the startup-config we see the same output as the current running-config
Router#**show startup-config**
Using 653 out of 129016 bytes!
version 12.4
(output ad nauseam)

If we feel that we want the router to start with no startup- config again, we can erase the startup-config and reload the router

Router#**erase startup-config**
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]
[OK]
Erase of nvram: complete
Now let's look at the startup-config and we'll see that it's now empty again.

Router#**show startup-config**
startup-config is not present
Router#

Lab 8 Description of lab: **Interface configuration**

GNS3 file Lab 8.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Router1	Router2
Ethernet 0/0		
Ethernet 0/1		

First let's get to global configuration mode and give R1 the hostname of Router1
Router>**enable**
Router#**conf t**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#**hostname Router1**

Now let's configure ethernet 0/0 (it's the interface that is connected to Router2). After you get to the interface type a "?" to see what some of the commands are.
Router1(config)#**int ethernet 0/0**
Router1(config-if)#
Router1(config-if)#**?**
Interface configuration commands:

Some of the more important commands are:

description	Interface specific description
exit	Exit from interface configuration mode
ip	Interface Internet Protocol config commands
shutdown	Shutdown the selected interface

The "description" command will allow you to give the interface a description that can be viewed in the "show interface" command, when you are in enable mode. The "exit" command will take you out of the interface mode and put you in global configuration. Whereas "end" will take you completely out of Global configuration and put you in enabled mode. The "ip" command is one that you will use quite a bit in future labs to configure the ip address of the interface. And the "shutdown" command is used most often as the "no shutdown" command to take the interface out of its default state of being disabled.
Let's use 2 commands to see what the output is after we type in "no shutdown" and "description"

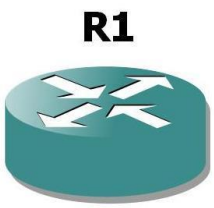
Router1(config-if)#
Router1(config-if)#**description Ethernet interface on Router1**
Router1(config-if)#**no shutdown**
Router1(config-if)#
*Mar 1 00:08:19.575: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Mar 1 00:08:20.575: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to up
As you can see the output from the router after this command has been given now states that "Ethernet0/0, changed state to up". Let's go to privileged mode and see what the "show interfaces" command outputs for us on Ethernet0/0.
Router1#**show interfaces**
Ethernet0/0 is up, line protocol is up
Description: Ethernet interface on Router1
Now let's go to Router 2, give it a hostname of Router2, and enable the interface (Ethernet 0/0) that is connected to Router 1.
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config)#**interface ethernet 0/0**
Router2(config-if)#**no shutdown**
Router2(config-if)#
Router2(config-if)#**end**
Router2#sh
Router2#**show cdp neighbors**
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
Router1	Eth 0/0	151	R S I	3640	Eth 0/0

Router2#
This shows you the device (Router1) that is connected to this router (Router2) and the interface that it is connected by.

Lab 9

Description of lab: **configure ip address and show ip interface brief**

GNS3 file Lab 1.net	
	Router 1
IOS	12.4
Router Model	3640
Hostname Name	
Ethernet 0/0	**10.0.0.100 255.255.255.0**
Ethernet 0/1	
Serial 0/0	
Serial 0/1	

As in the lab before we will be configuring the interface of the router in global configuration. This time we will configure an ip address and then confirm the information using the "show ip interface brief". So first let's see what information show ip interface brief gives us on the router, before we configure it.

Router>**enable**
Router#
Router#**show ip interface brief**

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	unset	administratively down	down
BRI0/0	unassigned	YES	unset	administratively down	down
BRI0/0:1	unassigned	YES	unset	administratively down	down
BRI0/0:2	unassigned	YES	unset	administratively down	down
Ethernet0/1	unassigned	YES	unset	administratively down	down

The show ip interface brief gives us an opportunity to see all of the interfaces and some of the important information about them. As you can see no ip addresses have been assigned and all of the interfaces are disabled. So let's put an ip address on Ethernet0/0 and enable it
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.

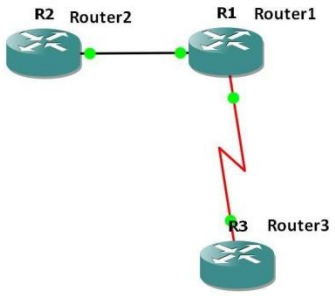
Router(config)**#interface ethernet 0/0**
Router(config-if)**#ip address 10.0.0.100 255.255.255.0**
Router(config-if)**#no shutdown**
00:03:58: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
From the output we know that the interface has been enabled, but we don't know if the ip address has been actually configured. So, let's go to privileged mode and see what the output from show ip interface brief tells us.

Router(config-if)**#end**
Router**#show ip interface brief**

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	10.0.0.100	YES	manual	up	up
BRI0/0	unassigned	YES	unset	administratively down	down
BRI0/0:1	unassigned	YES	unset	administratively down	down
BRI0/0:2	unassigned	YES	unset	administratively down	down
Ethernet0/1	unassigned	YES	unset	administratively down	down

As we can see only Ethernet0/0 has been affected and we can see that the ip address has been implemented as well as the interface being enabled. (The protocol may go down if you don't run this command within the "hold down" period, simply because the interface is not attached to another running interface.)

Lab 10 Description of lab: **IP (internet Protocol)**

GNS3 file Lab 10.net			
	Router 1	Router 2	Router 3
Router Model	3640	3640	3640
IOS	12.4	12.4	12.4
Hostname Name	Router1	Router2	Router3
Ethernet 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Ethernet 0/1			
Serial 0/0			**172.16.10.2 255.255.255.0**
Serial 1/0	172.16.10.1 255.255.255.0		

Since Router 1 will be connected to the other two routers we will configure it first

Router>**enable**
Router#conf t
Router(config)#**hostname Router1**
Router1(config)#**interface ethernet 0/0**
Router1(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**interface serial 1/0**
Router1(config-if)#**ip address 172.16.10.1 255.255.255.0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**end**
Since interface serial 1/0 is a serial port, we want to see if it is the DCE or the DTE end of the cable. When you physically connect the cables you can check the connections by sight, but since we are using an emulator this gives us a chance to use one of the "show" commands to find out about our connections.

Router1#show controllers serial 1/0

cable type: V.11 (X.21) DCE cable, received clockrate 2015232
As you can see the cable is a DCE cable (there is a long list of information that is outputted with this command, you will need to find this line to determine the cable termination type**). The DCE is the end of the cable that needs to set the clock rate. So, let's go back to the serial interface (interface serial 1/0) and add a clockrate of 64000**
Router#**conf t**
Router(config)#
Router1(config-if)#**interface serial 1/0**
Router1(config-if)#**clock rate 64000**
Router1(config-if)#**no shutdown**

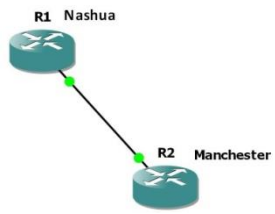
Now we'll configure Router 2
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config)#**int ethernet 0/0**
Router2(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router2(config-if)#**no shutdown**

Now Router 3
Router>**enable**
Router#conf t
Router3(config)#**hostname Router3**
Router3(config)#**interface serial 0/0**
Router3(config-if)#**ip address 172.16.10.2 255.255.255.0**
Router3(config-if)#**no shutdown**

Now we'll ping Router 2 and Router 3 from Router 1, to check our connectivity.
Router1#**ping 10.1.1.2**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5)

Router1#**ping 172.16.10.2**
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
Router1#

Lab 11 Description of lab: **ARP**

GNS3 file Lab 11.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name		
Ethernet 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0

In this activity we are first going to go to Global configuration mode and assign ip addresses to both routers 1 and router 2.

First Router 1

Router>**enable**
Router#**conf t**
Router(config)#**interface ethernet 0/0**
Router(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router(config-if)#**no shutdown**

Now Router 2

Router>**enable**
Router#**conf t**
Router(config)#**interface ethernet 0/0**
Router(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router(config-if)#**no shutdown**
Router(config-if)#

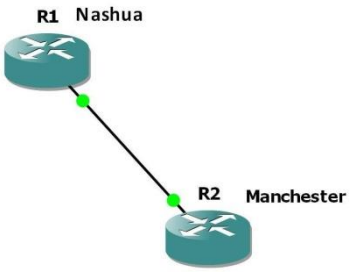
Now let's check to see if we have a connection between the two routers by pinging from router 2 to router 1

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 80 percent (4/5), round-trip min/avg/max = 116/136/168 ms

Now we'll view the ARP table. What addresses are there?

Router#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 10.1.1.2 - cc01.0aac.0000 ARPA Ethernet0/0
Internet 10.1.1.1 0 cc00.0aac.0000 ARPA Ethernet0/0

Lab 12 Description of lab: **Host Tables**

GNS3 file Lab 11.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Nashua	Manchester
Ethernet 0/0	195.42.36.10 255.255.255.240	195.42.36.12 255.255.255.240

In this activity we are going to go to Global configuration mode and give each router a hostname. Then we will configure the Ethernet port that we have each of them connected to.

Start with R1
Router>enable
Router#conf t
Router(config)#**hostname Nashua**
Nashua(config)#**interface ethernet 0/0**
Nashua(config-if)#**ip address 195.42.36.10 255.255.255.240**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#

Now R2
Router>enable
Router#conf t
Router(config)#hostname Manchester
Manchester(config)#**interface ethernet 0/0**
Manchester(config-if)#**ip address 195.42.36.12 255.255.255.240**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#
Manchester(config-if)#**exit**
Manchester(config)#

Now that we have both routers configured we're going to add the name "Nashua" along with its ip address to the host table of Router 2.

Manchester(config)#**ip host Nashua 195.42.36.10**
Manchester(config)#**exit**
Manchester#

Now let's test it by pinging "Nashua" instead of using the ip address.

Manchester#**ping Nashua**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 195.42.36.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 120/138/144 ms

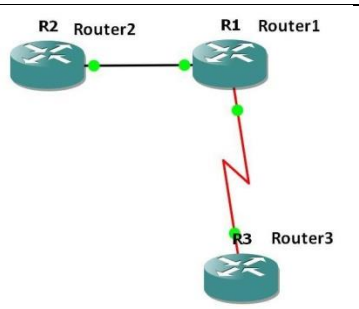
The ping command doesn't require case sensitive names, so you don't need to worry about typing the name with a capital letter. Now let's look at the host table and see what it shows us about "Nashua"
Manchester#**show hosts**

Host Port Flags Age Type Address(es)
Nashua None (perm, OK) 0 IP 195.42.36.10
Manchester#

Now go to the "Nashua" router and type the "show hosts" command and compare the output results.

Lab 13 Description of lab: **Static ip Routes**

Router 1

GNS3 file Lab 10.net			
	Router 1	Router 2	Router 3
Router Model	3640	3640	3640
IOS	12.4	12.4	12.4
Hostname Name	Router1	Router2	Router3
Ethernet 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Ethernet 0/1			
Serial 1/0	12.5.10.1 255.255.255.0		
Serial 0/0			12.5.10.2 255.255.255.0

Let's configure the three routers according to the chart.

First Router1

Router>enable
Router#conf t
Router(config)#**hostname Router1**
Router1(config)#**int e0/0**
Router1(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**int s1/0**
Router1(config-if)#**ip address 12.5.10.1 255.255.255.0**
Router1(config-if)#**clockrate 64000**
Router1(config-if)#**no shutdown**
Router1(config-if)#

Now Router2
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config)#**int e 0/0**
Router2(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router2(config-if)#**no shutdown**
Router2(config-if)#

NowRouter3
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router3**
Router3(config)#**int s0/0**
Router3(config-if)#**ip address 12.5.10.2 255.255.255.0**
Router3(config-if)#**no shutdown**
Router3(config-if)#
Router3(config-if)#**end**

Now from Router3 let's ping both interfaces on Router 1 and the interface on Router2
Router3#ping **12.5.10.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.5.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/64/84 ms
Router3#
Router3#**ping 10.1.1.2**

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)
Router3#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 0 percent (0/5)

So we can ping the serial interface on Router 1 that is attached to Router3, but not the interface on Router2 or even the interface on the other side of Router1. For those we'll need to give ip routes on router3 and router1.
So, first on Router3 let's provide it with the information to reach the other side of Router1. This information starts off with the network address and subnet of the network we want to reach and then we give it the interface that will lead to that network.
Router3#***conf t***
Router3(config)#**ip route 10.1.1.0 255.255.255.0 12.5.10.1**

As you can see we start with "ip route" then we typed in 10.1.1.0 (which is the network address) followed by the subnet mask. Then we typed in "12.5.10.1", which is the interface that leads to that network.
Now let's ping to the other side of Router1 and see if we get a result.
Router3#**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 = 48/69/96 ms
Router3#

Now let's try the interface of Router2

Router3#**ping 10.1.1.2**

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)
Router3#
Router 3 now knows about the network 10.1.1.0, as we can see on the "ip route" output.

Router3#**show ip route**

10.0.0.0/24 is subnetted, 1 subnets
S 10.1.1.0 [1/0] via 12.5.10.1
12.0.0.0/24 is subnetted, 1 subnets
C 12.5.10.0 is directly connected, Serial0/0
Router3#

However, Router2 does not know about the 12.5.10.0 network since it doesn't know where to send it; it drops the packet after it arrives.

So let's add the ip route on Router2 that is known. First we'll look at Router2 and its "show ip route" output to confirm what Router2 knows

```
Router2(config-if)#end
Router2#sh
Router2#show ip route

    10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0/0
Router2#
```

As you can see there is no information on the 12.5.10.0 network, so let's add it.

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

Now let's ping Router2 from Router3

```
Router3#ping 12.5.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.5.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/74/132 ms
Router3#
```

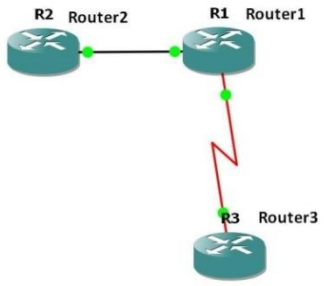
If we look at the "show ip route" on Router2 now, we should see the 12.5.10.0

```
Router2#sh ip route

    10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0/0
    12.0.0.0/24 is subnetted, 1 subnets
S       12.5.10.0 [1/0] via 10.1.1.1
Router2#
```

Lab 14

Description of lab: **Router RIP**

GNS3 file Lab 10.net			
	Router 1	Router 2	Router 3
Router Model	3640	3640	3640
IOS	12.4	12.4	12.4
Hostname Name	Router1	Router2	Router3
Ethernet 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Ethernet 0/1			
Serial 1/0	172.16.10.1 255.255.255.0		
Serial 0/0			172.16.10.2 255.255.255.0

In this lab we will be configuring Router Rip. So let's first configure the routers according to the above chart.

First Router1

Router>**enable**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#
Router1(config)#**int e0/0**
Router1(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**int s1/0**
Router1(config-if)#**ip address 172.16.10.1 255.255.255.0**
Router1(config-if)#**clockrate 64000**
Router1(config-if)#**no shutdown**
Router1(config-if)#

Next Router2
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config-if)#
Router2(config)#int **e0/0**
Router2(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router2(config-if)#**no shutdown**

Finally Router3
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router3**
Router3(config)#**int s0/0**
Router3(config-if)#**ip address 172.16.10.2 255.255.255.0**
Router3(config-if)#**no shutdown**
Router3(config-if)#

Now we'll ping the direct connections to make sure the ip addresses are correct
First Router3.
Router3#**ping 172.16.10.1**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/85/120 ms

Then Router2
Router2#**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 80 percent (4/5), round-trip min/avg/max = 120/137/144 ms
Router2#

Now we'll configure the routers for RIP
First Router1
Router1(config)#**router rip**
Router1(config-router)#**network 10.0.0.0**
Router1(config-router)#**network 172.16.0.0**
Router1(config-router)#

Now Router2
Router2(config)#**router rip**
Router2(config-router)#**network 10.0.0.0**
Router2(config-router)#

Then Router3
Router3(config)#**router rip**
Router3(config-router)#**network 172.16.0.0**

Now from Router3 let's ping the interface on Router2
Router3#**ping 10.1.1.2**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 156/208/264 ms
Router3#

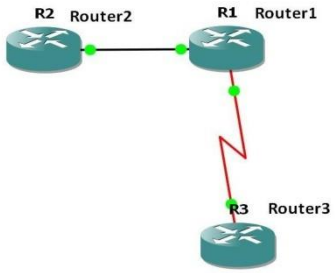
From Router2 lets ping the interface on Router3
Router2#**ping 172.16.10.2**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 192/232/264 ms
As you can see, setting up router RIP is much simpler than setting the direct ip. Routes on the network.
Let's go to Router 3 and look at the "show ip route**" output.**
Router3#**show ip route**

172.16.0.0/24 is subnetted, 1 subnets
C 172.16.10.0 is directly connected, Serial0/0
R 10.0.0.0/8 [120/1] via 172.16.10.1, 00:00:06, Serial0/0
Now let's look at the "show ip protocol***" output on Router1***
Router1#**show ip route**
172.16.0.0/24 is subnetted, 1 subnets
C 172.16.10.0 is directly connected, Serial1/0
10.0.0.0/24 is subnetted, 1 subnets
C 10.1.1.0 is directly connected, Ethernet0/0

As you can see Router RIP fills in the routing table for each router simply from the network information.

Lab 15 Description of lab: **EIGRP**

GNS3 file Lab 10.net			
	Router 1	Router 2	Router 3
Router Model	3640	3640	3640
IOS	12.4	12.4	12.4
Hostname Name	Router1	Router2	Router3
Ethernet 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Ethernet 0/1			
Serial 1/0	172.16.10.1 255.255.255.0		
Serial 0/0			172.16.10.2 255.255.255.0

EIGRP is an advanced distance-vector routing protocol, which allows routers within the EIGRP-routed network, to share a list of destination networks. For EIGRP we'll first set up the routers then we'll configure EIGRP as the routing protocol.

First Router1

Router>**enable**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#
Router1(config)#**int e0/0**
Router1(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**int s1/0**
Router1(config-if)#**ip address 172.16.10.1 255.255.255.0**
Router1(config-if)#**clockrate 64000**
Router1(config-if)#**no shutdown**
Router1(config-if)#

Next Router2
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config-if)#
Router2(config)#int **e0/0**
Router2(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router2(config-if)#**no shutdown**

Finally Router3
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router3**
Router3(config)#**int s0/0**
Router3(config-if)#**ip address 172.16.10.2 255.255.255.0**
Router3(config-if)#**no shutdown**
Router3(config-if)#

Now we'll ping the direct connections to make sure the ip addresses are correct
First Router3.
Router3#**ping 172.16.10.1**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/85/120 ms
Then Router2
Router2#**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 80 percent (4/5), round-trip min/avg/max = 120/137/144 ms
Router2#

So, if you've pinged the interfaces successfully, we'll move on to the configuring of EIGRP as the routing protocol. So we'll start with Router1 and we'll get back to Global Configuration mode and add both of the networks that Router1 is connected to. We'll also need to create an autonomous number for the system. The number itself, can be any number 1-65535, but that number needs to be consistent throughout the configuration (on all of the routers you want to belong to that "group").

Router1#**conf t**
Router1config)#
Router1(config)#**router eigrp20**

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

Now we'll setup Router2
Router2(config)#router eigrp 20
Router2(config-router)#network 10.0.0.0

Then Router3
Router3(config)#**router eigrp 20**
Router3(config-router)#**network 172.16.0.0**

Now on Router3 we'll ping the interface of Router2 to see if our protocol is working.
Router3#**ping 10.1.1.1**

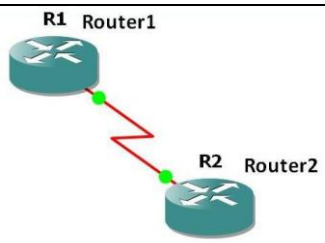
Let's look at the "show ip route" output onRouter3. You'll notice that the network "172.16.0.0/24**" is marked as "C" which is designated** "connected"**whereas the network "**10.1.1.0/24**" is marked as "**D" **which is designated** "EIGRP"**This shows us which route**
Router3#**show ip route**
*Mar 1 00:26:50.991: %SYS-5-CONFIG_I: Configured from console by consoleoute
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 1 subnets
C 172.16.10.0 is directly connected, Serial0/0
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D 10.1.1.0/24 [90/2195456] via 172.16.10.1, 00:00:00, Serial0/0
D 10.0.0.0/8 [90/2195456] via 172.16.10.1, 00:00:00, Serial0/0

So, as you can see EIGRP fills in the destination location to the routers on the networks that are not directly attached to it. Also for every destination, a successor and a feasible successor are identified and stored in the table if they exist.

Lab 16 Description of lab: **PPP Encapsulation w/ CHAP**

GNS3 file Lab 4.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Router1	Router2
Ethernet 0/0		
Ethernet 0/1		
Serial 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0
Serial 1/0		

PPP encapsulation is commonly used to establish a direct connection between two clients. We use CHAP (Challenge-Handshake Authentication Protocol) to authenticate the connection, which gives added security to the connection. First we'll setup Router1

Router>
Router>**enable**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#**username Router2 password JohnMason**
Router1(config)#**int s0/0**
Router1(config-if)#**ip address 10.1.1.1 255.255.255.0**
Router1(config-if)#**encapsulation ppp**
Router1(config-if)#**ppp authentication chap**
Router1(config-if)#**no shutdown**
Router1(config-if)#**no keepalive**

Next we'll setup Router2. However we won't put the user name or password in to show that CHAP is protecting Router1.

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

Router2(config)#**int s0/0**
Router2(config-if)#**ip address 10.1.1.2 255.255.255.0**
Router2(config-if)#**encapsulation ppp**
Router2(config-if)#**ppp authentication chap**
Router2(config-if)#**no shutdown**
Router2(config-if)#**no keepalive**
Router2(config-if)#**end**

Now ping Router1.
Router2#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 0 percent (0/5)

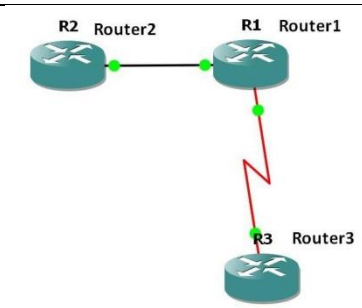
Now let's put the user name and password in Global Configuration Mode.
Router2#**conf t**
Router2(config)#**username Router1 password JohnMason**
Router2(config)#
Router2(config)#**end**

Now when we ping we can reach Router1 since CHAP recognizes the username and password.

Router2#**ping 10.1.1.1**
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 40/54/72 ms

If you have any problems use "show interfaces serial 0/0" to see if there are any problems with the PPP. Pay attention to the "lcp state" (Encapsulation PPP, LCP Open). If there are problems use the "debug ppp authentication"

Lab 17 Description of lab: **Using Traceroute to test connectivity.**

GNS3 file Lab 10.net			
	Router 1	Router 2	Router 3
Router Model	3640	3640	3640
IOS	12.4	12.4	12.4
Hostname Name	Router1	Router2	Router3
Ethernet 0/0	192.168.1.1 255.255.255.0	192.168.1.2 255.255.255.0	
Ethernet 0/1			
Serial 1/0	192.168.2.1 255.255.255.0		
Serial 0/0			192.168.2.2 255.255.255.0

First let's set the ip addresses on all three routers. We'll start with Router 1.

Router>**en**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#**int e0/0**
Router1(config-if)#**ip address 192.168.1.1 255.255.255.0**
Router1(config-if)#**no shut**
Router1(config)#**int s1/0**
Router1(config-if)#**ip address 192.168.2.1 255.255.255.0**
Router1(config-if)#**clockrate 64000**
Router1(config-if)#**no shut**

Now Router2

Router>
Router>**en**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config)#**int e0/0**
Router2(config-if)#**ip address 192.168.1.2 255.255.255.0**
Router2(config-if)#**no shut**

Then Router3
Router>**en**
Router#**conf t**
Router(config)#**hostname Router3**
Router3(config)#**int s0/0**
Router3(config-if)#**ip address 192.168.2.2 255.255.255.0**
Router3(config-if)#**no shut**

Let's ping Router3 from Router2
Router2#**ping 192.168.2.2**
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
We cannot ping Router 3 yet, but let's see what the "traceroute" looks like
Router2#**traceroute 192.168.2.2**

Type escape sequence to abort.
Tracing the route to 192.168.2.2

1 * * *
2 * * *
3 * * *
4 * * *
5 * * *
6 * * *
7 * * *
8 * * *

Since we haven't set up the routes on Router1 and Router2, we can't see Router3. So, we'll setup Router Rip. First on Router1 and Router2
Router1(config-if)#**exit**
Router1(config)#**router rip**
Router1(config-router)#**network 192.168.1.0**
Router1(config-router)#**network 192.168.2.0**

Now we'll setup Router Rip on Router2
Router2(config-if)#**exit**
Router2(config)#**router rip**
Router2(config-router)#**network 192.168.1.0**
Router2(config-router)#**end**

Now let's ping Router3 from Router2 again and then check "traceroute**"**
Router2#**ping 192.168.2.2**
*Mar 1 00:04:31.639: %SYS-5-CONFIG_I: Configured from console by console

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Router2#traceroute 192.168.2.2

Type escape sequence to abort.
Tracing the route to 192.168.2.2

1 192.168.1.1 148 msec 132 msec 168 msec
2 * * *
3 * * *
4 * * *
5 * * *
6 * * *
7 * * *

This time we can see the route to Router3, but since we haven't configured Router Rip on Router3, Router3 won't respond. So, now we'll setup Router Rip on Router3
Router3(config-if)#exit
Router3(config)#router rip
Router3(config-router)#network 192.168.2.0
Router3(config-router)#

Let's ping Router2 and Router3 from Router1
Router1#ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/68/96 ms

Now we'll ping Router3from Router2

Router2#ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 216/236/268 ms

Now we'll use "traceroute**" on Router2 to see where the packet went.**

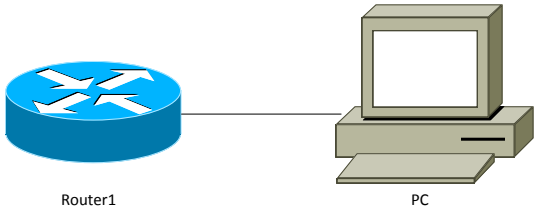
Router2#traceroute 192.168.2.2

Type escape sequence to abort.
Tracing the route to 192.168.2.2

1 192.168.1.1 124 msec 156 msec 140 msec
2 192.168.2.2 216 msec 228 msec *
Router2#

Now we have a complete ping and as we can see from "traceroute" the packet went to both Router1 and then to Router2

Lab 18 Description of lab: **Backing up your Routers running-config**

		
	Router 1	
Model	3640	Model
IOS	12.4	IOS
Hostname Name	Nashua	Hostname Name
Ethernet 0/0	10.1.1.2 255.255.255.0	Ethernet 0/0

For this lab we use a router, a "loopback computer" and "WinAgents TFTP Server".

First let's configure the router

Router>**enable**
Router#**conf t**
Router(config)#**hostname Nashua**
Nashua(config)#**int e0/0**
Nashua(config-if)#**ip address 10.1.1.2 255.255.255.0**
Nashua(config-if)#**no shutdown**
Router(config-if)#**no keepalive**
Nashua(config-if)#**end**
Nashua#

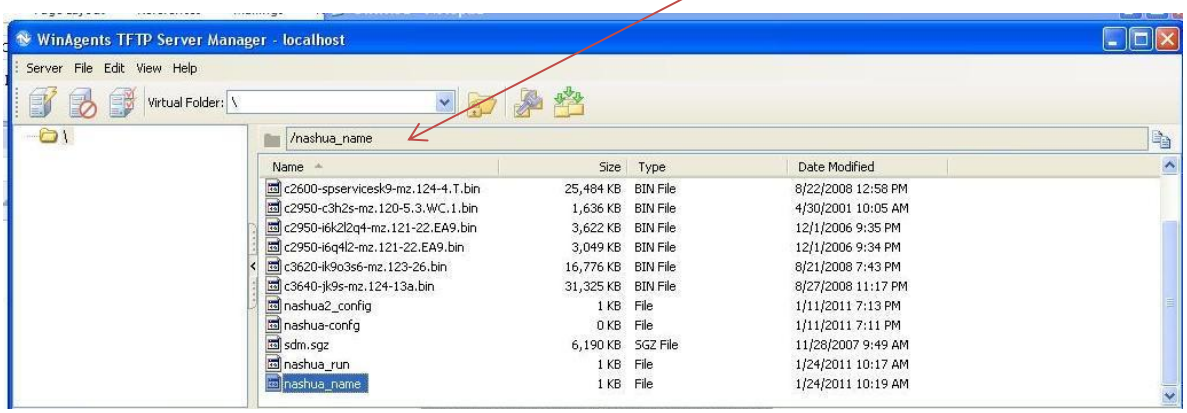
Now make sure the Ethernet interface of the computer is configured from Control Panel/Network Connections right click on the "Local Area connection" that the Ethernet cable is plugged into and select "properties", then highlight the "internet protocol [TCP/IP] and select properties and select the "fill in the following ip address" with 10.1.1.9 and a subnet of 255.255.255.0. Open the command prompt and ping the router to make sure you have connectivity. Open the program "Winagents TFTP Server Manager" and make sure you are connected to the local server.
Then go to the router and ping the computer to make sure you can connect to it. Then copy the running-config to the TFTP server.

Nashua#**copy running-config tftp:**
Address or name of remote host []? **10.1.1.9**
Destination filename [nashua-confg]? **nashua_name**
540 bytes copied in 2.556 secs (211 bytes/sec)

You should be able to see the file in the lower pane of the upper middle window of "Winagents TFTP Server Manager". Select the file so that it appears in the address bar above that window. This will make it accessible to the router. Next change the name of the router from "Nashua" to "badrouter", so that we can see that the old running-config has been loaded.

Nashua#
Nashua# **conf t**
Nashua(config)#**hostname badrouter**
badrouter(config)#**end**
badrouter#

Now we'll copy the running-config back to the router. Make sure that the file is highlighted in "Winagents TFTP Server Manager".

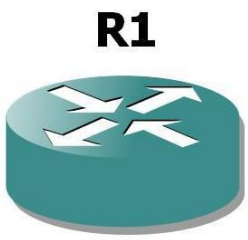


badrouter#**copy tftp running-config**
Address or name of remote host []? **10.1.1.9**
Source filename []? **nashua_name**
Destination filename [running-config]?
Accessing tftp://10.1.1.9/nashua_name...
Loading nashua_name from 10.1.1.9 (via FastEthernet0/0): !
[OK - 507 bytes]
507 bytes copied in 9.508 secs (53 bytes/sec)
Nashua#
Nashua#

So as you can see the router has reverted back to the name that you had given it before "Nashua". This shows that you have copied the previous running-config

Lab 19

Description of lab: **Copying and pasting commands**

GNS3 file Lab 1.net	
	Router 1
Model	3640
IOS	12.4
Hostname Name	Router1
Ethernet 0/0	1.1.1.1 255.255.255.0

Open "Notepad" on your computer and type the following lines. Then select the lines and "copy" the following lines onto the "clipboard".

```
en
conf t
hostname Router1
interface Ethernet0/0
ip address 1.1.1.1 255.255.255.0
no shutdown
exit
exit
exit
exit
```

Take you mouse and right click in the "Putty" window at the "Router1>" prompt. This will "paste" those commands to the router interface.

The output should look like this:

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname Router1
```

```
Router1(config)#interface Ethernet0/0
Router1(config-if)#ip address 1.1.1.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#exit
Router1#exit

Router1 con0 is now available


Press RETURN to get started.
```

After you press enter you should see the prompt:

Router1>

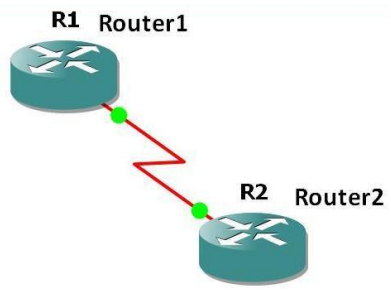
Get into Privileged EXEC Mode and type "show ip interface brief"

```
Router1>enable
Router1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            1.1.1.1         YES manual up                    up
Ethernet0/1            unassigned      YES unset  administratively down down
Ethernet0/2            unassigned      YES unset  administratively down down
Ethernet0/3            unassigned      YES unset  administratively down down
Router1#
```

As you can see the Ethernet port 0/0 has been assigned the address of 1.1.1.1 and is enabled.

Lab 20

Description of lab*:* **Frame Relay**

Lab 4.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Nashua	Manchester
Serial 0/0	No ip address	No ip address
Serial 0/0.1	192.168.12.2 255.255.255.252	192.168.12.1 255.255.255.252

First we'll set up the two routers.
We'll start with Router 1 (Nashua)
Router>
Router>**enable**
Router#**conf t**
Router(config)#**hostname Nashua**
Nashua(config)#

Then Router 2 (Manchester)

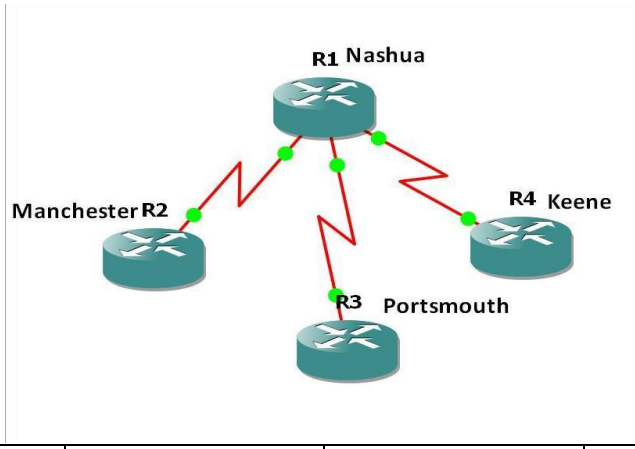
Router>
Router>**enable**
Router#**conf t**
Router(config)#**hostname Manchester**

Now let's set the interfaces. Since we are going to run the frame relay through a sub interface, we will set Serial0/0 with "no ip address" and "no keepalive" so as to keep the connection open.
First "Nashua"
Nashua(config)#**interface Serial0/0**

Nashua(config-if)#**no ip address**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Then Manchester
Manchester(config)#**interface Serial0/0**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**

Now let's set the sub interfaces. These will have the ip addresses and the frame relay information, after we set that up we'll re-enter the Serial 0/0 interface and set "no shutdown" to enable the interface.
First "Nashua"
Nashua(config-if)#**interface Serial0/0.1 point-to-point**
Nashua(config-subif)#**ip address 192.168.12.2 255.255.255.252**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 102**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config)#**interface Serial0/0**
Nashua(config-if)#**no shutdown**
Then "Manchester"
Manchester(config-if)#**interface Serial0/0.1 point-to-point**
Manchester(config-subif)#**ip address 192.168.12.1 255.255.255.252**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 102**
Manchester(config-fr-dlci)#**exit**
Manchester(config-subif)#**exit**
Manchester(config)#**interface Serial0/0**
Manchester(config-if)#**no shutdown**
Now let's use the "show frame-relay map" to see information on the Frame relay we have set up.
Nashua(config-if)#end
Nashua#
Nashua#show frame-relay map
Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
Nashua#

Lab21 Description of lab: **Frame Relay Hub and Spoke Technology**

GNS3 file Lab 21.net	

	Router 1	Router 2	Router 3	Router 4
Router Model	3640	3640	3640	3640
IOS	12.4	12.4	12.4	12.4
Hostname Name	Nashua	Manchester	Portsmouth	Keene
Serial 0/0	10.1.1.1	10.1.1.2	10.1.2.2	10.1.3.2
Serial 0/1	10.1.2.1			
Serial 0/2	10.1.3.1			
Serial 0/1				

In this lab we will be configuring 4 routers 1 of which will be acting as the hub for the other three. This kind of scenario would be similar to what a corporate headquarters might have with 3 of its branch offices. We'll be configuring each of the "branch" routers to the "headquarters" router using frame relay and the sub interface scenario, like the last lab. However, since we will be using three interfaces on the "Nashua" router we'll have to match the interfaces on the other routers so that the sub interfaces match.

Let's configure the "Nashua" router with the information needed to interface with the "Manchester" router first.

Router>**enable**
Router#**conf t**
Router(config)#**hostname Nashua**
Nashua(config)#
Nashua(config)# **interface Serial0/0**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#
Nashua(config-if)#**interface Serial0/0.1 point-to-point**
Nashua(config-subif)#**ip address 10.1.1.1 255.255.255.0**

Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 102**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config)#

Now let's setup the "Manchester" router
Router>**enable**
Router#**conf t**
Router(config)#**hostname Manchester**
Manchester(config)#**interface Serial0/0**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#
Manchester(config-if)#**interface Serial0/0.1 point-to-point**
Manchester(config-subif)#**ip address 10.1.1.2 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 102**
Manchester(config-fr-dlci)#**exit**
Manchester(config-subif)#**exit**
Manchester(config)#

Notice that the dlci number is the same on both of these as well as the sub interface. Now we'll go back to the "Nashua" router and configure the next interface for the "Portsmouth" router
Nashua(config-if)#**interface Serial0/1**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#
Nashua(config-if)#**interface Serial0/1.2 point-to-point**
Nashua(config-subif)#**ip address 10.1.2.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 103**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config)#

So, you can see that interface that the Nashua router is using to connect to "Portsmouth" is "Serial0/1". This means that if we want the sub interface numbers to be consistent we will have to hook up to the same interface on the "Portsmouth" router.
Router>**enable**

Router#**conf t**
Router(config)#**hostname Portsmouth**
Portsmouth(config)#**interface Serial0/1**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**no shutdown**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#
Portsmouth(config-if)#**interface Serial0/1.2 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.2.2 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 103**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Portsmouth(config)#

Now we'll go back to the "Nashua" router and configure the next interface for the "Keene" router
Nashua(config-if)#**interface Serial0/2**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#
Nashua(config-if)#**interface Serial0/2.3 point-to-point**
Nashua(config-subif)#**ip address 10.1.3.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 104**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#exit
Nashua(config)#

Again we've attached the cable to the interface on the Keene router to match the same interface that the Nashua router is using for that interface, so that our sub interface numbers are consistent.
Router>**enable**
Router#**conf t**
Router(config)#**hostname Keene**
Keene(config)#**interface Serial0/2**
Keene(config-if)#**no ip address**

Keene(config-if)#**no shutdown**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#
Keene(config-if)#**interface Serial0/2.3 point-to-point**
Keene(config-subif)#**ip address 10.1.3.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 104**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#

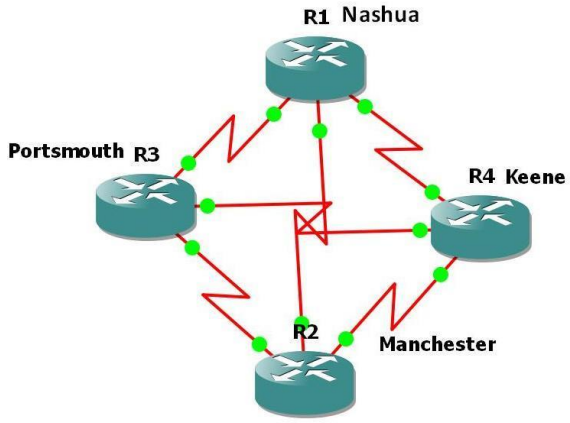
So, we'll ping all three of the routers from Nashua to see that we've set them up correctly.
Nashua#ping 10.1.1.2
Nashua#ping 10.1.2.2
Nashua#ping 10.1.3.2

The "show frame-relay map**" will show us all three of the frame relays running on Nashua.**
Nashua#**show frame-relay map**
Serial0/0.1 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
Serial0/1.2 (up): point-to-point dlci, dlci 103(0x67,0x1870), broadcast
Serial0/2.3 (up): point-to-point dlci, dlci 104(0x68,0x1880), broadcast

The "show ip route**" gives us the details on which serial port is connected to which ip address.**

Nashua#**show ip route**
10.0.0.0/24 is subnetted, 3 subnets
C 10.1.3.0 is directly connected, Serial0/2.3
C 10.1.2.0 is directly connected, Serial0/1.2
C 10.1.1.0 is directly connected, Serial0/0.1
We can also look at the "show frame-relay pvc" to see information on Permanent Virtual Circuit mapping for the router.

Lab22 Description of lab: **Frame Relay Full Mesh Topology**



	Router 1	Router 2	Router 3	Router 4
Router Model	3640	3640	3640	3640
IOS	12.4	12.4	12.4	12.4
Hostname Name	Nashua	Manchester	Portsmouth	Keene
Serial 0/0				
Serial 0/1				
Serial 0/2				

First configure router 1 to be "Nashua" and set the ip routes and sub interfaces for the "Manchester" router.

Router>**enable**
Router#**conf t**
Router(config)#**hostname Nashua**
Nashua(config)#
Nashua(config)#**interface Serial0/0**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#**interface Serial0/0.1 point-to-point**
Nashua(config-subif)#**ip address 10.1.1.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 102**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config)#

Then set the ip routes and sub interfaces to the "Portsmouth" router.

Nashua(config-if)#**interface Serial0/1**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#**interface Serial0/1.2 point-to-point**
Nashua(config-subif)#**ip address 10.1.2.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 103**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config)#

Finally set the ip routes and sub interfaces to the "Keene" router.

Nashua(config-if)#**interface Serial0/2**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#**interface Serial0/2.3 point-to-point**
Nashua(config-subif)#**ip address 10.1.3.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 104**
Nashua(config-fr-dlci)#**exit**
Nashua(config-subif)#**exit**
Nashua(config-if)#

Next we configure router 2 to be the "Manchester" router and set the ip routes and sub interfaces to the "Nashua" router

Router>**enable**
Router#**conf t**
Router(config)#**hostname Manchester**
Manchester(config)#**interface Serial0/0**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#**interface Serial0/0.1 point-to-point**
Manchester(config-subif)#**ip address 10.1.1.2 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 102**
Manchester(config-fr-dlci)# **exit**
Manchester(config-subif)# **exit**

Then set the ip routes and sub interfaces to the "Portsmouth" router
Manchester(config-if)#**interface Serial0/2**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#**interface Serial0/2.4 point-to-point**
Manchester(config-subif)#**ip address 10.1.4.1 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 105**
Manchester(config-fr-dlci)# **exit**
Manchester(config-subif)# **exit**

Finally set the ip routes and sub interfaces to the "Keene" router.
Manchester(config-if)#**interface Serial0/1**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#**interface Serial0/1.5 point-to-point**
Manchester(config-subif)#**ip address 10.1.5.1 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 106**
Manchester(config-fr-dlci)#**exit**
Manchester(config-subif)#**exit**

Then we configure router 3 to be the "Portsmouth" router and set the ip routes and sub interfaces to the "Nashua" router
Router>**enable**
Router#**conf t**
Router(config)#**hostname Portsmouth**
Portsmouth(config)#**interface Serial0/1**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**no shutdown**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/1.2 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.2.2 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 103**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Now we set the ip routes and sub interfaces to the "Keene" router.

Portsmouth(config-if)#**interface Serial0/0**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**no shut**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/0.6 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.6.1 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 107**
Portsmouth(config-fr-dlci)#exit
Portsmouth(config-subif)#exit
Portsmouth(config)#

Then set the ip routes and sub interfaces to the "Manchester" router.
Portsmouth(config-if)#**interface Serial0/2**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**no shutdown**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/2.4 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.4.2 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 105**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Portsmouth(config)#

Now we'll configure the last router (router 4) to be the "Keene" router and set the ip routes and sub interfaces to the "Nashua" router
Router>**enable**
Router#**conf t**
Router(config)#**hostname Keene**
Keene(config)#**interface Serial0/2**
Keene(config-if)#**no ip address**
Keene(config-if)#**no shutdown**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/2.3 point-to-point**
Keene(config-subif)#**ip address 10.1.3.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 104**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**

Next set the ip routes and sub interfaces to the "Manchester" router.

Keene(config-if)#**interface Serial0/1**
Keene(config-if)#**no ip address**
Keene(config-if)#**no shutdown**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/1.5 point-to-point**
Keene(config-subif)#**ip address 10.1.5.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 106**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#

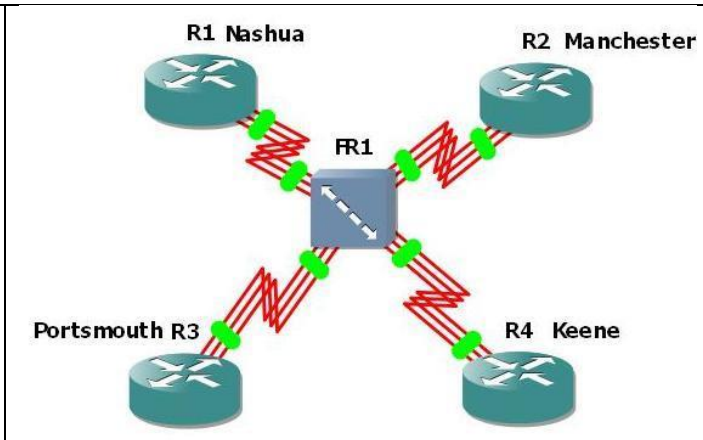
Finally set the ip routes and sub interfaces to the "Portsmouth" router

Keene(config-if)#**interface Serial0/0**
Keene(config-if)**#no ip address**
Keene(config-if)#**no shutdown**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/0.6 point-to-point**
Keene(config-subif)#**ip address 10.1.6.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 107**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#
Keene(config-if)#end

Now let's ping the routers and see if we have connections all the way around

Nashua#**ping 10.1.1.2**
Nashua#**ping 10.1.2.2**
Nashua#**ping10.1.3.2**
Manchester#**ping 10.1.1.1**
Manchester#**ping10.1.4.2**
Manchester#**ping10.1.5.2**
Portsmouth#**ping 10.1.2.1**
Portsmouth#**ping10.1.4.1**
Portsmouth#**ping 10.1.6.2**
Keene(config)#**ping10.1.3.1**
Keene(config)#**ping10.1.5.1**
Keene(config)# **ping10.1.6.1**

Lab23Description of lab: Frame Relay Full Mesh Topology with Switch Relay



	Router 1	Router 2	Router 3	Router 4
Router Model	3640	3640	3640	3640
IOS	12.4	12.4	12.4	12.4
Hostname Name	Nashua	Manchester	Portsmouth	Keene
Serial 0/0	**10.1.1.1**			
Serial 0/1				
Serial 0/2				

In this lab we will be setting up a Frame-Relay that looks the same as the one in Lab 22, however this time we will be using a "Frame-Relay switch. Note that there is no IP address on the port, but instead on the sub interface. We've also used the "no keepalive" command so that the port won't shut down. First we'll configure Router 1 as "Nashua" and Serial 0/0 to "Manchester".

Router>**enable**
Router#**conf t**
Router(config)#**hostname Nashua**
Nashua(config)#
Nashua(config)#**interface Serial0/0**
Nashua(config-if)#**no ip address**
Nashua(config-if)#**no shutdown**
Nashua(config-if)#**encapsulation frame-relay**
Nashua(config-if)#**no keepalive**
Nashua(config-if)#**interface Serial0/0.1 point-to-point**
Nashua(config-subif)#**ip address 10.1.1.1 255.255.255.0**
Nashua(config-subif)#**no shutdown**
Nashua(config-subif)#**frame-relay interface-dlci 102**

Nashua(config-fr-dlci)**#exit**
Nashua(config-subif)**#exit**
Nashua(config)#

Next we'll configure Serial 0/1 to "Portsmouth". You will notice that the "frame-relay interface-dlci" numbers are 102, 103 and 104 on the outgoing ports. Later on you will see that the corresponding ports on the other routers will have the transverse numbers (201. 301 and 401) on their ends that are connected to these ports.
Nashua(config-if)**#interface Serial0/1**
Nashua(config-if)**#no ip address**
Nashua(config-if)**#no shutdown**
Nashua(config-if)**#encapsulation frame-relay**
Nashua(config-if)**#no keepalive**
Nashua(config-if)**#interface Serial0/1.2 point-to-point**
Nashua(config-subif)**#ip address 10.1.2.1 255.255.255.0**
Nashua(config-subif)**#no shutdown**
Nashua(config-subif)**#frame-relay interface-dlci 103**
Nashua(config-fr-dlci)#exit
Nashua(config-subif)#exit
Nashua(config)#

Finally we'll configure Serial 0/2 to "Keene".
Nashua(config-if)**#interface Serial0/2**
Nashua(config-if)**#no ip address**
Nashua(config-if)**#no shutdown**
Nashua(config-if)**#encapsulation frame-relay**
Nashua(config-if)**#no keepalive**
Nashua(config-if)**#interface Serial0/2.3 point-to-point**
Nashua(config-subif)**#ip address 10.1.3.1 255.255.255.0**
Nashua(config-subif)**#no shutdown**
Nashua(config-subif)**#frame-relay interface-dlci 104**
Nashua(config-fr-dlci)**#exit**
Nashua(config-subif)**#exit**
Nashua(config)#

Now, we'll configure Router 2 as "Manchester" and Serial 0/0 back to "Nashua". Again notice the "frame-relay interface-dlci" number. It is 201, which indicates that it is the receiving end of "frame-relay interface-dlci 102" from the Nashua router. You can set these numbers a different way if you'd like, the significance is for you to decide.
Router>**enable**
Router#**conf t**
Router(config)#**hostname Manchester**
Manchester(config)#**interface Serial0/0**
Manchester(config-if)#**no ip address**

Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#**interface Serial0/0.1 point-to-point**
Manchester(config-subif)#**ip address 10.1.1.2 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 201**
Manchester(config-fr-dlci)#**exit**
Manchester(config-subif)#**exit**
Manchester(config)#

Next we'll configure Serial 0/2 to Portsmouth.
Manchester(config-if)#**interface Serial0/2**
Manchester(config-if)#**no ip address**
Manchester(config-if)#**no shutdown**
Manchester(config-if)#**encapsulation frame-relay**
Manchester(config-if)#**no keepalive**
Manchester(config-if)#**interface Serial0/2.4 point-to-point**
Manchester(config-subif)#**ip address 10.1.4.1 255.255.255.0**
Manchester(config-subif)#**no shutdown**
Manchester(config-subif)#**frame-relay interface-dlci 105**
Manchester(config-fr-dlci)#**exit**
Manchester(config-subif)#**exit**
Manchester(config)#interface Serial0/2

Finally we'll configure Serial 0/1 to Keene
Manchester(config-if)#interface Serial0/1
Manchester(config-if)#no ip address
Manchester(config-if)#encapsulation frame-relay
Manchester(config-if)#no keepalive
Manchester(config-if)#interface Serial0/1.5 point-to-point
Manchester(config-subif)#ip address 10.1.5.1 255.255.255.0
Manchester(config-subif)#no shutdown
Manchester(config-subif)#frame-relay interface-dlci 106
Manchester(config-fr-dlci)#exit
Manchester(config-subif)#exit
Manchester(config)#interface Serial0/1
Manchester(config-if)#no shutdown

Now, we'll configure Router 3 as "Portsmouth" and Serial 0/1 to Nashua.

Router>**enable**
Router#**conf t**
Router(config)#**hostname Portsmouth**
Portsmouth(config)#**interface Serial0/1**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/1.2 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.2.2 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 301**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Portsmouth(config)#**interface Serial0/1**
Portsmouth(config-if)#**no shutdown**

Next we'll configure Serial 0/0 to Keene

Portsmouth(config-if)#**interface Serial0/0**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/0.6 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.6.1 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 107**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Portsmouth(config)#**interface Serial0/0**
Portsmouth(config-if)#**no shutdown**

Finally we'll configure Serial 0/2 to Manchester

Portsmouth(config-if)#**interface Serial0/2**
Portsmouth(config-if)#**no ip address**
Portsmouth(config-if)#**encapsulation frame-relay**
Portsmouth(config-if)#**no keepalive**
Portsmouth(config-if)#**interface Serial0/2.4 point-to-point**
Portsmouth(config-subif)#**ip address 10.1.4.2 255.255.255.0**
Portsmouth(config-subif)#**no shutdown**
Portsmouth(config-subif)#**frame-relay interface-dlci 501**
Portsmouth(config-fr-dlci)#**exit**
Portsmouth(config-subif)#**exit**
Portsmouth(config)#**interface Serial0/2**
Portsmouth(config-if)#**no shutdown**

Finally we'll configure Router as "Keene" and Serial 0/2 to Nashua.

Router>**enable**
Router#**conf t**
Router(config)#**hostname Keene**
Keene(config)#**interface Serial0/2**
Keene(config-if)#**no ip address**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/2.3 point-to-point**
Keene(config-subif)#**ip address 10.1.3.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 401**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#**interface Serial0/2**
Keene(config-if)#**no shutdown**

Next we'll configure Serial 0/1 to Manchester

Keene(config-if)#**interface Serial0/1**
Keene(config-if)#**no ip address**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/1.5 point-to-point**
Keene(config-subif)#**ip address 10.1.5.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 601**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#**interface Serial0/1**
Keene(config-if)#**no shutdown**

Finally we'll configure Serial 0/0 to Portsmouth

Keene(config-if)#**interface Serial0/0**
Keene(config-if)#**no ip address**
Keene(config-if)#**encapsulation frame-relay**
Keene(config-if)#**no keepalive**
Keene(config-if)#**interface Serial0/0.6 point-to-point**
Keene(config-subif)#**ip address 10.1.6.2 255.255.255.0**
Keene(config-subif)#**no shutdown**
Keene(config-subif)#**frame-relay interface-dlci 701**
Keene(config-fr-dlci)#**exit**
Keene(config-subif)#**exit**
Keene(config)#interface Serial0/0
Keene(config-if)#no shutdown

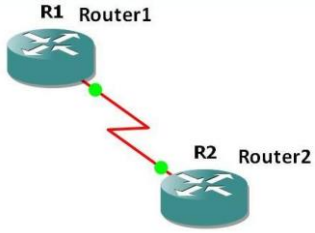
Go to Nashua
Nashua#**ping 10.1.1.2**
Nashua#**ping 10.1.2.2**
Nashua#**ping 10.1.3.2**

Go to Manchester
Manchester#**ping 10.1.1.1**
Manchester#**Ping 10.1.4.2**
Manchester#**ping 10.1.5.2**

Go to Portsmouth
Portsmouth#**ping 10.1.2.1**
Portsmouth#**ping 10.1.6.2**
Portsmouth#**ping 10.1.4.1**

Go to Keene
Keene#**ping 10.1.3.1**
Keene#**ping 10.1.5.1**
Keene#**ping 10.1.6.1**

Lab24 Description of lab: **configure 2 routers with an ACL**

Lab 4.net		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Router1	Router2
Ethernet 0/0		
Ethernet 0/1		
Serial 0/0	**172.16.1.1 255.255.255.192**	**172.16.1.2 255.255.255.192**
Serial 0/1		

Router1
Router>**en**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#**int s 0/0**
Router1(config-if)#**clock rate 64000**
Router1(config-if)#**ip address 172.16.1.1 255.255.255.192**
Router1(config-if)#**no shutdown**
Router1(config-if)#**end**
Switch to Router2
Router>**en**
Router#**conf t**
Router(config)#**hostname Router2**
Router2(config)#**int s 0/0**
Router2(config-if)#**ip address 172.16.1.2 255.255.255.192**
Router2(config-if)#**no shutdown**
Router2(config-if)#**end**
Router2#**ping 172.16.1.1**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/77/96 ms

Switch to Router1
Router1#**ping 172.16.1.2**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/72/96 ms
Router1#
Router1#**conf t**
Router1(config)#**ip route 0.0.0.0 0.0.0.0 s0/0 172.16.1.2**
Router1(config)#**end**
Router1#

Switch to Router2
Router2#**conf t**
Router2(config)#**ip route 0.0.0.0 0.0.0.0 s0/0 172.16.1.1**
Router2(config)#**interface loopback 10**
Router2(config-if)#
Router2(config-if)#**ip address 10.10.10.3 255.255.255.128**
Router2(config-if)#**exit**
Router2(config)#**int loopback 20**
Router2(config-if)#**ip address 10.20.20.3 255.255.255.240**
Router2(config-if)#**exit**
Router2(config)#**int loopback 30**
Router2(config-if)#**ip address 10.30.30.3 255.255.255.248**
Router2(config-if)#**exit**
Router2(config)#**exit**
Router2#**ping 172.16.1.1**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/66/96 ms

Router2#**ping 172.16.1.1 source loopback 10**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.10.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/58/96 ms

Router2#**ping 172.16.1.1 source loopback 20**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.20.20.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/58/96 ms

Router2#**ping 172.16.1.1 source loopback 30**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.30.30.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/57/96 ms

NOTE: The wildcard masks used in ACLs are configured in the same way as those for EIGRP and OSPF. To determine the wildcard mask, you simply subtract the network mask for the network on which you want to match with the ACL from the Broadcast mask. This concept is illustrated in the subtraction table shown below:

Broadcast Mask			
255	255	255	255
[minus] Subnet Mask			
255	255	255	128
[equals] Wildcard Mask			
0	0	0	127

In our example, the subnet mask of the 10.10.10.0/25 subnet is 255.255.255.128. If this is subtracted from the Broadcast mask of 255.255.255.255 the result is 0.0.0.127, which is the wildcard mask we use to use in the ACL match for this subnet. Using the same concept, the subnet mask of the10.20.20.0/28 subnet is 255.255.255.240. If we use the above table to determine the wildcard mask, we would get the following:

Broadcast Mask			
255	255	255	255
[minus] Subnet Mask			
255	255	255	240
[equals] Wildcard Mask			
0	0	0	15

And finally, the subnet mask of the 10.30.30.0/29 subnet is 255.255.255.248. If we used the same table to get the wildcard mask, we would end up with the following:

Broadcast Mask			
255	255	255	255
[minus] Subnet Mask			
255	255	255	248
[equals] Wildcard Mask			
0	0	0	7

It is extremely important to practice creating wildcards for ACLs. Take time out to practice these until you are extremely comfortable with them. ACLs are a very important part of the CCNA certification and in the real world. The access-list [number] remark [description] is a good idea, so that you'll know what each line of the ACL is referring to. You won't see the description in the "show access-lists" output but you will see it in the "show running-config" output.

Task 3

Switch to Router1

Router1>**en**
Router1#**conf t**
Router1(config)#**access-list 10 remark 'permit From Router2 loopback10'**
Router1(config)#**access-list 10 permit 10.10.10.0 0.0.0.127**
Router1(config)#**access-list 10 remark 'deny from Router2 loopback20'**
Router1(config)#**access-list 10 deny 10.20.20.0 0.0.0.15**
Router1(config)#**access-list 10 remark 'permit From Router2 loopback30'**
Router1(config)#**access-list 10 permit 10.30.30.0 0.0.0.7**
Router1(config)#**int s0/0**
Router1(config-if)#**ip access-group 10 in**
Router1(config-if)#**end**
Router1#**show ip access-lists**
Standard IP access list 10
  10 permit 10.10.10.0, wildcard bits 0.0.0.127
  20 deny 10.20.20.0, wildcard bits 0.0.0.15
  30 permit 10.30.30.0, wildcard bits 0.0.0.7

Router2#**ping 172.16.1.1 source loopback 10**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.10.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/77/96 ms

Router2#**ping 172.16.1.1 source loopback 20**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.20.20.3
U.U.U
Success rate is 0 percent (0/5)
Router2#**ping 172.16.1.1 source loopback 30**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.30.30.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/79/96 ms
Router2#
NOTE: Whenever you see a ping fail and the router shows U.U.U it is typically because your ping request was administratively prohibited by an ACL on the other end.
The second lesson to be learned in this exercise is that even though our ACL configuration focused on R3 Loopback10, Loopback20, and Loopback30, because we did not explicitly allow the Serial0/1 subnet between R1 and R3, this is implicitly denied at the end of the ACL.
Keep this in mind, i.e. if traffic is not explicitly permitted, it is implicitly denied. It is very important to understand this aspect in regards to Access Control Lists. Explicitly configured statements show as matches against ACL entries, but implicit deny matches do not.

Router1#**show access-lists**
Standard IP access list 10
    10 permit 10.10.10.0, wildcard bits 0.0.0.127 (15 matches)
    20 deny   10.20.20.0, wildcard bits 0.0.0.15 (11 matches)
    30 permit 10.30.30.0, wildcard bits 0.0.0.7 (15 matches)

Lab 25 Description of lab: **configure 2 routers with an extended access-list**

		
	Router 1	Router 2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Router1	Router2
Ethernet 0/0	172.16.12.1 255.255.255.0 172.16.12.2 255.255.255.0 secondary 172.16.12.3 255.255.255.0 secondary	**172.16.11.1 255.255.255.0** **172.16.11.2 255.255.255.0 secondary**
Serial 1/0	172.16.10.1 255.255.255.0	**172.16.10.2 255.255.255.0**

To complete this lab you will need 2 routers with one Ethernet port and one serial port on each.

Router 1:

router>**en**
router#**conf t**
router(config)#
router(config)#**hostname Router1**
Router1(config)#**int e0/0**
Router1(config-if)#**ip address 172.16.12.2 255.255.255.0 secondary**
Router1(config-if)#**ip address 172.16.12.3 255.255.255.0 secondary**
Router1(config-if)#**ip address 172.16.12.1 255.255.255.0**
Router1(config-if)#**no shut**
Router1(config-if)#**int s0/1**
Router1(config-if)#**ip address 172.16.10.1 255.255.255.0**
Router1(config-if)#**clockrate 64000**
Router1(config-if)#**no shut**
Router1(config-if)#
Router1(config-if)#**ip access-group 100 in**
Router1(config-if)#
Router1(config-if)#**no ip classless**
Router1(config)#
Router1(config)#**ip route 172.16.11.0 255.255.255.0 172.16.10.2**
Router1(config)#**access-list 100 permit ip host 172.16.11.2 host 172.16.12.3 log**

Now we'll configure Router 2
router>en
router#conf t
router(config)#
router(config)#**hostname Router2**
Router2(config)#**interface loopback0**
Router2(config-if)#
Router2(config-if)#**ip address 172.16.13.1 255.255.255.0**
Router2(config)#
Router2(config)#**int e0/0**
Router2(config-if)#**ip address 172.16.11.2 255.255.255.0 secondary**
Router2(config-if)#**ip address 172.16.11.1 255.255.255.0**
Router2(config-if)#**no shut**
Router2(config-if)#
Router2(config-if)#**no keepalive**
Router2(config-if)#**int s1/0**
Router2(config-if)#**ip address 172.16.10.2 255.255.255.0**
Router2(config-if)#**no shut**
Router2(config-if)#**no ip classless**
Router2(config)#
Router2(config)#**ip route 172.16.12.0 255.255.255.0 172.16.10.1**

Now we'll test the route
Router2#**ping**
Protocol [ip]:
Target IP address: **172.16.12.3(secondary address on Router1 E0)**
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: **5**
Extended commands [n]: **y**
Source address or interface: **172.16.11.2(secondary address on Router2 E0)**
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.3, timeout is 5 seconds:
Packet sent with a source address of 172.16.11.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/80/120 ms
Router2#

Now let's change the Access-list on Router1.
First we will need to remove the old access-list

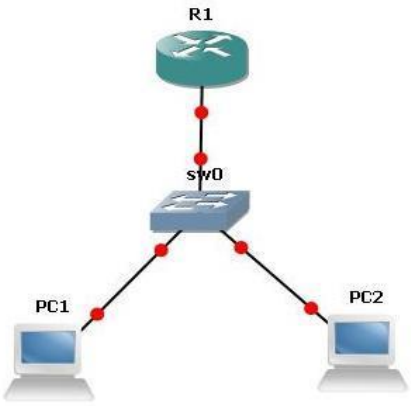
Router1(config)#**no access-list 100**
The "100" is used because of the number we gave to this list.Check to make sure the access-list has been removed
Router1#**show access-lists**

Next we will create a new access list, this time denying access
Router1(config)#access-list 100 deny ip host 172.16.11.2 host 172.16.12.3 log

Again check to see if it there
Router1#show access-lists
Extended IP access list 100
  10 deny ip host 172.16.11.2 host 172.16.12.3 log

Next we go to Router2 and test the results
Router2>**en**
Router2#**ping**
Protocol [ip]:
Target IP address: **172.16.12.3**
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: **5**
Extended commands [n]: y
Source address or interface: 172.16.11.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.3, timeout is 5 seconds:
Packet sent with a source address of 172.16.11.2
U.U.U
Success rate is 0 percent (0/5)
Again notice the "U.U.U" which tells us that the ACL is preventing the ping, rather than the normal ".....
Success rate is 0 percent (0/5)" You will see from non-routable failure.

Lab 26 Description of lab: **Setup a simple VLAN**

				
	Router 1	PC1	PC2	sw0
Router Model	3640	3640	3640	3640
IOS	12.4	12.4	12.4	12.4
Hostname Name	Router1	PC1	PC2	sw0
Fa 0/0	10.1.1.4	10.1.1.1	10.1.1.2	
Fa 0/1				
Fa 0/2				

Configure r1 as Router1 and give it an ip address.

Configure Router
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router1
Router1(config)#int fa0/0
Router1(config-if)#ip address 10.1.1.4 255.255.255.0
Router1(config-if)#no shutdown

Then PC1

Router>enable
Router#conf
Router#configure t
Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ho
Router(config)#hostname PC1
PC1(config)#int fa0/0
PC1(config-if)#ip add
PC1(config-if)#ip address 10.1.1.1 255.255.255.0
PC1(config-if)#no sh
PC1(config-if)#no shutdown
PC1(config-if)#end

Now PC2

Router>enable
Router#configure terminal
Router(config)#hostname PC2
PC2(config)#int fa0/0
PC2(config-if)#ip add
PC2(config-if)#ip address 10.1.1.2 255.255.255.0
PC2(config-if)#no sh
PC2(config-if)#no shutdown

Now go to Router1 and ping PC1 and PC2

Router1#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 80 percent (4/5), round-trip min/avg/max = 36/68/128 ms

Router1#**ping 10.1.1.2**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/63/108 ms

Now go to sw0

Router>enable
Router#configure terminal
Router(config)#hostname sw0
sw0(config)#end
sw0#show cdp neighbors detail

The output from this command should give you the details of all the devices connected to the "switch) (Router1 PC1 and PC2), including the name of the device, the ip address and what interface is attached.
Here is a brief example of some of the output regarding Router1:

```
-------------------------
Device ID: Router1
Entry address(es):
  IP address: 10.1.1.4
Platform: Cisco 3640,  Capabilities: Router Switch IGMP
Interface: FastEthernet0/0,  Port ID (outgoing port): FastEthernet0/0
```

Now lets ping PC2 from PC1 and make sure they are communicating.

```
PC1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/53/96 ms
PC1#
```

Go back to sw0 and we'll configure the vlans. First let's look at the VLANs already configured.

```
sw0#show vlan-switch
```

As we can see from the output we only have a VLAN 1. This is the default VLAN that is on the "switch" before it has been configured. So let's configure a VLAN 20 and assign it to port fa0/2, which is the port that PC2 is on.

```
sw0#vlan database
sw0(vlan)#vlan 1
VLAN 1 modified:
sw0(vlan)#vlan 20
VLAN 20 added:
   Name: VLAN0020
sw0(vlan)#apply
APPLY completed.
sw0(vlan)#exit
APPLY completed.
Exiting....
sw0#conf t
sw0(config)#int fa0/2
sw0(config-if)#switchport access vlan 20
sw0(config-if)#switchport mode access
sw0(config-if)#end
sw0#show vlan-switch
```

As you can see, we now have a new VLAN and it is assigned to port fa0/2. So lets go back to PC1 and ping PC2.

PC1#**ping 10.1.1.2**

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)

Now that we have PC2 on VLAN 20 and PC1 is on VLAN 1. Since Router1 is also on VLAn 1, let's see what happens when we ping PC1 and then PC2.

Router1#**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 = 32/53/104 ms
Router1#**ping 10.1.1.2**

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)

So, again we can see that PC1 is accessible and PC2, because it is on a separate VLAN, is not. Now let's add PC1 to VLAN 20.

sw0(config)#**int fa0/1**
sw0(config-if)#**switchport access vlan 20**
sw0(config-if)#**switchport mode access**
sw0(config-if)#end

As you can see from the output of "show vlan-switch" the port that PC2 is on, is now part of VLAN 20.

sw0#**show vlan-switch**

```
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/0, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15
20   VLAN0020                         active    Fa0/1, Fa0/2
```

So let's ping PC1 from PC2

PC2#**ping 10.1.1.2**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

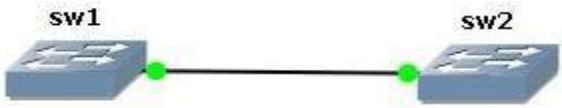
But, as you can see from pinging the 2 pc's from Router1. Router1 itself cannot communicate with either "PC"

Router1#**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 0 percent (0/5)
Router1#ping **10.1.1.2**

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)

Lab 27 Description of lab: **Setup a simple VTP**

		
	Sw1	sw2
Router Model	3640	3640
IOS	12.4	12.4
Hostname Name	Sw1	Sw1
fa 0/0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0

First we'll set an ip address on Switch 1
Router>**en**
Router#**conf t**
Router(config)#**hostname sw1**
sw1(config)#**int vlan 1**
sw1(config-if)#**ip address 10.1.1.1 255.255.255.0**
sw1(config-if)#**no shutdown**

Next, we'll set an ip address on Switch 2
Router>**en**
Router#**conf t**
Router(config)#**hostname sw2**
sw2(config)#**int vlan 1**
sw2(config-if)#**ip address 10.1.1.2 255.255.255.0**
sw2(config-if)#**no shutdown**
sw2(config-if)#**end**
sw2#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 80 percent (4/5), round-trip min/avg/max = 44/63/104 ms

Now configure sw1 vlan database

```
sw1#vlan database
sw1(vlan)#vlan 8
VLAN 8 added:
   Name: VLAN0008
sw1(vlan)#vlan 14
VLAN 14 added:
   Name: VLAN0014
sw1(vlan)#exit
APPLY completed.
Exiting....
```

Then we assign the VLANs to the port ranges

```
sw1#conf t
sw1(config)#interface range fastEthernet 0/2 - 5
sw1(config-if-range)#switchport access vlan 8
sw1(config-if-range)#exit
sw1(config)#interface range fa0/6 - 10
sw1(config-if-range)#switchport access vlan 14
sw1(config-if-range)#exit
sw1(config)#exit
sw1#show vlan-switch

VLAN Name                       Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                    active    Fa0/0, Fa0/1, Fa0/11, Fa0/12
                                          Fa0/13, Fa0/14, Fa0/15
8    VLAN0008                   active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
14   VLAN0014                   active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                          Fa0/10
1002 fddi-default               active
1003 token-ring-default         active
1004 fddinet-default            active
1005 trnet-default              active
```

Next, we make sure that sw1 is acting as the server (this is the default setting, but we'll set it to make sure).

```
sw1#vlan database
sw1(vlan)#vtp server
Device mode already VTP SERVER.
sw1(vlan)#vtp domain JohnMason
Changing VTP domain name from NULL to JohnMason
sw1(vlan)#vtp password gns3
Setting device VLAN database password to gns3.
sw1(vlan)#exit
APPLY completed.
Exiting....
```

No, we set sw2 as the client

sw2#**vlan database**
sw2(vlan)#**vtp client**
Setting device to VTP CLIENT mode.
sw2(vlan)#**vtp domain JohnMason**
Changing VTP domain name from NULL to JohnMason
sw2(vlan)#**vtp password gns3**
Setting device VLAN database password to gns3.

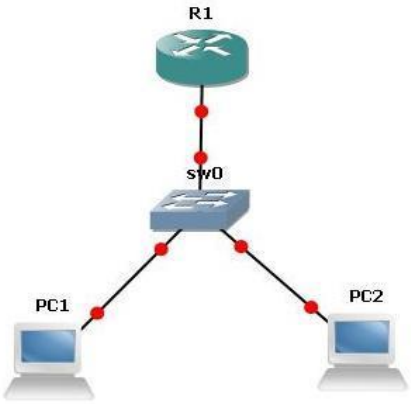
Then create the trunk link that will transport the VLAN configurations from switch 1 to switch 2

sw1#**conf t**
sw1(config)#**int fa0/12**
sw1(config-if)#**switchport mode trunk**
sw1(config-if)#end
sw1#

And then on switch 2 we also set the trunk link.

sw2#**conf t**
sw2(config)#**int fa0/12**
sw2(config-if)#**switchport mode trunk**
sw2(config-if)#**end**

Lab 28 Description of lab:**Router on a stick**

				
	Router 1	Switch 1	PC1	PC2
Router Model	3640	3640	3640	3640
IOS	12.4	12.4	12.4	12.4
Hostname Name	Router1	Switch1	PC1	PC2
Vlan 10	192.168.1.1 /24		192.168.1.2/24	
Vlan 20	192.168.2.1 /24			192.168.2.2 /24

We set up the Switch by setting up the VLANs first.

Router>**en**
Router#**vlan database**
Router(vlan)#**vlan 1**
VLAN 1 modified:
Router(vlan)#**vlan 10**
VLAN 10 added:
  Name: VLAN0010
Router(vlan)#**vlan 20**
VLAN 20 added:
  Name: VLAN0020
Router(vlan)#**apply**
APPLY completed.
Router(vlan)#**exit**
APPLY completed.

Exiting....
Router#**conf t**
Then assign the vlans to the ports that the PC's are on and set the trunking to the Router.
Router(config)#**hostname Switch1**
Switch1(config)#**int fa0/0**
Switch1(config-if)#**switchport trunk encapsulation dot1q**
Switch1(config-if)#**switchport mode trunk**
Switch1(config-if)#**int fa0/1**
Switch1(config-if)#**switchport access vlan 10**
Switch1(config-if)#**int fa0/2**
Switch1(config-if)#**switchport access vlan 20**
Switch1(config-if)#**switchport mode access**
Switch1(config-if)#**exit**
Switch1(config)#**exit**

Now we'll configure Router1, this is called the "Router on a Stick". The router will be connecting the 2 Vlans together. So, first we enter the Vlan data.
Router>**en**
Router#**conf t**
Router(config)#**hostname Router1**
Router1(config)#**exit**
Router1#vlan database
Router1(vlan)#**vlan 10**
VLAN 10 added:
  Name: VLAN0010
Router1(vlan)#**vlan 20**
VLAN 20 added:
  Name: VLAN0020
Router1(vlan)#**apply**
APPLY completed.
Router1(vlan)#**exit**
APPLY completed.
Exiting....
Now we will create sub interfaces on fa 0/0 with ip addresses for each vlan.
Router1#conf t
Router1(config)#**int fa0/0**
Router1(config-if)#**no shutdown**
Router1(config-if)#**int fa0/0.10**
Router1(config-subif)#**encapsulation dot1q 10**
Router1(config-subif)#**ip address 192.168.1.1 255.255.255.0**
Router1(config-subif)#**int fa0/0.20**
Router1(config-subif)#**encapsulation dot1q 20**
Router1(config-subif)#**ip address 192.168.2.1 255.255.255.0**
Router1(config-subif)#**end**

PC1 is connected to the 'switch' on the switches port fa 0/1. This will be VLAN 10.
Router>en
Router#conf t

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname PC1
PC1(config)#int fa0/0
PC1(config-if)#no shut
PC1(config-if)#ip address 192.168.1.2 255.255.255.0
PC1(config-if)#exit
PC1(config)#no logging console
PC1(config)#exit

Let's ping the router from PC1
PC1#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 16/49/84 ms

Now we'll set the gateway of last resort, to the router.
PC1#**conf t**
PC1(config)#**ip route 0.0.0.0 0.0.0.0 192.168.1.1**
PC1(config)#

PC2 is connected to the 'switch' on the switches port fa 0/2. This will be VLAN 20.
Router>**en**
Router#**conf t**
Router(config)#**hostname PC2**
PC2(config)#int **fa0/0**
PC22config-if)#**no shut**
PC2(config-if)#**ip address 192.168.2.2 255.255.255.0**
PC2(config-if)#**exit**
PC2(config)#**no logging console**
PC2(config)#**exit**

Let's ping the router from PC2
PC2#**ping 192.168.2.1**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 16/49/84 ms

Now we'll set the gateway of last resort, to the router.
PC2#**conf t**
PC2(config)#**ip route 0.0.0.0 0.0.0.0 192.168.2.1**