Faculty of Computer Science and Engineering

Computer Networks Lab

CE413L

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Computer Networks Lab Manual

Contents

Grading

Grading in Percentage	
Assessment Items	**Percentage**
Lab Performance	**40%**
Midterm Exam	**20%**
OEL	**10%**
Final Exam	**30%**

CLO_PLO Mapping

Sr. No	Course Learning Outcomes	PLOs PLOs are for BS (CE) only	Taxonomy Level (Psychomotor Domain)
CLO_1	Efficiently identify underlying features of networking and be to analyze solutions of networks	PLO-3	P3 (Guided response)
CLO_2	Efficiently apply solutions to practical problems of networking	PLO-5	P3 (Guided response)

CLO Assessment Mechanism

Assessment Items	CLO1	CLO2
Lab Performance	✓	
Midterm	✓	
OEL	-	✓
Final	-	✓

Rubric

Rubric for Computer Networks Lab **Instructor:**_____

Class/ Section _____

Date: _____

Criteria	Excellent	Good	Average	Need Improvement
Completeness (3)	3	2-2.5	1-1.5	<1
Simulation (2)	2	1.5	1	<1
Layout (2)	2	1.5	1	<1
Viva (3)	3	2-2.5	1-1.5	<1

Lab 01

WireShark (Ethereal) Tutorial & Remote Packet Capturing

The aims of today's lab are to introduce a packet sniffer to the students. There are several typesof Packet Capturing Software, The one which is most commonly used and is freely available i.e. Open Source is known as WireShark.

Learning Objectives

Our aims today are:

- Learning the GUI of Wireshark.
- To Capture Packets via WireShark
- Packet Capturing of your PC
- OSI(Open System Interconnection) model

WireShark (Ethereal) Tutorial & Remote Packet Capturing Background

Wireshark is a software protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education.

Before June 2006, Wireshark was known as Ethereal. A packet sniffer (also known as a network analyzer or protocol analyzer) is computer software that can Intercept and log data traffic passing over a data network. As data streams travel back and forth over the network, the sniffer "captures" each protocol data unit (PDU) and can decode and analyze its content according to the appropriate RFC or other specifications.

Wireshark is programmed to recognize the structure of different network protocols. This enables it to display the encapsulation and individual fields of a PDU and interpret their meaning. It is a useful tool for anyone working with networks and can be used with most labs in the CCNA courses for data analysis and troubleshooting.

For informationand to download the program go to - http://www.Wireshark.org, it's also pricedright: it's free! Our administrator has already installed Wireshark. Meanwhile, if you have a personal PC and Internet access, you can install Wireshark onto your PC. You can find the free software at Http://www.wireshark.org Once Wireshark is up, please followinstructions in the file Wireshark Tutorial. You need to answer the questions when you are finished with reading this tutorial and Upload the Answers on the course website.

Running Wireshark

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed. Initially, no data will be displayed in the various windows.

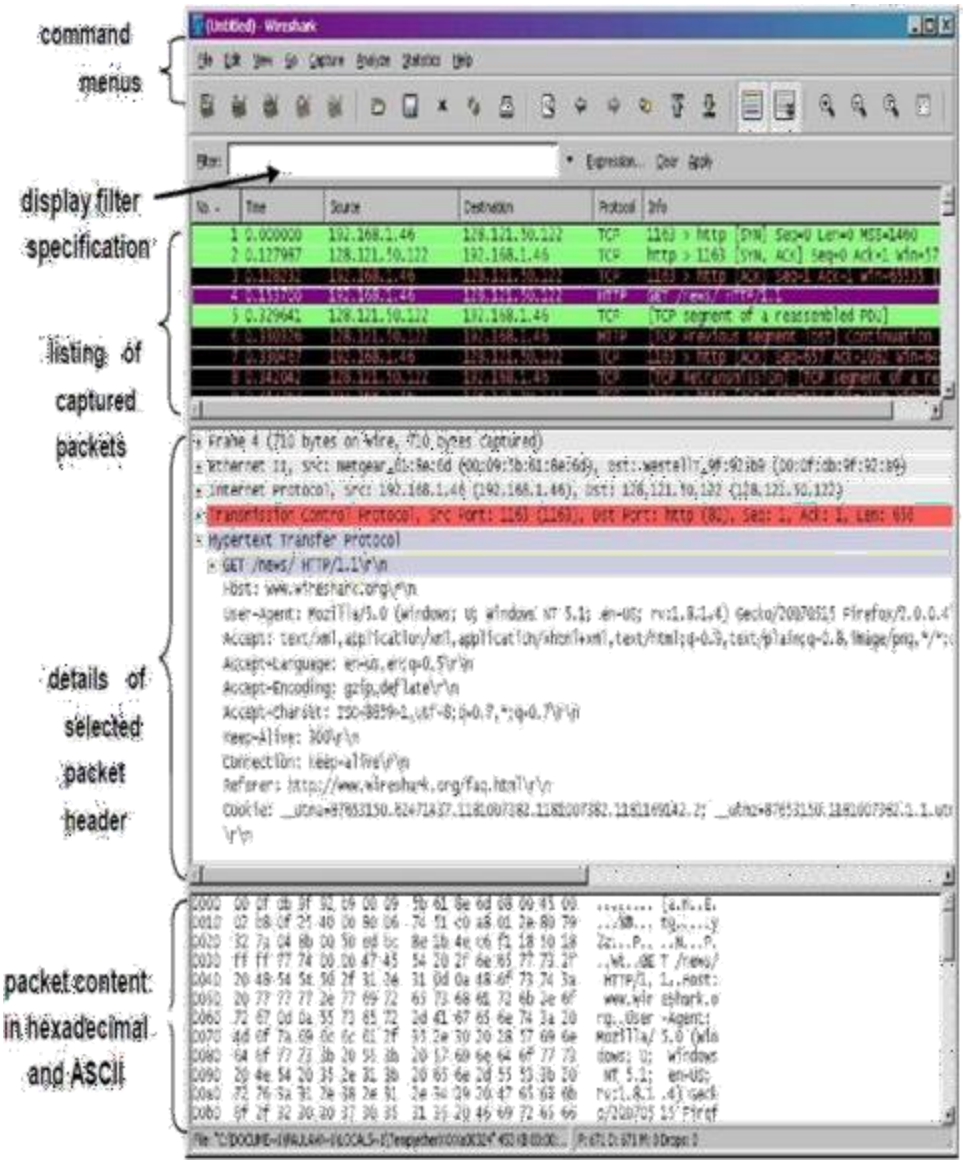


Figure 2 Wireshark Graphical User Interface

The Wireshark interface has five major components: The command menus are standard pull down menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

The packet-listing window displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type fieldliststhe highest level protocol that sent or received this packet, i.e., the Protocol that is the source or ultimate sink for this packet.

The packet-header details window provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet's one-

line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/receivedover an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized.

Finally, details about the highest level protocol that sent or received this packet are also provided. The packet-contents window displays the entire contents of the captured frame, in both ASCII and hexadecimal format. Towards the top of the Wireshark graphical user interface, is the packet display filter field, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

Taking Wireshark for a Test Run The best way to learn about any new piece of software is to try it out! We'll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following:

1. Start up your favorite web browser, which will displayyour selectedhomepage. Start up the Wireshark software. You will initially see a window similar to that shown in Figure 2, except that no packet data will be displayed in the packet listing, packet-header, or packet-contentswindow, since Wireshark has not yet begun capturing packets.
2. To begin packet capture, select the Capture pull down menu and select Options. This will cause the "Wireshark: Capture Options" window to be displayed, as shown in Figure 3.

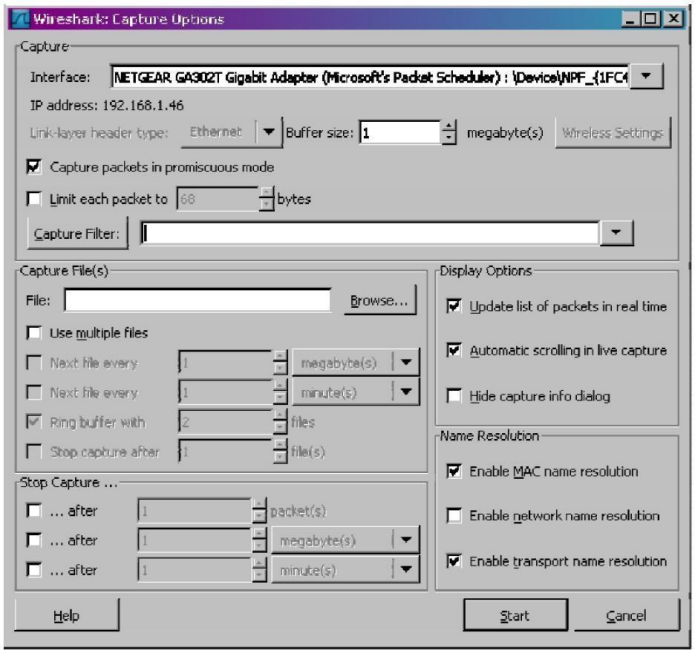


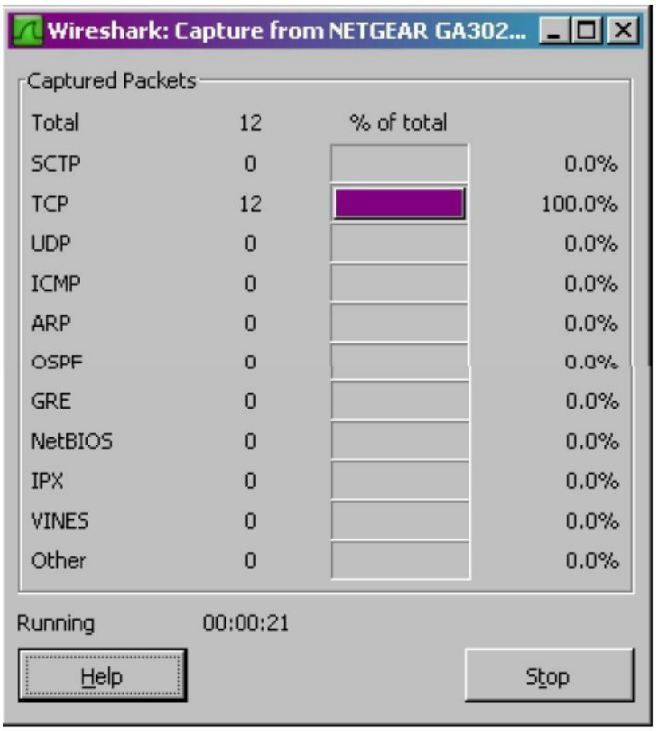
Figure 3 wireshark capture window options

3. You can use most of the default values in this window, but uncheck "Hide capture info dialog" under Display Options. The networks interfaces (i.e., the physical connections) that your computer has to the

network will be shown in the Interface pull downmenu at the top of the Capture Optionswindow. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - all packets being sent/received from/by your computer are now being captured by Wireshark!

4. Once you begin packet capture, a packet capture summary window will appear, as shown in Figure 4. This window summarizes the number of packets of various types that are being captured, and

(Importantly!) Contains the Stop button that will allow you to stop packet capture. Don't stop packet



capture yet.

Figure 4 wireshark packet capture window

5. While WireShark is running, enter the URL: www.yahoo.com or some other website URL and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at www.yahoo.com and exchange HTTP messages with the server in order to download this page. The Ethernet frames containing these HTTP messages will be captured by WireShark.

After your browser has displayed the page, stop WireShark packet capture by selecting stop in the WireShark capture window. This will cause the WireShark capture window to disappear and the main WireShark window to display all packets captured since you began packet capture. The main WireShark window should now look similar to Figure 2. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the www.yahoo.com web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the *Protocol* column in Figure 2). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user.

We'll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than "meets the eye"!

7. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in WireShark) into the display filter specification window at the top of the main WireShark window. Then select Apply (to the right of where you entered"http"). This will cause only HTTP message to be displayed in the packet-listing window.

8. Select the first http message shownin the packet-listing window. This should be the HTTP GET message that was sent from your computer to the www.yahoo.com HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window3. By clicking plus-and-minus boxes to the left side of the packet details window, minimize the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol.

Your Wireshark display should now look roughly as shown in Figure 5. (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packetheader window).

9. Exit Wireshark

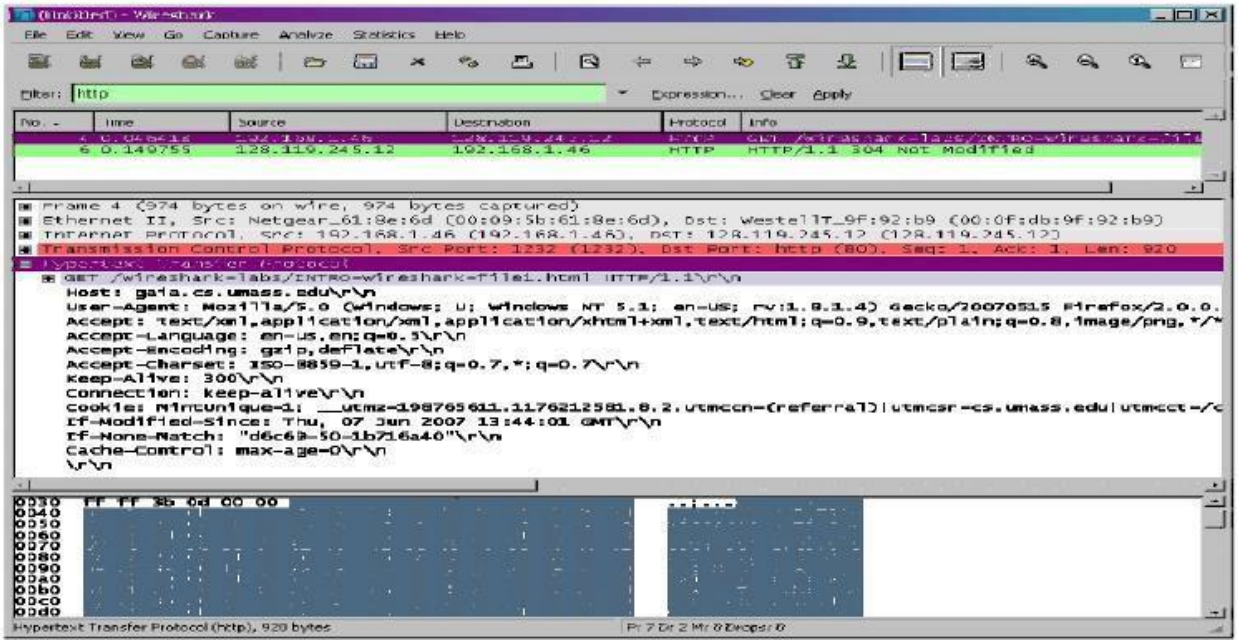


Figure 5 wireshark display after step 9

Computer network is a group of two or more computers those are agree to share information, resources and data between themunder certain networking terms. Internetworking isa way to connect two or more computer networks. In this tutorial I will explain what computer network is, why we need to break a large computer network in smaller networks and how networking devices are used to create an internetwork.

Exercise Part-I

Launch a web browser on the computer that is running Wireshark. Enter the URL of the Eagle Server of **example.com** .When the webpage has fully downloaded, stop the Wireshark packet capture. Filter http for the following Questions

Answer the following questions:
1. Is your browser running HTTP version 1.0 or 1.1?
Answer:

2. What languages (if any) does your browser indicate that it can accept to the server?
Answer:

3. What is the IP address of your computer? Of the www.yahoo.com?
Answer:

4. What's the header length of the IP Packet?
Answer:

5. What is the status code returned from the server to your browser?
Answer:

6. When was the HTML file, which you are retrieving, last modified at the server?
Answer:

7. Write down the Source and Destination Mac Addresses?
Answer:

8. What protocols are in the Ethernet frame?
Answer:

9. How Wireshark can help you to find Network

delays?

Answer:

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 02
Introduction to Packet Tracer and Basic configuration of Switch

Dear Students today's lab exercise will result in the following outcomes Please ensure you complete all the steps yourself and don't indulge in any kind of cross talk or interference in the work of other students.

Learning Objectives:

- Introduction to Packet Tracer
- Configuration of a Cisco Switch
- Identifying the Modes of a Cisco Switch
- Some Basic Commands running

Introduction to Packet Tracer and Basic configuration of Switch

Objectives:

This lab teaches the basics of using Packet Tracer. Packet Tracer is a protocol simulator developed by Dennis Frezzo and his team at Cisco Systems. Packet Tracer (PT) is a powerful and dynamic tool that displays the various protocols used in networking, in either Real Time or Simulation mode and also provides us with the real environment of different devices like PC, Cisco switches, routers etc.

Overview:

Packet Tracer provides a Virtual Network Environment that models the behavior of networks, including its routers, switches, protocols, servers etc. This helps us to work on router and switches environment without purchasing them and allow us o learn their configurations.

In this lab you‟ll learn how to connect two PC‟s together via simple cross-over cable, connecting PC to a switch and also configure it for basic functions. You‟ll also learn few basic commands to configure a switch.

Lab Instructions:

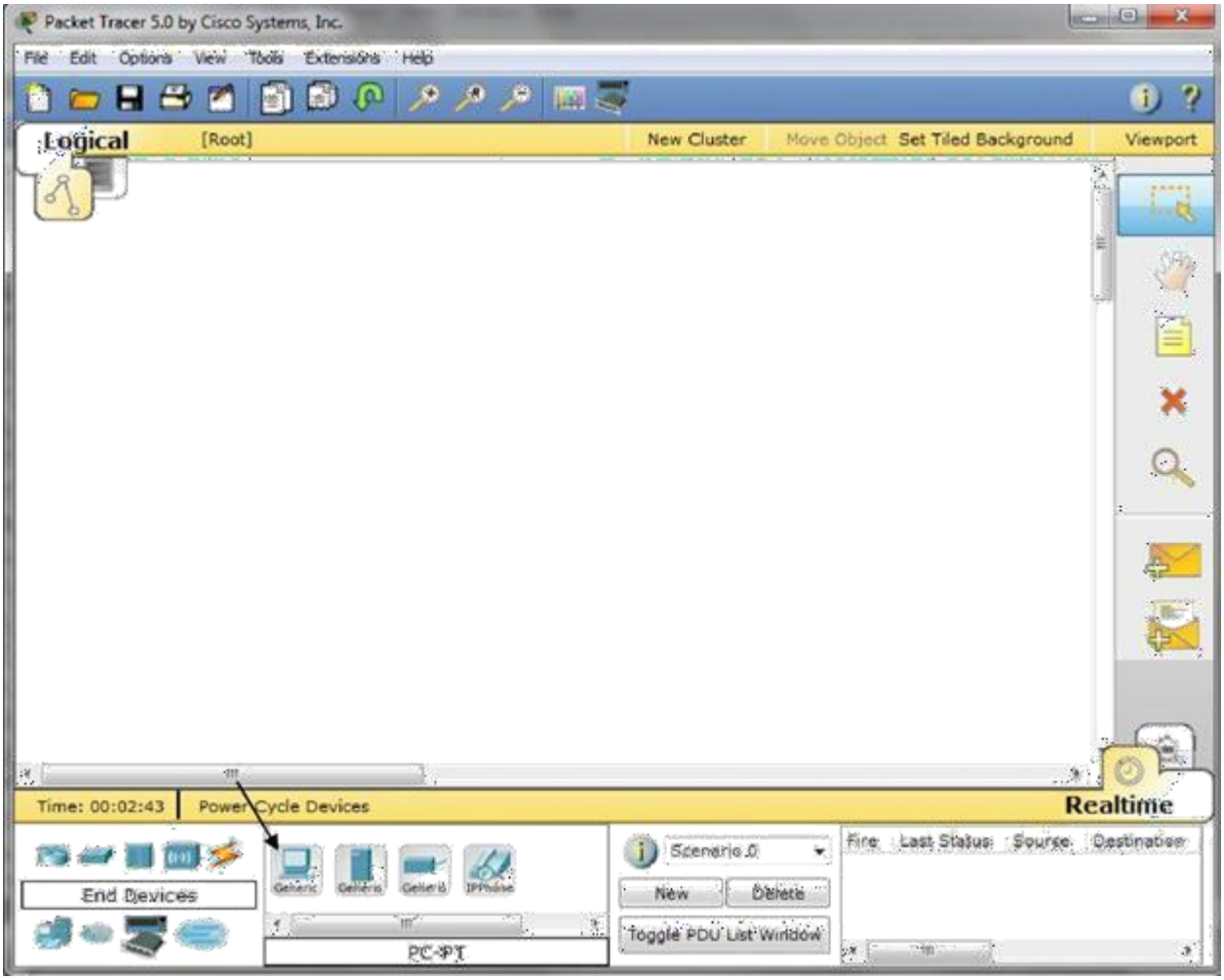
Connecting two PCs via cross-over cable.

Step 1:

Launch packet tracer.

Step 2:

Select end devices in the left bottom corner (Ctrl+Alt+E) and select PC (Generic) your cursor will suddenly change in plus sign

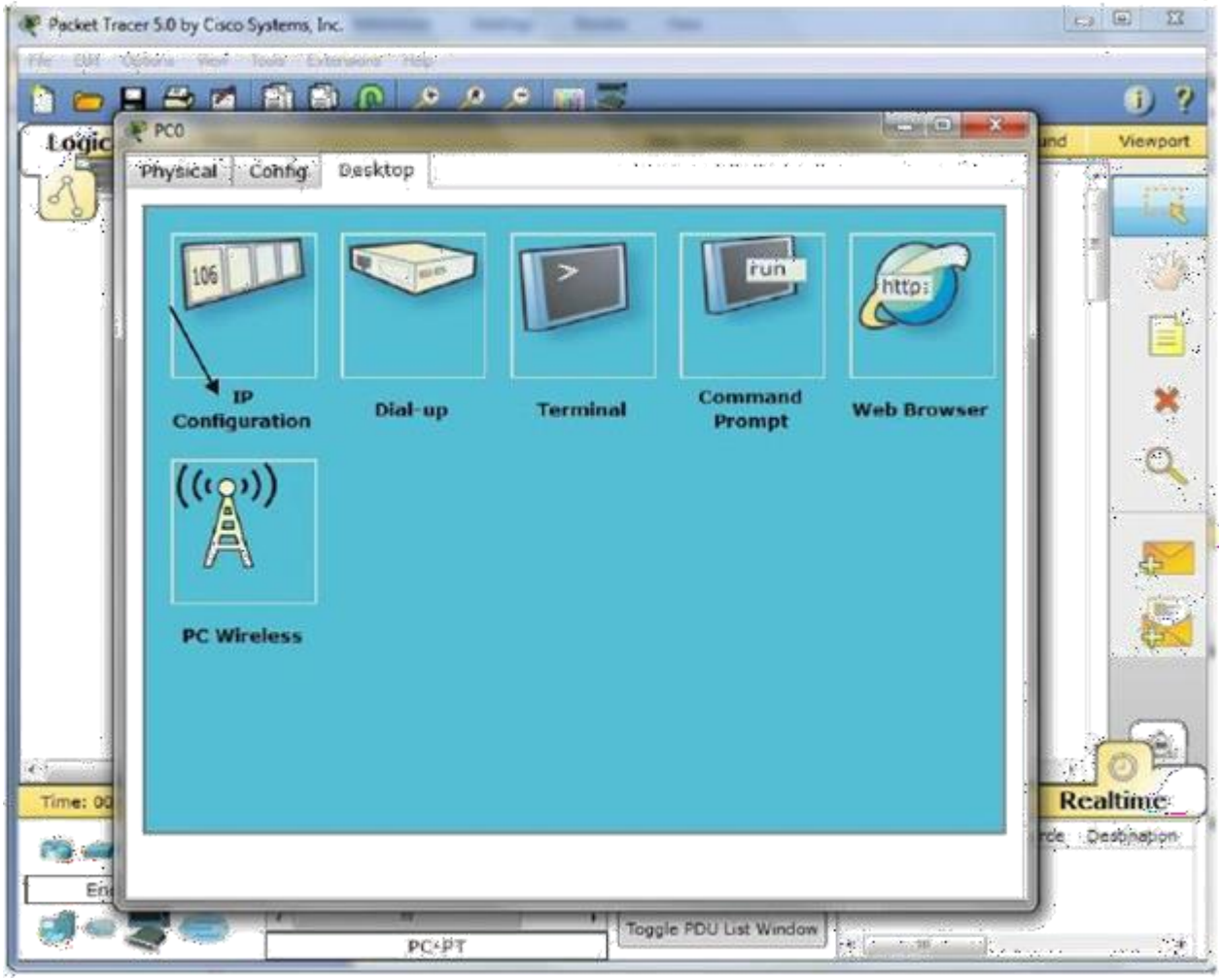


Step 3:
Now place the PC on the workspace above devices. Do it for another PC.

Step 4:
Now Select flash sign from left bottom side select the cross-over cable and connect two PCs.

Step 5:

Double click on PCs one by one and select the Desktop tab from new appeared window. Then select ip configuration and give an ip address to each PC of same ip range and subnet mask too.



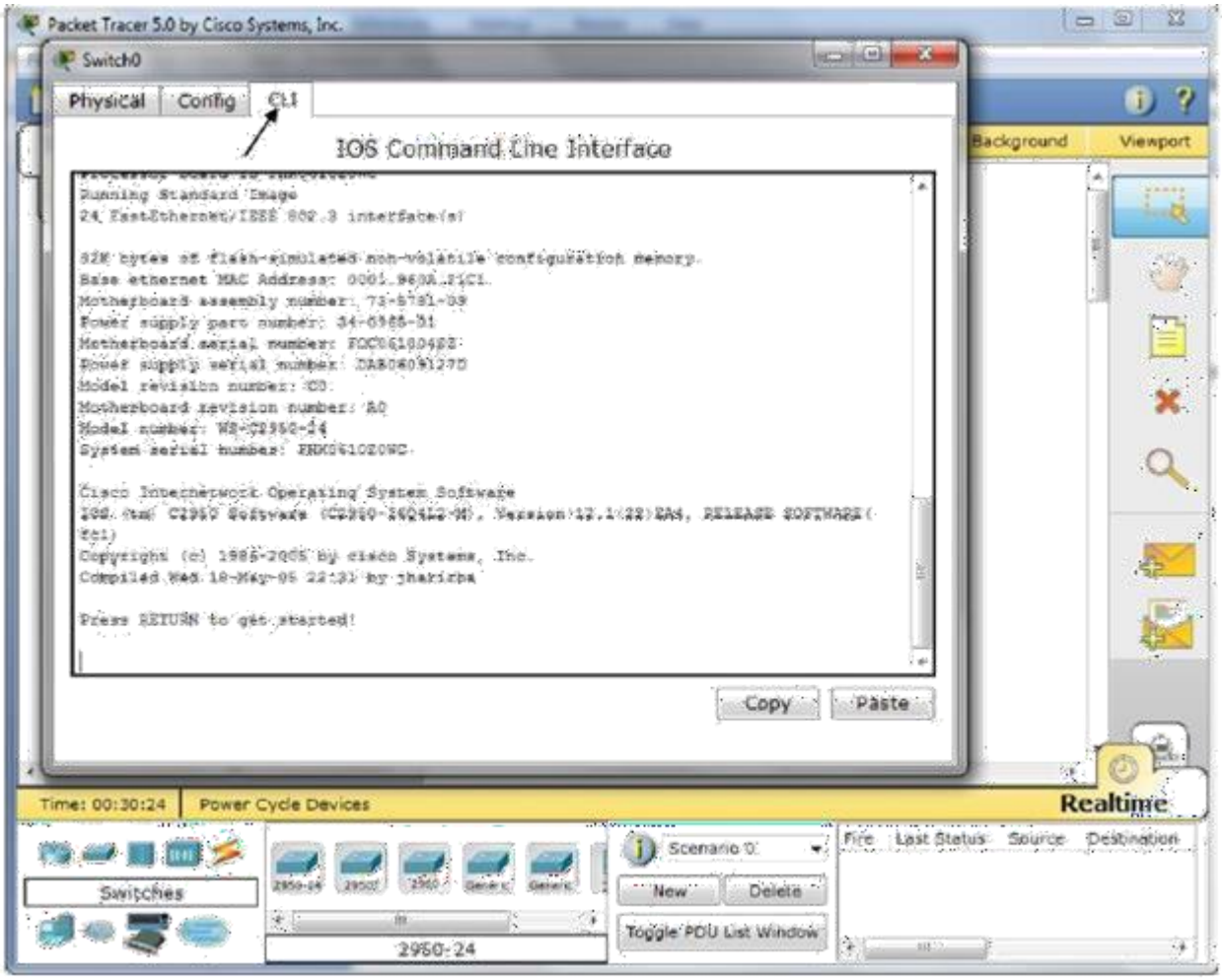
Step 6:
After assigning ip addresses to both pc select Command prompt of any pc and ping the other ip address. You'll receive ping replies on successful connection and ip address assignment.

Configuring Switch

Step 1:
Select switches from left bottom tabs and pick first switch 2950-24 and paste it on workspace above it.

Step 2:

Now double click on switch to go into command line interface of the switch. Select CLI from new appeared window.



Step 3:
Press “Enter” to get into user mode.

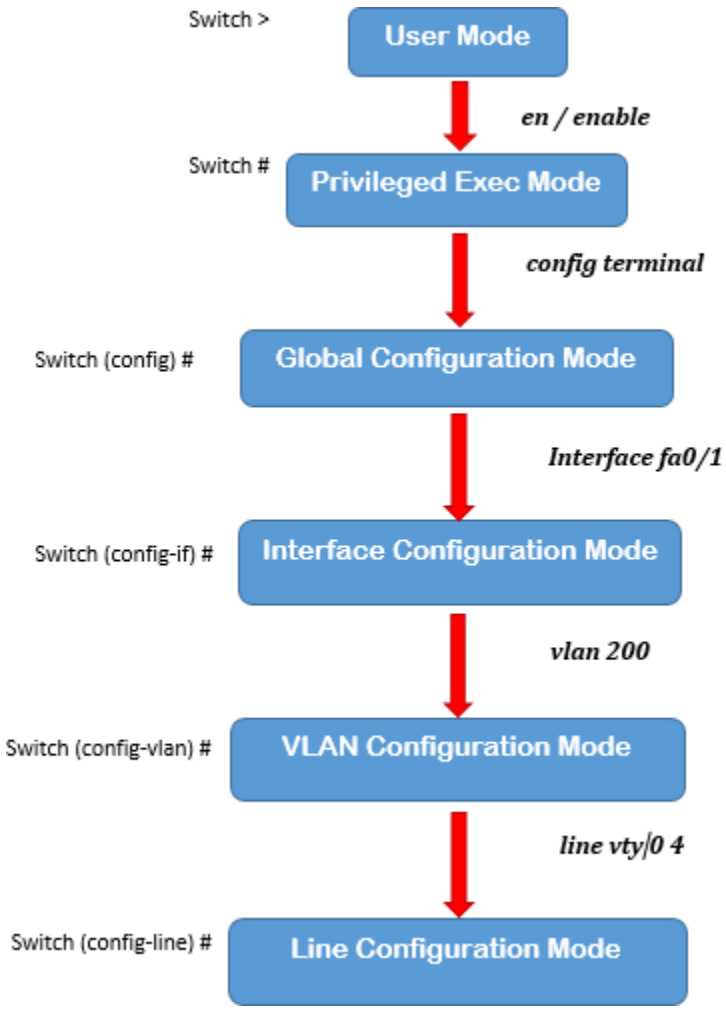
Step 4:
Enter command “enable” or “en” to enter in EXEC mode.
Switch>enable

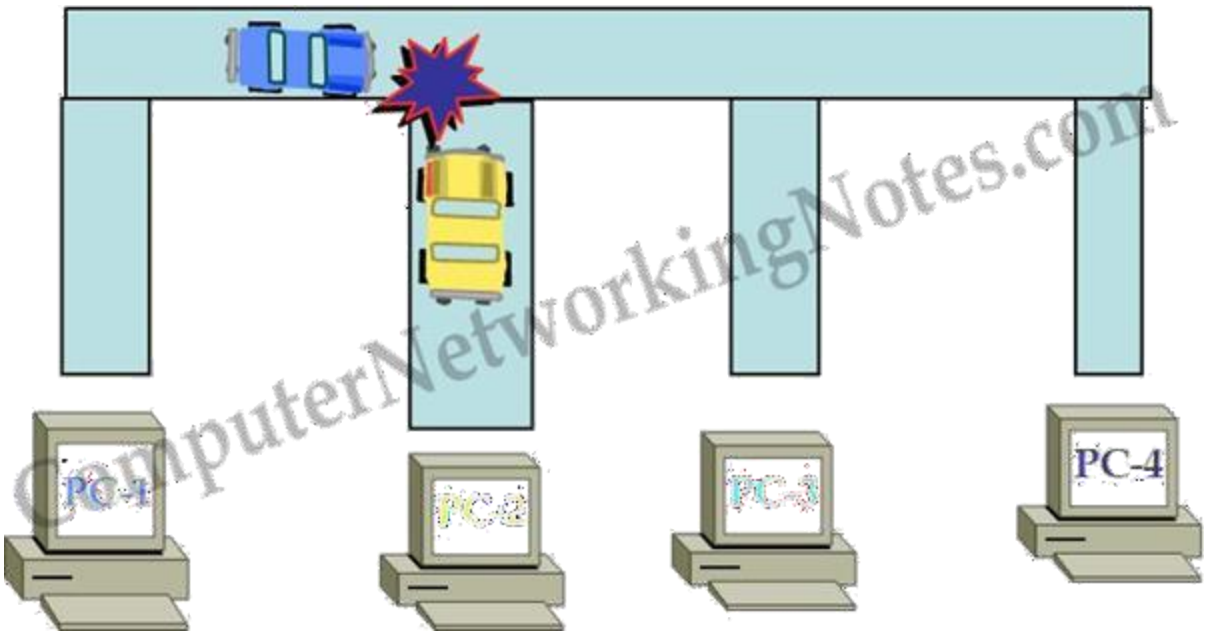
Step 5: In privileged EXEC mode, type ? Note the list of available commands.
After getting into EXEC mode „>“ sign will change into „#“. There are now more available commands compared to user EXEC mode. In addition to the basic monitoring commands, configuration and management commands can now be accessed.

Step 6: Change to global configuration mode.
Switch#configure terminal
Switch(config)#

Step 7: In global configuration mode, type ?. Note the list of available commands.

Switch Modes





CSMA/CD

Carrier Sense Multiple Access/ Collision Detection (CSMA/CD) is a process that is used to send the data in shared environment.

- This mechanism is only used in HUB or shared environment.
- All devices have equal priority.
- In this process only one device can send data at a time.
- Before a device send data, it will first sense the wire to ensure that no other device is currently sending data. If other device is currently using the media, it will have to wait till that transmission is over. If no device is currently using wire it can send the data.
- If two or more devices simultaneously sense wire and see no data in it, they could place their data on the wire at same time.
- In this situation collision will occur.
- When a collision is occurred, a special jam signal is created in the wire.
- Jam signal has a waiting time period.
- All devices have to wait till jam signal time period is over.
- Once this time period is over, devices can sense the wire again.
- Devices those data lost in collision have to resend the same piece of data again.

Collision domain

Collision domainis a group of computers those share same collision. More computers you put in network, more collisionyou will experience. Collision seriously effect network performance. Collision should be less than one percent of total traffic. If it increases, we will have to implement collision removal devices. Bridges, switches, routers, multilayer switches can control the collision.

Summary

Device	Collision	Broadcast
HUB	Single collision domain	Single broadcast domain
Bridge	Per port collision domain	Single Broadcast domain
Switch	Per port collision domain	Single Broadcast domain
Router	Per port collision domain	Per port broadcast domain
Multilayer switch	Per port collision domain	Per port broadcast domain

Switch Functions at Layer 2

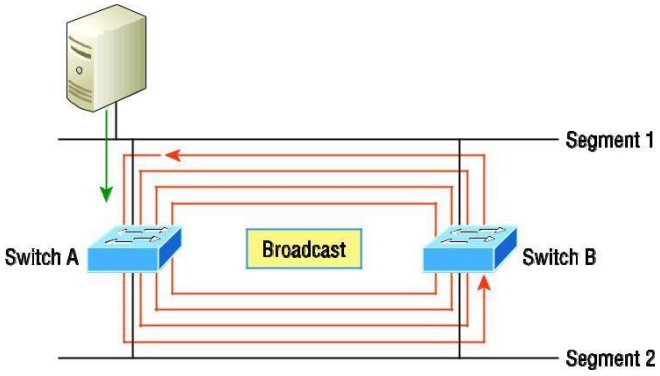
Three Switch Functions at Layer 2 There are three distinct functions of layer 2 switching that are vital for you to remember:

- Address learning.
- Forward/filter decisions.
- Loop avoidance.

Address learning Layer 2 switches remember the source hardware address of each frame received on an interface and enter this information into a MAC database called a forward/filter table.

Forward/filter decisions when a frame is received on an interface, the switch looks at the destination hardware address, and then chooses the appropriate exit interface for it in the MAC database. This way, the frame is only forwarded out of the correct destination port.

Loop avoidance if multiple connections between switches are created for redundancy purposes, network loops can occur. Spanning Tree Protocol (STP) is used to prevent network loops while still permitting redundancy. Next, I'm going to talk about address learning and forward/filtering decisions. Loop avoidance is beyond the scope of the objectives being covered in this chapter.



Address Learning

When a switch is first powered on, the MAC forward/filter table (CAM) is empty, as shown in Figure 3.1. When a device transmits and an interface receives a frame, the switch places the frame's source address in the MAC forward/filter table, allowing it to refer to the precise interface the sending device is located on. The switch then has no choice but to flood the network with this frame out of every port except the source port because it has no idea where the destination device is actually located. If a device answers this flooded frame and sends a frame back, then the switch will take the source address from that frame and place that MAC address in its database as well, associating this address with the interface that received the frame. Because the switch now has both of the relevant MAC addresses in its filtering table, the two devices can now make a point-to-point connection. The switch doesn't need to flood the frame as it did the first time because nowthe framescan and will only be forwardedbetween these two devices. This is exactly why layer 2 switches are so superior to hubs. In a hub network, all frames are forwarded out all ports every time—no matter what. Figure 3.2 shows the processes involved in building a MAC database.

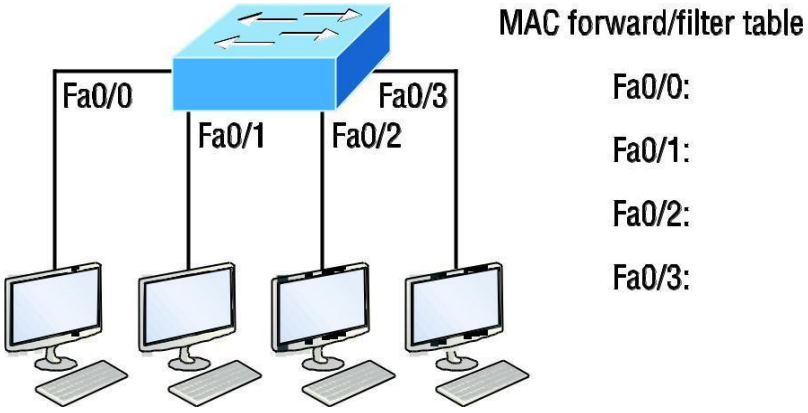


Fig 3.1

In this figure, you can see four hosts attached to a switch. When the switch is powered on, it has nothing in its MAC address forward/filter table, just as in Figure 3.1. But when the hosts start communicating, the switch places the source hardware address of each frame into the table along with the port that the frame's source address corresponds to. Let me give you an example of how a forward/filter table is populated using Figure 3.2:

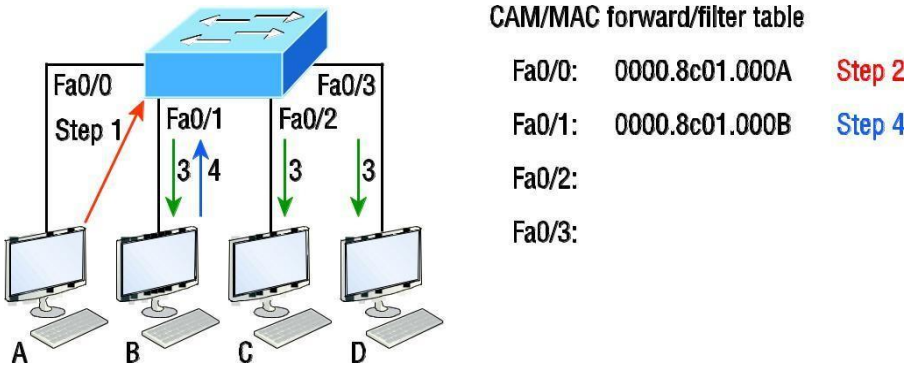


Fig 3.2

1. Host A sends a frame to Host B. Host A's MAC address is 0000.8c01.000A; Host B's MAC address is 0000.8c01.000B.

2. The switch receives the frame on the Fa0/0 interface and places the source address in the MAC address table.

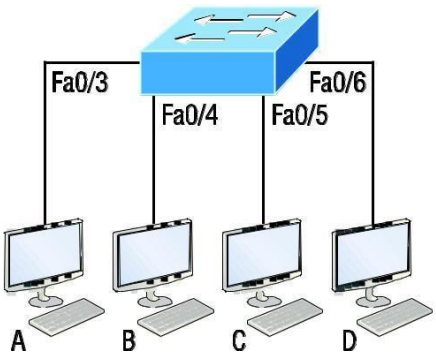
3. Since the destination address isn't in the MAC database, the frame is forwarded out all interfaces except the source port.

4. Host B receives the frame and responds to Host A. The switch receives this frame on interface Fa0/1 and places the source hardware address in the MAC database.

5. Host A and Host B can now make a point-to-point connection and only these specific devices will receive the frames. Hosts C and D won't see the frames, nor will their MAC addresses be found in the database because they haven't senta frame to the switch yet. If Host A and Host B don't communicate to the switch again within a certain time period, the switch will flush their entries fromthe database to keep it as current as possible.

Forward/Filter Decisions

When a frame arrives at a switch interface, the destination hardware address is compared to the forward/filter MAC database. If the destination hardware addressis knownand listed in the database, the frame is only sent out of the appropriate exit interface. The switch won't transmit the frame out any interface except for the destination interface, whichpreserves bandwidth on the other network segments. This process is called frame filtering. But if the destination hardware address isn't listed in the MAC database, then the frame will be flooded out all active interfaces except the interface it was received on. If a device answers the flooded frame, the MAC database is then updated with the device's location— its correct interface. If a host or server sends a broadcast on the LAN, by default, the switch will flood the frame out all active ports except the source port. Remember, the switch createssmaller collision domains, but it's always still one large broadcast domain by default. In Figure 3.3, Host A sends a data frame to Host D. What do you think the switch will do when it receives the frame from Host A?



Switch# show mac address-table

VLAN	Mac Address	Ports
1	0005.dccb.d74b	Fa0/4
1	000a.f467.9e80	Fa0/5
1	000a.f467.9e8b	Fa0/6

Fig 3.3 Forward/filter table

Let's examine Figure 3.4 to find the answer.

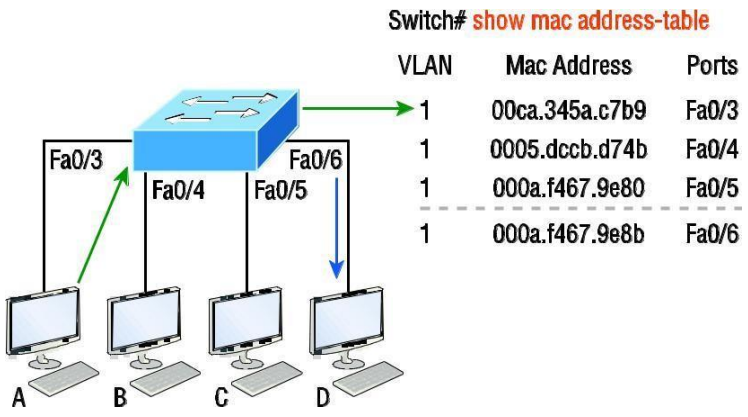


Fig 3.4

Since Host A's MAC address is not in the forward/filter table, the switch will add the source address and port to the MAC address table, then forward the frame to Host D. It's really important to remember that the source MAC is always checked first to make sure it's in the CAM table. After that, if Host D's MAC address wasn't found in the forward/filter table, the switch would've flooded the frame out all ports except for port Fa0/3 because that's the specific port the frame was received on.

CAM Table

```
Switch#sh   mac address-table
Vlan        Mac Address     Type        Ports
----        -----------     --------    -----
1           0005.dccb.d74b              DYNAMIC     Fa0/1
1           000a.f467.9e80              DYNAMIC     Fa0/3
1           000a.f467.9e8b              DYNAMIC     Fa0/4
1           000a.f467.9e8c              DYNAMIC     Fa0/3
1           0010.7b7f.c2b0              DYNAMIC     Fa0/3
1           0030.80dc.460b              DYNAMIC     Fa0/3
1           0030.9492.a5dd              DYNAMIC     Fa0/1
```

The commands structure of a router.

Mode	Definition
User exec mode	Limited to basic monitoring commands
Privileged exec mode	Provides access to all other router commands

Global configuration mode	Commands that affect the entire system [AU: Includes commands that affect…? Also in the next one? Doesn't seem like the mode is commands.]
Specific configuration modes	Commands that affect interfaces/processes only
Setup mode	Interactive configuration dialog

You can configure the following administrative functions on a router and switch:

- Hostnames
- Banners
- Passwords
- Interface descriptions

Hostname

We use the hostname command to set the identity of the router. This is only locally significant, meaning it
doesn't affect how the router performs name lookups or how the device actually works on the internetwork.

Switch#**config t**
Switch(config)#**hostname Todd**

Banners

Message of the day (MOTD) banners are the most widely used banners because they give a message to anyone connecting to the router via Telnet or an auxiliary port or even through a console port as seen here:

Todd(config)#**banner motd ?**

LINE c banner-text c, where 'c' is a delimiting
character Todd(config)#**banner motd #**

Enter TEXT message. End with the character '#'.

$ **Acme.com network, then you must disconnect immediately.**

#

Todd(config)#**^Z** (Press the control key + z keys to return to privileged mode)

Passwords

There are 5 types of password you can configure on switch.

1. Enable secret.
2. Enable password.
3. Line-Console

4. Line vty(Virtual Terminal) for telnet
5. ssh(secure shell)

1. **Enable Password**
   Todd(config)#enable password todd

2. **Enable Secret**

The enable password you have chosen is the same as your enable secret. This is not recommended. Re-enter the enable password.

3. **Console Password**
   Todd(config-if)#line con 0
   Todd (config-line)#password console
   Todd (config-line)#login

1. **Line vty(Virtual Terminal)**
   Todd (config-line)#line vty 015
   Todd (config-line)#password
   telnet Todd (config-line)#login
2. ssh(secure shell)
   Beyound the scope of this lab.

Encrypting your Password

To manually encrypt your passwords, use the service password-encryption command. Here's how:

Todd#**config t**
Todd(config)#**service password-encryption**
Todd(config)#**exit**
Todd#**show run**
Building configuration...
!
!
enable secret 4 ykw.3/tgsOuy9.6qmgG/EeYOYgBvfX4v.S8UNA9Rddg
enable password 7 1506040800

Interface Description

Setting descriptions on an interface is another administratively helpful thing, and like the hostname, it's also only locally significant.

Todd#**config t**
Todd(config)#**int fa0/1**
Todd(config-if)#**description Sales VLAN Trunk Link**
Todd(config-if)#**^Z**
Todd#

And on a router serial WAN:

Todd#**config t**

Todd(config)#**int s0/0/0**

Todd(config-if)#**description WAN to Miami**

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

To verify the interfaces

Todd#**sh ip interface brief**

Interface States

If an interface is shut down, it'll display as administratively down when you use the show interfaces command (sh int for short):

Router#**sh int f0/0**

FastEthernet0/1 is administratively down, line protocol is down
[output cut]

You can bring up the router interface with the no shutdown command (no shut for short):

Router(config)#**int f0/0**

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

*August 21 13:45:08.455: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up

Router(config-if)#**do show int f0/0**

FastEthernet0/0 is up, line protocol is up

[output cut]

TASKS:

Tasks related to the lab will be provided by the lab instructor

Lab 03

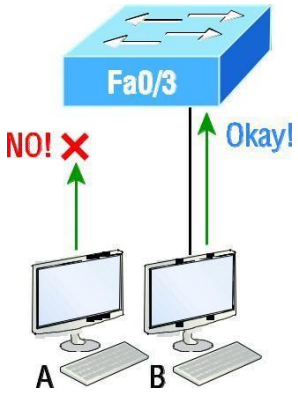
Switch Security and Configuration of DHCP, DNS, FTP and http Server

Learning Objectives:

- Configure and verify Switch Port Security features such as Sticky MAC address limitation Static / dynamic Violation modes Err disable Shutdown Protect restrict Shutdown unused ports
- To learn about Arithmetic Operators
- What is DHCP?
- Difference between static and Dynamic Addressing
- DHCP Configuration on a Server
- Enable Domain Name System (DNS) Service of a Server
- Configuration of File Transfer Protocol Server
- Configuration of http server

Port Security

It's usually not a good thing to have your switches available for anyone to just plug into and play around with. But just how do we actually prevent someone from simply plugging a host into one of our switch ports? By default, MAC addresses will just dynamically appear in your MAC forward/filter database and you can stop them in their tracks by using port security!



Here are your options for configuring port security:

```
Switch#config t                          //go to configuration mode
Switch(config)#int fa0/1                 //access your port
Switch(config-if)#switchport mode access //go to access mode so now you can secure this port
Switch(config-if)#switchport port-security //turn port security ON
```

You can use the *switchport port-security mac-address* command to assign individual MAC addresses to each switch port.

```
Switch(config-if)#switchport port-security maximum 3 //range 1 0 3072
Switch(config-if)#switchport port-security mac –address ---------
Switch(config-if)#switchport port-security mac –address E0-69-95-9A-C9-D7
```

Switch(config-if)#switchport port-security mac –address E8-72-95-9A-B5-D7

```
Switch#
Switch#show port-security interface fa0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0005.5E57.A982:1
Security Violation Count   : 0
```

Now, only the defined mac addresses can access fa0/1, but what happens when any other user tries to connect to this port? For that, we have 2 options: either simply restricting the user or restricting plus turning the interface off.

Switch(config-if)#switchport port-security violation *restrict/shutdown*

One last thing, when assigning mac addresses, you need loads of time and effort. Instead, we can use mac-address sticky command.

```
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#switchport port-security maximum 2
Switch(config-if)#switchport port-security violation shutdown
```

Basically, with the sticky command you can provide static MAC address security without having to type in absolutely everyone's MAC address on the network. The first two MAC addresses coming into the port "stick" to it as static addresses and will be placed in the running-config, but when a third address tried to connect, the port would shut down immediately.

Note: As soon as you enable port-security on a port, it defaults to violation shutdown and a maximum of 1. You need to change the settings as you want them.

DHCP (Dynamic Host Configuration Protocol)

What is DHCP?

Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer from a defined range of numbers (i.e., a scope) configured for a given network.
DHCP assigns an IP address when a system is started, for example:

1. A user turns on a computer with a DHCP client.
2. The client computer sends a broadcast request (called a DISCOVER or DHCPDISCOVER), looking for a DHCP server to answer.
3. The router directs the DISCOVER packet to the correct DHCP server.
4. The server receives the DISCOVER packet. Based on availability and usage policies set on the server, the server determines an appropriate address (if any) to give to the client. The server then temporarily reserves that address for the client and sends back to the client an OFFER (or DHCPOFFER) packet, with that address information. The server also configures the client's DNS servers, WINS servers, NTP servers, and sometimes other services as well.
5. The client sends a REQUEST (or DHCPREQUEST) packet, letting the server know that it intends to use the address.
6. The server sends an ACK (or DHCPACK) packet, confirming that the client has a been given a lease on the address for a server-specified period of time.

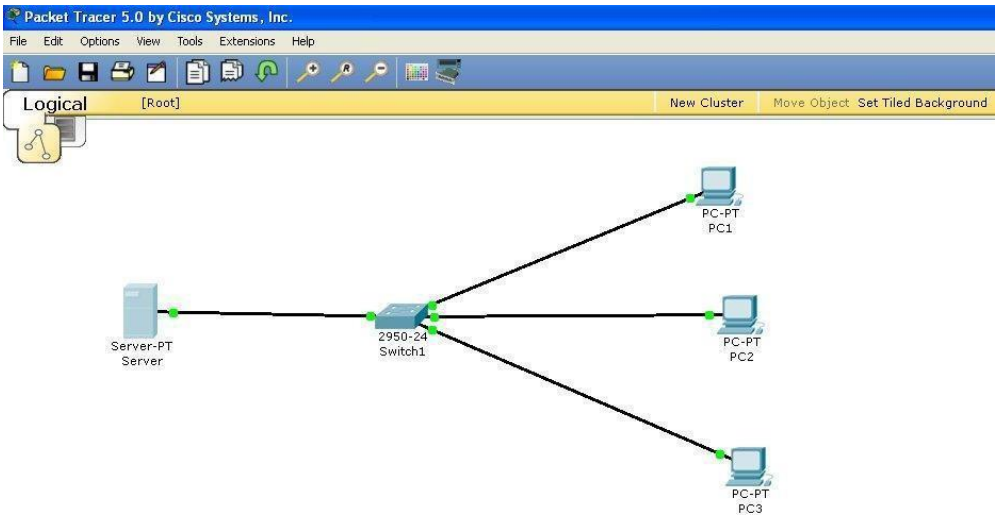
Static Addressing:

When a computer uses a static IP address, it means that the computer is manually configured to use a specific IP address.

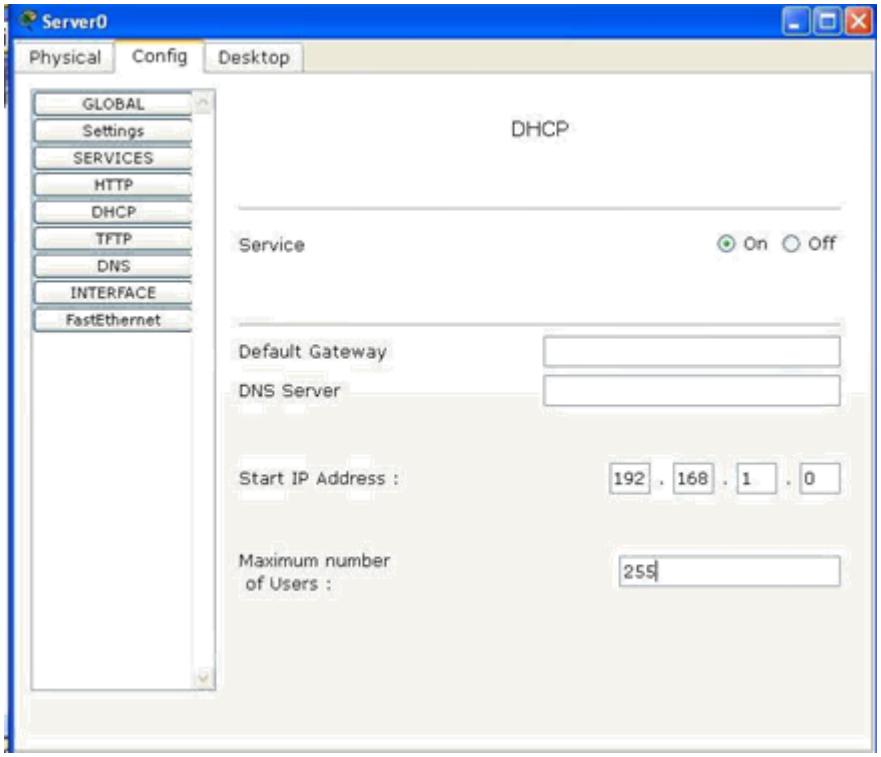
One problem with static assignment, which can result from user error or inattention to detail, occurs when two computers are configured with the same IP address. This creates a conflict that results in loss of service. Using DHCP to dynamically assign IP addresses minimizes these conflicts.

Configuring a Simple DHCP Server

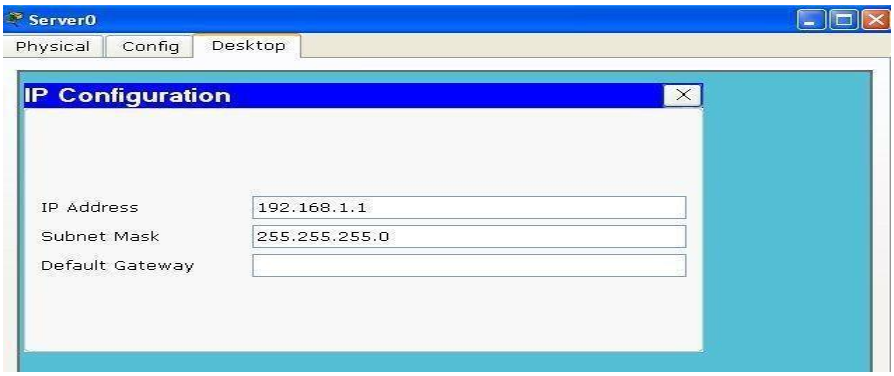
1. Open the Packet tracer
2. Draw a Network Diagram as shown below



3. Click on the Server PC and then Click on the Config tab. Enter the Starting IP address of your Network and the maximum number of nodes that you want on this Network.

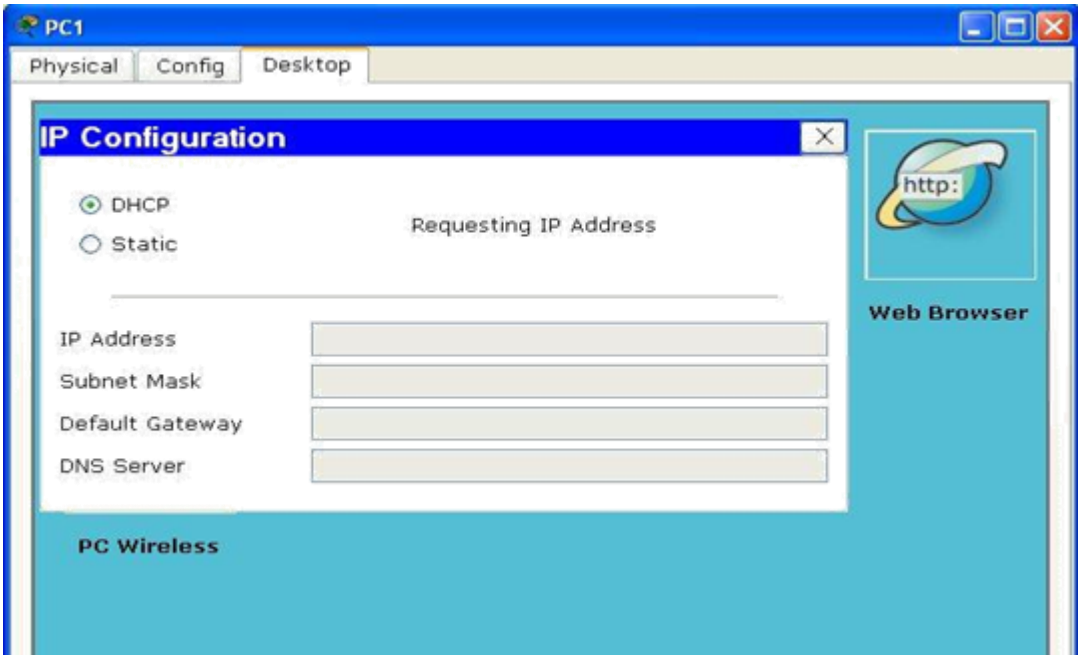


4. Also give the Server an IP Address



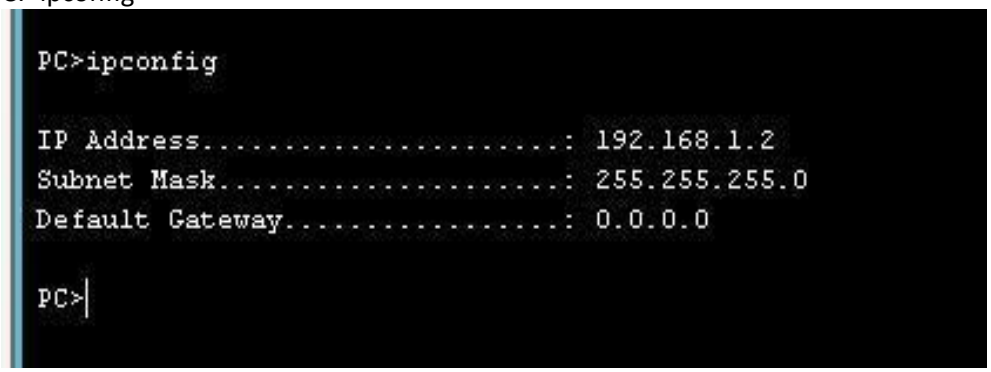
Client Configuration:

1. Click on a PC and then on Desktop Tab. Select DHCP as shown in the figure



2. Repeat this Step on each PC

To check the IP address of a PC, Go to its Command prompt and enter the following command.
C:>ipconfig



- **HTTP**

Hypertext Transfer Protocol (HTTP) is an application-layerprotocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers.

User's Browser (Client request) ⟷ Web Server (Server response)

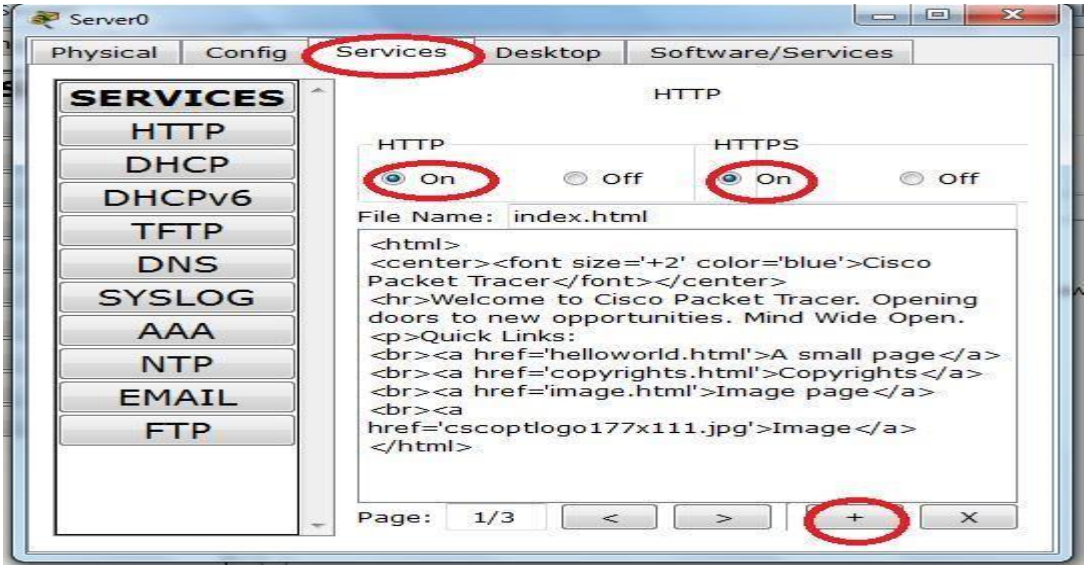
HTTP uses port 80, the port that web server accepts requests from.

- **HTTPS**

Secure HTTP ensures that the data passed between the browser and website is encrypted. (You can notice the padlock icon when browsing on HTTPS websites).

- **Configuring HTTP**
  1. Click on http Server, Select the desktop tab and give this server an IP address

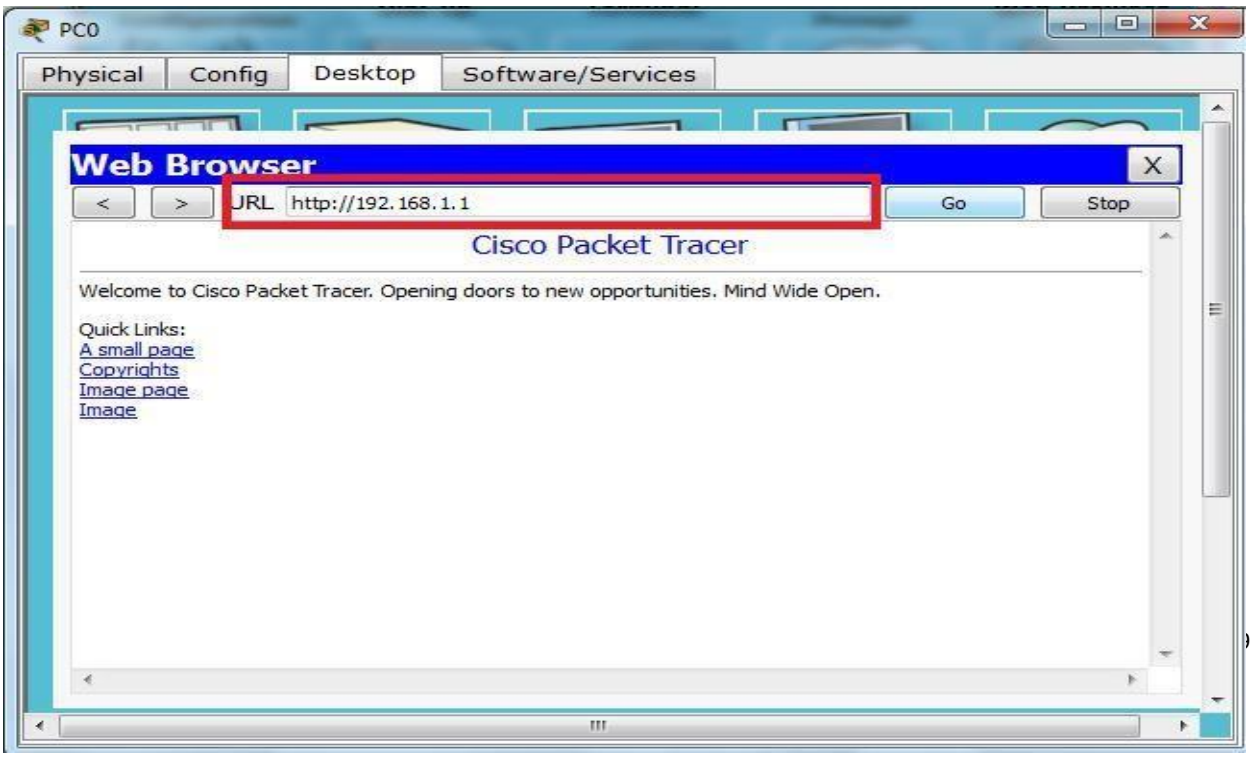
  2. After assigning the IP address to the server. Turn on the http service as shown in the following figure:



3. Turn off the remaining services such as FTP , DNS , DHCP etc for this server.

- **Access the http server**
  1. Assign static IP addresses to PCs.
  2. In the desktop tab, click on the web browser as shown in the following figure:

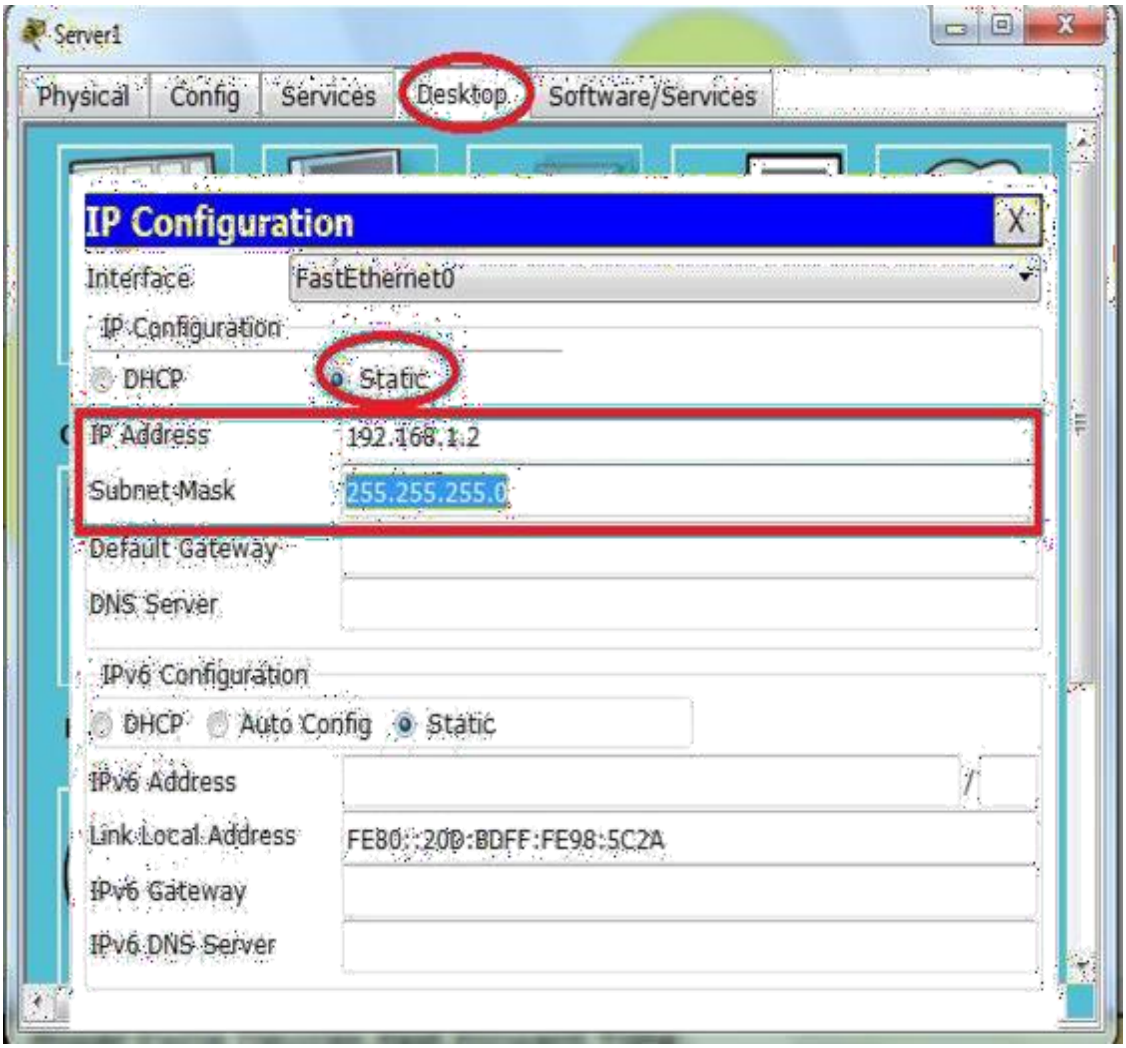
3. In the URL Type the ip address of the http server and click on Go, the web page will appear as shown in the following figure.



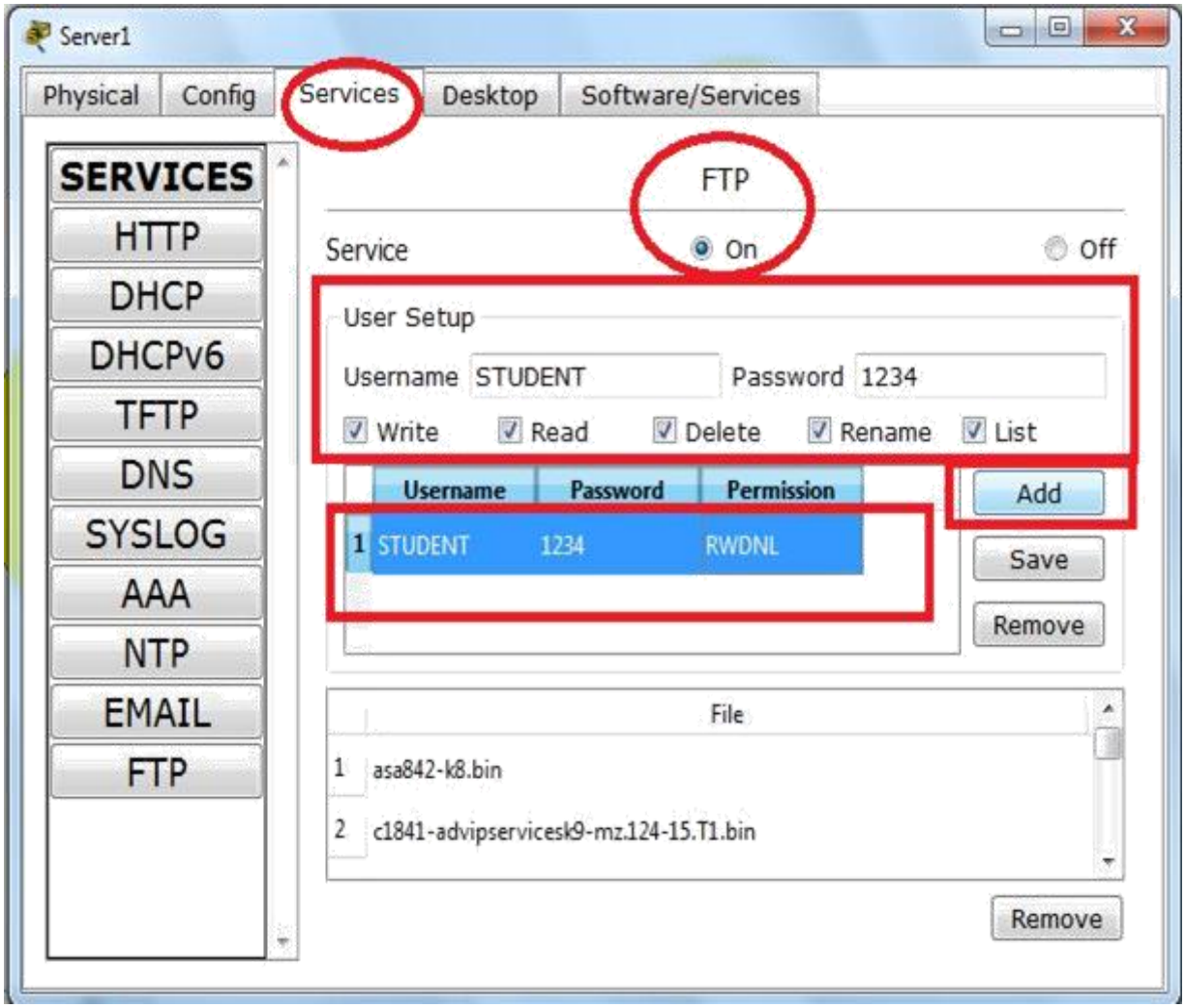
- **FTP**

File Transfer Protocol is network protocol for transferring files between computers. FTP works in a client-server model where an FTP server and FTP client perform the file transfer operation. An FTP server is configured in the network, and a specific file storage location (folder/system) is identified to become the shared storage, which will host the files you want to share. The end-users will access this file server via FTP to start copying the files to their local folder/system.

- **Configuring FTP**
  1. Click on FTP Server, Select the desktop tab and give this server an IP address



  2. After assigning the IP address to the server. Turn on the FTP service as shown in the following figure
  3. Set the user name and password
  4. Set the R/W permissions
  5. Add it

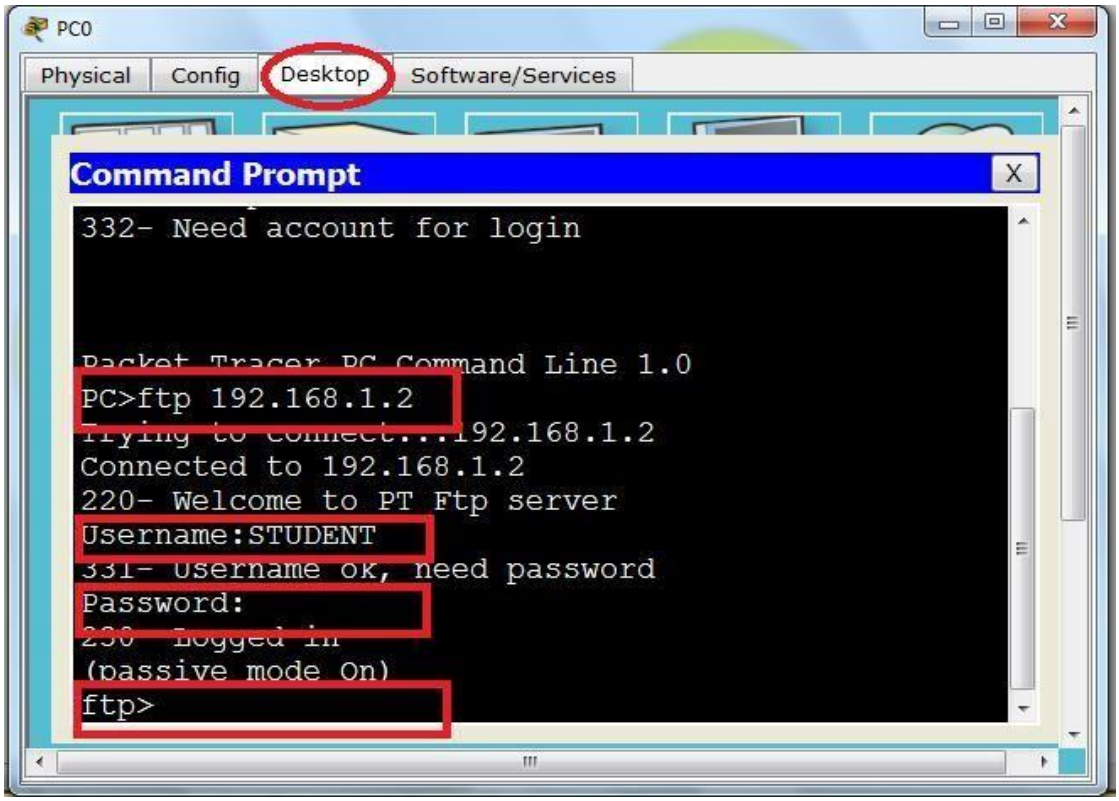


6. Turn off the remaining services such as HTTP, DNS , DHCP etc. for this server.

- **Access the FTP Server**

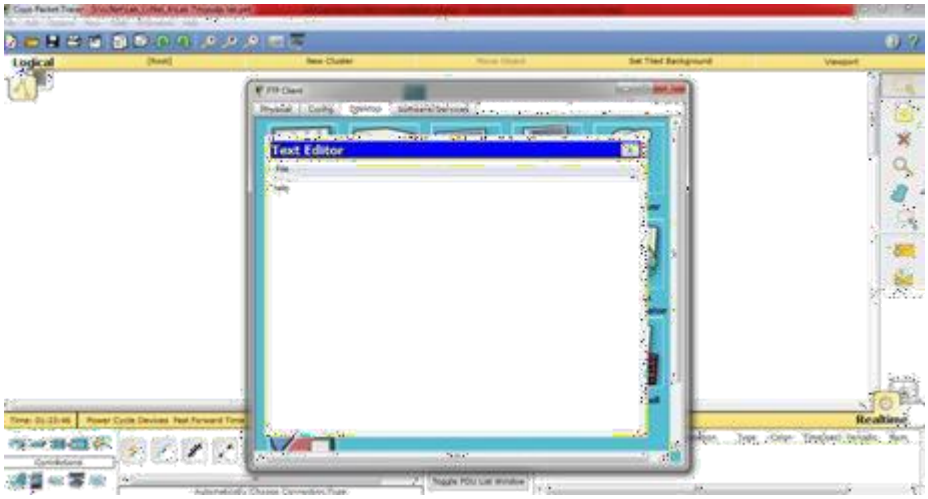
1. Under the desktop tab, click on the command prompt

2. Type ftp following by the ip address of the ftp server machine, as shown in the following figure:



- **Upload a file to FTP server**

  Click the Desktop tab of PC > Text Editor.

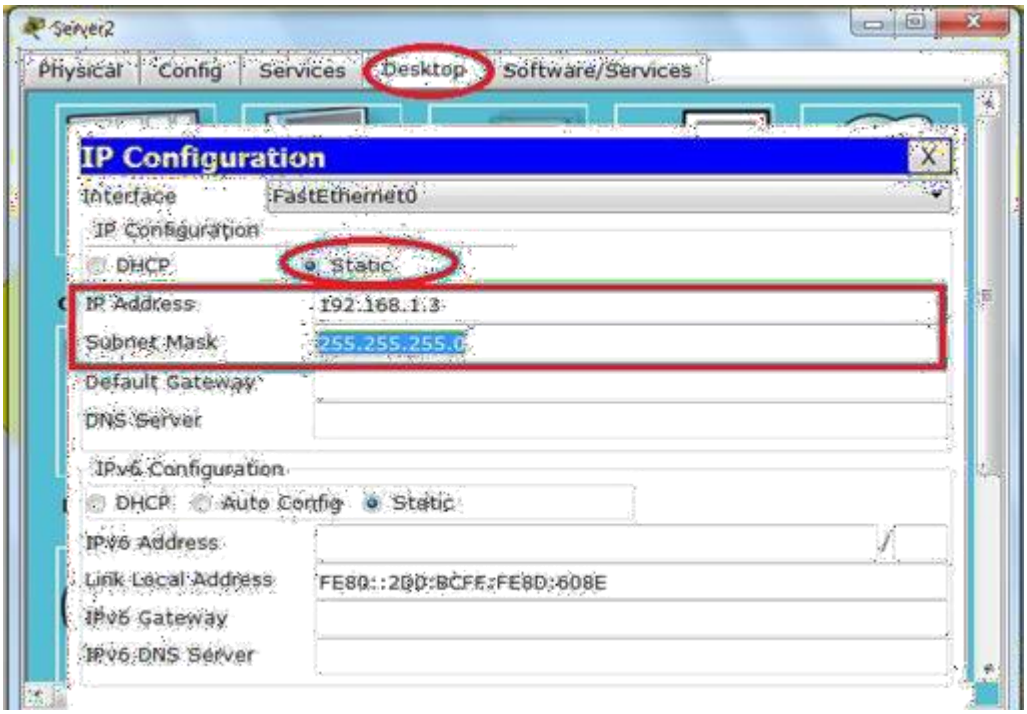


Save file by pressing Ctrl+s keys. Upload it on server by using command ftp> put file.txt

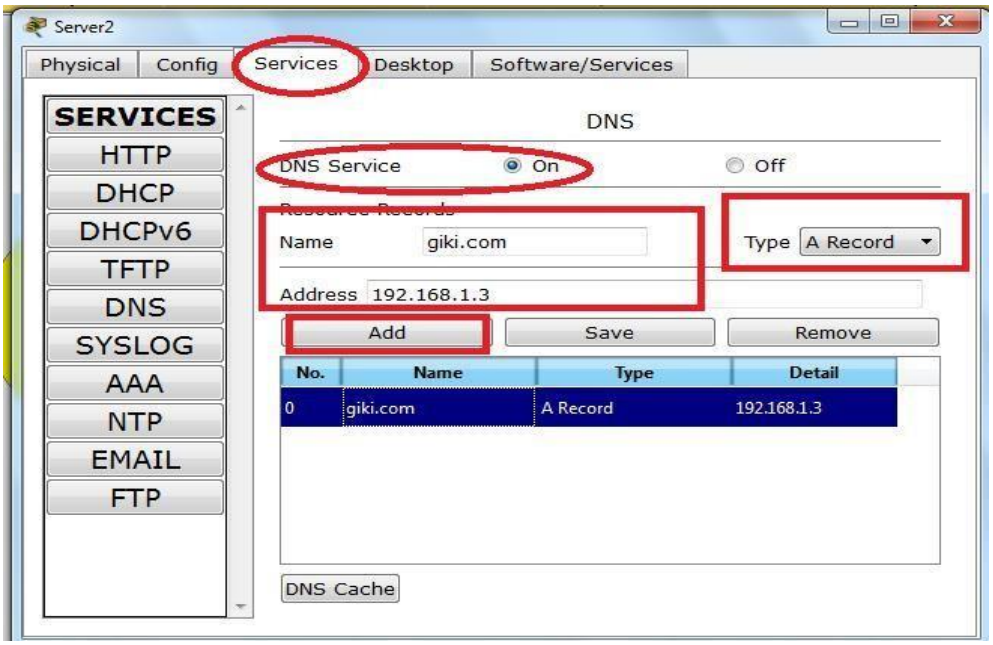
- **Download FTP file from server**
  Download the file.txt file: at the ftp> prompt, type get file.txt. The file.txt file is transferred to PC.
- **Delete FTP** ftp>
  delete file.txt

• Configuring DNS

1. Click on DNS Server, Select the desktop tab and give this server an IP address

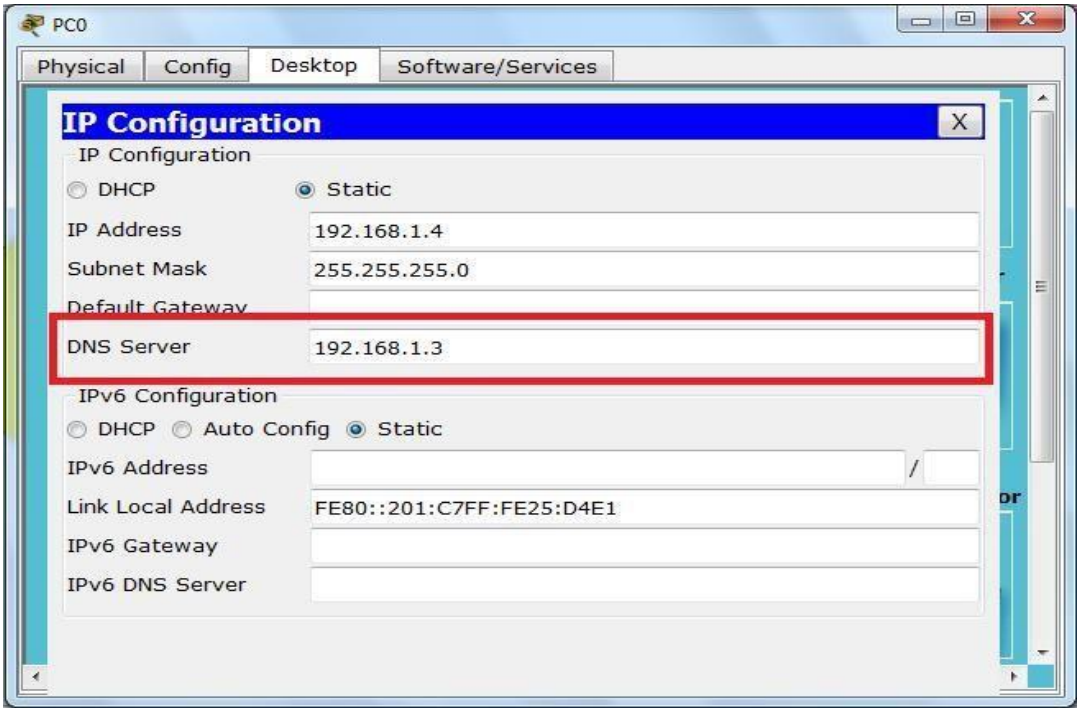


2. After assigning the IP address to the server. Turn on the DNS service as shown in the following figure



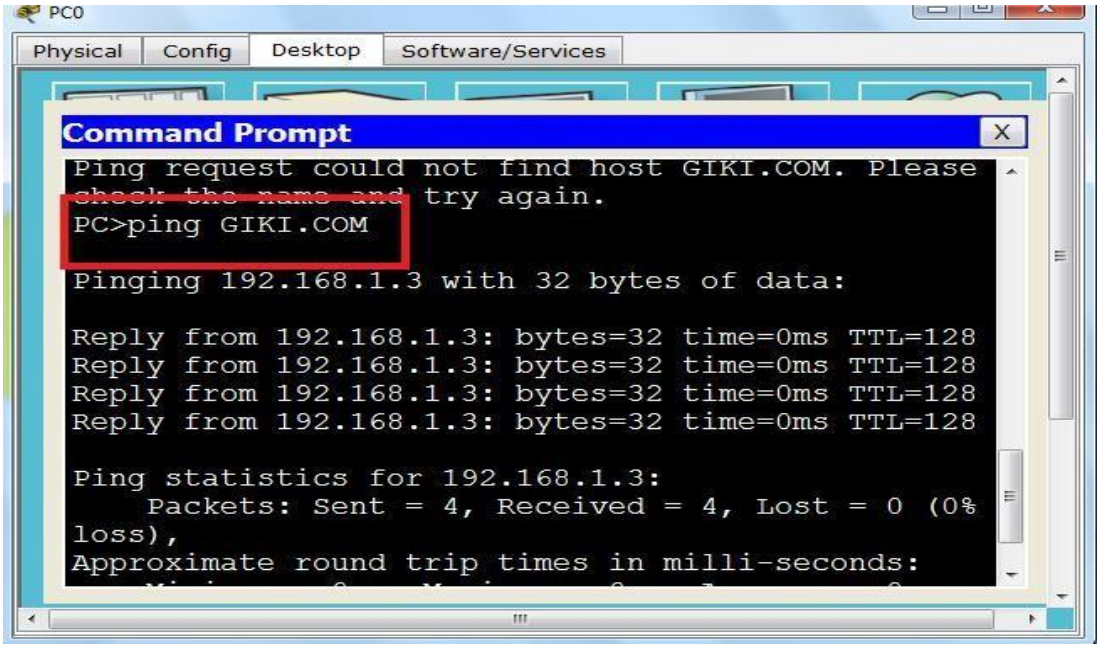
- **3 Accessing the DNS Server**

1. Type the IP Address of the DNS Server in the client pc as shown below;



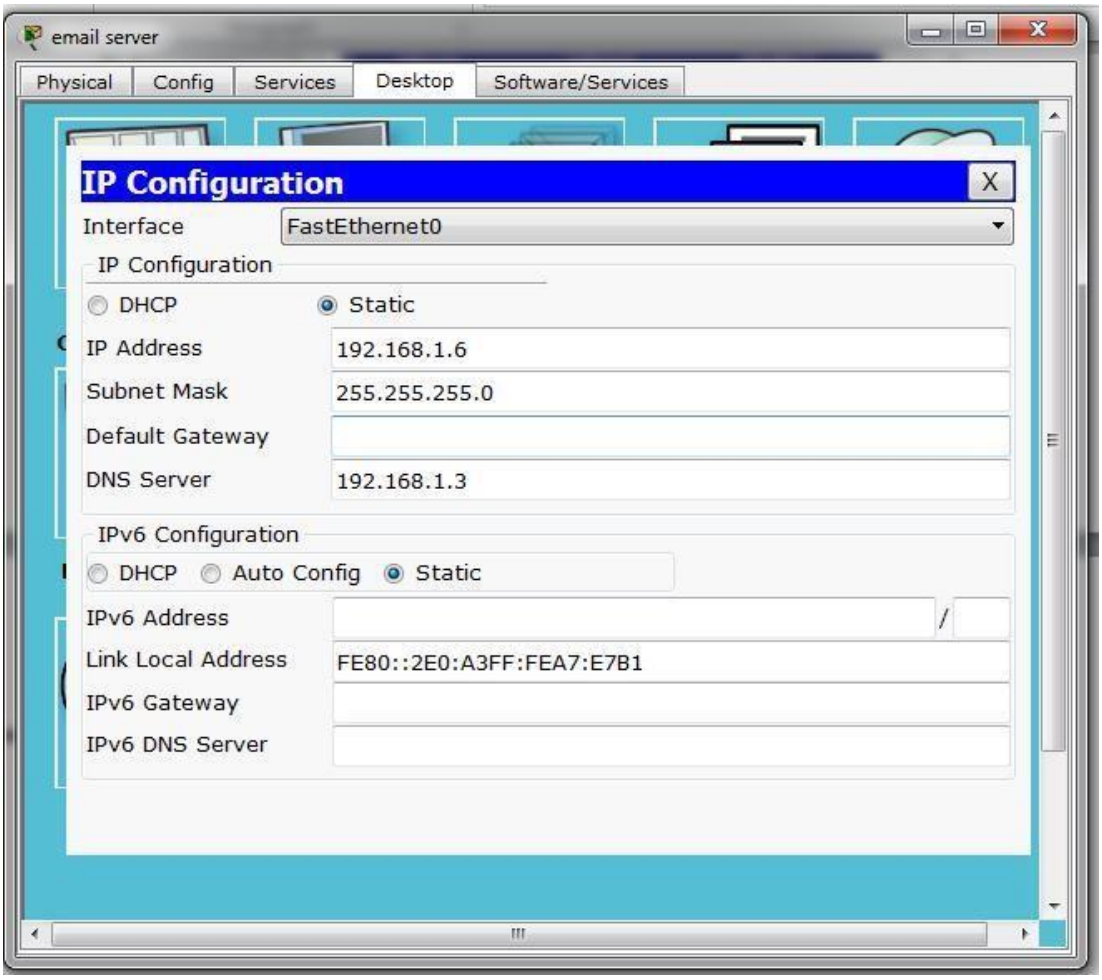
2. To access the giki.com domain click on the command prompt of the client pc as shown in the following figure

3. Type ping GIKI.COM

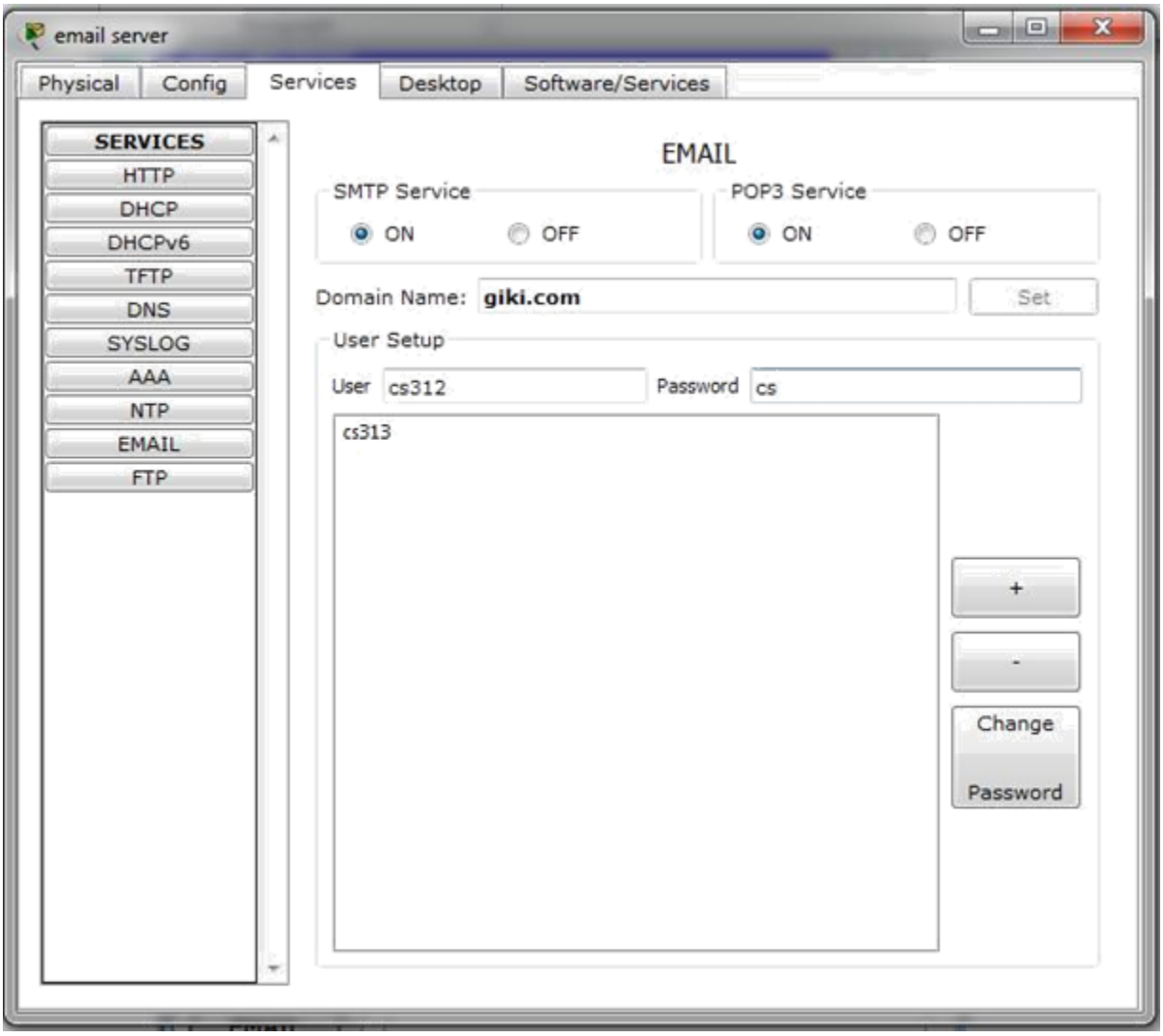


- **Configure Email Server**

Click on email Server, Select the desktop tab and give

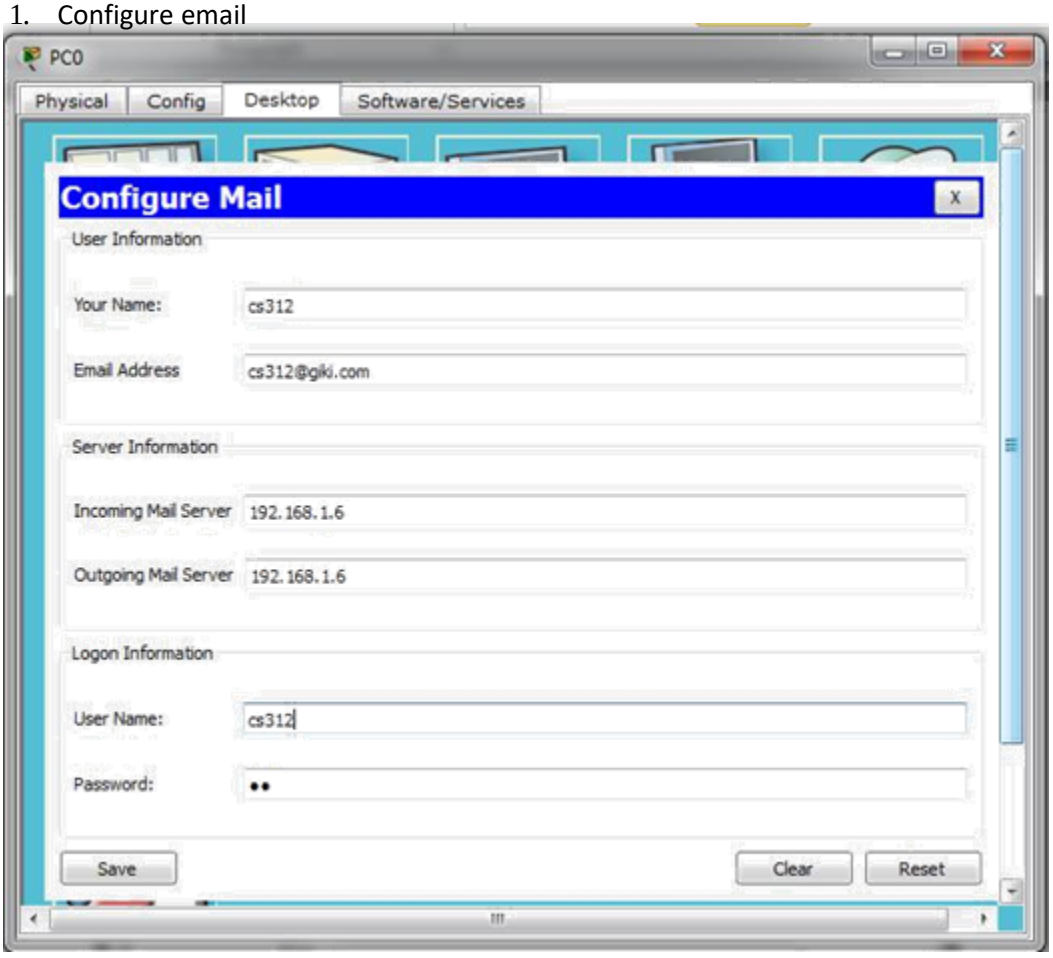


1. After assigning the IP address to the server. Turn on the email service and set domain name, user and password as shown in the following figure



- **4 Accessing the Email Account**

1. Configure email



- <u>Compose email</u>

a. Click **E-Mail Client** and click the **Desktop** tab > **E Mail** tool.

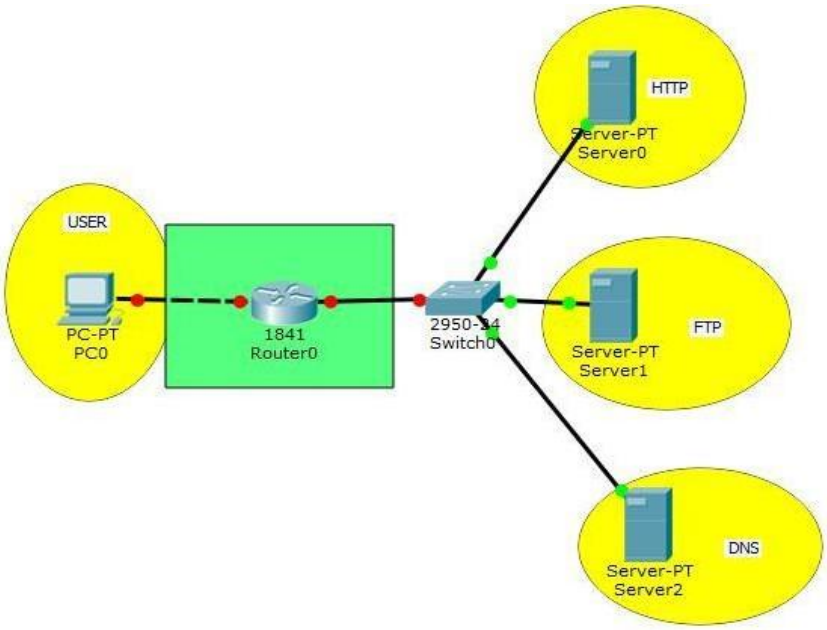
b. Click **Compose** and enter the following information:

1) **To:** cs313@giki.com

2) **Subject:** Personalize the subject line

3) **E-Mail Body:** Personalize the Email c. Click **Send**.

LAB EXERCISE



- Now your servers are on a different network from the addition of a router, access the servers from the client pc.
- What will happen when you turn off POP3 and SMTP services?
- Download, upload and delete a file from FTP server.

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 04

Class full IP Addressing, CIDR and Subnetting

Learning Objectives:

- Understanding IP Addresses
- Network Masks
- Understanding Subnetting
- Examples
- VLSM
- CIDR
- Appendix Sample Config Host/Subnet Quantities Table

Introduction

This document gives you basic information needed in order to configure your router for routing IP, such as how addresses are broken down and how subnetting works. You learn how to assign each interface on the router an IP address with a unique subnet. There are many examples to help tie everything together.

Requirements

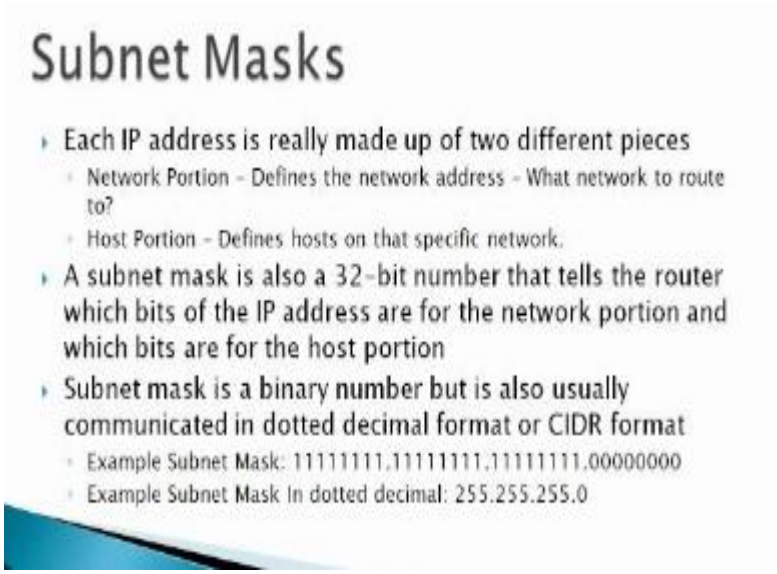
This Lab recommends that you have knowledge of these topics:

- Basic understanding of binary and decimal numbers.

Additional Information

If definitions are helpful to you, use these vocabulary terms to get you started:

- **Address** The unique number ID assigned to one host or interface in a network.
- **Subnet** A portion of a network sharing a particular subnet address.
- **Subnet mask** A 32−bit combination used to describe which portion of an address refers to the Subnet and which part refers to the host
- **Interface** A network connection.



If you have already received your legitimate address (es) from the Internet Network Information Center Inter NIC), you are ready to begin. If you do not plan to connect to the Internet, Cisco strongly suggests that you use reserved addresses from RFC 1918.

Understanding IP Addresses

An IP address is an address used in order to uniquely identify a device on an IP network. The address is made up of 32 binary bits, which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 – 11111111 binary.

Here is how binary octets convert to decimal: The right most bit, or least significant bit, of an octet hold a value of 2^0. The bit just to the left of that holds a value of 2^1. This continues until the left–most

bit, or most significant bit, which holds a value of 2^7. So if all binary bits are a one, the decimal equivalent would be 255 as shown here:

```
  1   1   1   1 1 1 1 1
128 64  32  16  8 4 2 1 (128+64+32+16+8+4+2+1=255)
```

Here is a sample octet conversion when not all of the bits are set to 1.

```
0  1 0 0 0 0 0 1
0 64 0 0 0 0 0 1 (0+64+0+0+0+0+0+1=65)
```

And this is sample shows an IP address represented in both binary and decimal.

10. 1. 23. 19 (decimal)
00001010.00000001.00010111.00010011 (binary)

These octets are broken down to provide an addressing scheme that can accommodate large and small networks. There are five different classes of networks, A to E. This document focuses on addressing classes A to C, s ince classes D and E are reserved and discussion of them is beyond the scope of this document.

Note: Also note that the terms "Class A, Class B" and so on are used in this document to help facilitate the understanding of IP addressing and subnetting. These terms are rarely used in the industry anymore because of the introduction of classless interdomain routing (CIDR).

Given an IP address, its class can be determined from the three high−order bits. Figure 1 shows the significance in the three high order bits and the range of addresses that fall into each class. For informational purposes, Class D and Class E addresses are also shown.

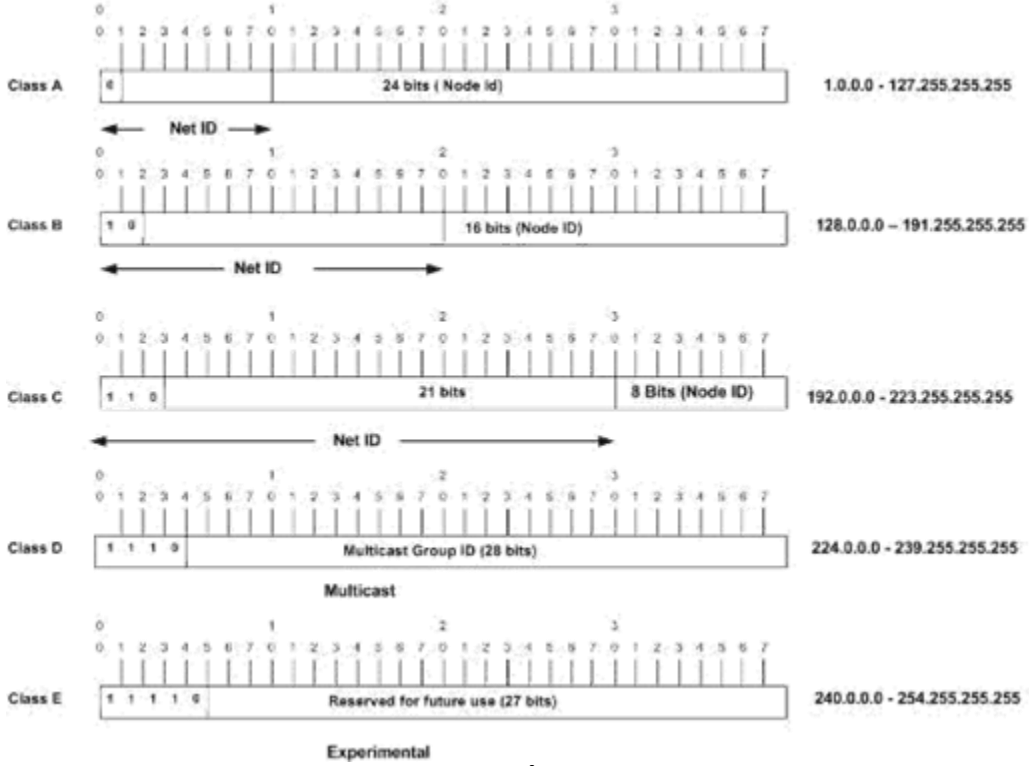


Figure 1:

In a Class A address, the first octet is the network portion, so the Class A example in Figure 1 has a major network address of 1.0.0.0 − 127.255.255.255. Octets 2, 3, and 4 (the next 24 bits) are for the network manager to divide into subnets and hosts as he/she sees fit. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16777214 hosts!).

In a Class B address, the first two octets are the network portion, so the Class B example in Figure 1 has a major network address of 128.0.0.0 − 191.255.255.255. Octets 3 and 4 (16 bits) are for local subnets and hosts. Class B addresses are used for networks that have between 256 and 65534 hosts.

In a Class C address, the first three octets are the network portion. The Class C example in Figure 1 has a major network address of 192.0.0.0 − 233.255.255.255. Octet 4 (8 bits) is for local subnets and hosts − perfect for networks with less than 254 hosts.

Network Masks

A network mask helps you know which portion of the address identifies the network and which portion of the address identifies the node. Class A, B, and C networks have default masks, also known as natural masks, as shown here:

Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0

An IP address on a Class A network that has not been subnetted would have an address/mask pair similar to: 8.20.15.1 255.0.0.0. To see how the mask helps you identify the network and node parts of the address, convert the address and mask to binary numbers.

8.20.15.1 = 00001000.00010100.00001111.00000001
255.0.0.0 = 11111111.00000000.00000000.00000000

Once you have the address and the mask represented in binary, then identifying the network and host ID is easier. Any address bits which have corresponding mask bits set to 1 represent the network ID. Any address bits that have corresponding mask bits set to 0 represent the node ID.

```
8.20.15.1   =  00001000.00010100.00001111.00000001  255.0.0.0   =
                                                    11111111.00000000.00000000.00000000
                                                    ___________________________________
                          Net id  |                          host id

netid =          00001000 = 8
hostid = 00010100.00001111.00000001 = 20.15.1
```

Understanding Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique

network/subnetwork ID. Any device, or gateway, connecting *n* networks/subnetworks has *n* distinct IP addresses, one for each network / subnetwork that it interconnects.

In order to subnet a network, extend the natural mask using some of the bits from the host ID portion of the address to create a subnetwork ID. For example, given a Class C network of 204.17.5.0 which has a natural mask of 255.255.255.0, you can create subnets in this manner:

```
204.17.5.0    -  11001100.00010001.00000101.00000000
255.255.255.224  -  11111111.11111111.11111111.11100000
                              ----------------------|sub|----
```

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device *since host ids of all zeros or all ones are not allowed* (it is very important to remember this). So, with this in mind, these subnets have been created.

204.17.5.0	255.255.255.224	host address range	1 to 30
204.17.5.32	255.255.255.224	host address range	33 to 62
204.17.5.64	255.255.255.224	host address range	65 to 94
204.17.5.96	255.255.255.224	host address range	97 to 126
204.17.5.128	255.255.255.224	host address range 129	to 158
204.17.5.160	255.255.255.224	host address range 161	to 190
204.17.5.192	255.255.255.224	host address range	193 to 222
204.17.5.224	255.255.255.224	host address range	225 to 254

Note: There are two ways to denote these masks. First, since you are using three bits more than the "natural" Class C mask, you can denote these addresses as having a 3-bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. This second method is used with CIDR. With this method, one of these networks can be described with the notation prefix/length. For example, 204.17.5.32/27 denotes the network 204.17.5.32 255.255.255.224. When appropriate the prefix/length notation is used to denote the mask throughout the rest of this document. The network subnetting scheme in this section allows for eight subnets, and the network might appear as:

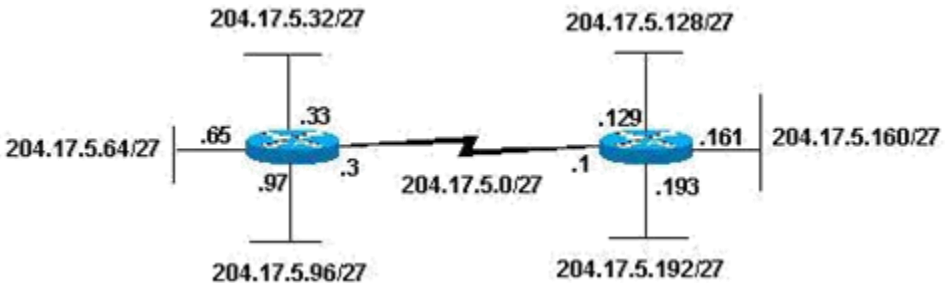


Figure 2:

Notice that each of the routers in Figure 2 is attached to four subnetworks, one subnetwork is common to both routers. Also, each router has an IP address for each subnetwork to which it is attached. Each subnetwork could potentially support up to 30 host addresses.

This brings up an interesting point. The more host bits you use for a subnet mask, the more subnets you have available. However, the more subnets available, the less host addresses available per subnet. For example, a Class C network of 204.17.5.0 and a mask of 255.255.255.224 (/27) allows you to have eight subnets, each with 32 host addresses (30 of which could be assigned to devices). If you use a mask of 255.255.255.240 (/28), the break down is:

```
204.17.5.0   -  11001100.00010001.00000101.00000000
255.255.255.240  -  11111111.11111111.11111111.11110000
                              ----------|sub |---
```

Since you now have four bits to make subnets with, you only have four bits left for host addresses. So in this case you can have up to 16 subnets, each of which can have up to 16 host addresses (14 of which can be assigned to devices).

Take a look at how a Class B network might be subnetted. If you have network 172.16.0.0 , then you know that its natural mask is 255.255.0.0 or 172.16.0.0/16. Extending the mask to anything beyond 255.255.0.0 means you are subnetting. You can quickly see that you have the ability to create a lot more subnets than with the Class C network. If you use a mask of 255.255.248.0 (/21), how many subnets and hosts per subnet does this allow for?

```
172.16.0.0   -  10101100.00010000.00000000.00000000
255.255.248.0   -  11111111.11111111.11111000.00000000
                          ----------| sub |----------
```

You are using five bits from the original host bits for subnets. This allows you to have 32 subnets (2^5). After using the five bits for subnetting, you are left with 11 bits for host addresses. This allows eachsubnet so have 2048 host addresses (2^{11}), 2046 of which could be assigned to devices.

Note: In the past, there were limitations to the use of a subnet 0 (all subnet bits are set to zero) and all ones subnet (all subnet bits set to one). Some devices would not allow the use of these subnets. Cisco Systems devices allow the use of these subnets when the**ip subnet zero** command is configured.

Examples

Sample Exercise 1

Now that you have an understanding of subnetting, put this knowledge to use. In this example, you are given two address / mask combinations, written with the prefix/length notation, which have been assigned to two devices. Your task is to determine if these devices are on the same subnet or different subnets. You can do this by using the address and mask of each device to determine to which subnet each address belongs.

DeviceA: 172.16.17.30/20
DeviceB: 172.16.28.15/20

Determining the Subnet for DeviceA:

```
172.16.17.30 – 10101100.00010000.00010001.00011110
255.255.240.0 – 11111111.11111111.11110000.00000000
                          ––––––––––––––––––| sub|––––––––––––
subnet =                  10101100.00010000.00010000.00000000 = 172.16.16.0
```

Looking at the address bits that have a corresponding mask bit set to one, and setting all the other address bits to zero (this is equivalent to performing a logical "AND" between the mask and address), shows you to which subnet this address belongs. In this case, DeviceA belongs to subnet 172.16.16.0.

Determining the Subnet for DeviceB:

```
172.16.28.15 – 10101100.00010000.00011100.00001111
255.255.240.0 – 11111111.11111111.11110000.00000000
                          ––––––––––––––––––| sub|––––––––––––
subnet =                  10101100.00010000.00010000.00000000 = 172.16.16.0
```

From these determinations, DeviceA and DeviceB have addresses that are part of the same subnet.

Sample Exercise 2

Given the Class C network of 204.15.5.0/24, subnet the network in order to create the network in Figure 3 with the host requirements shown.

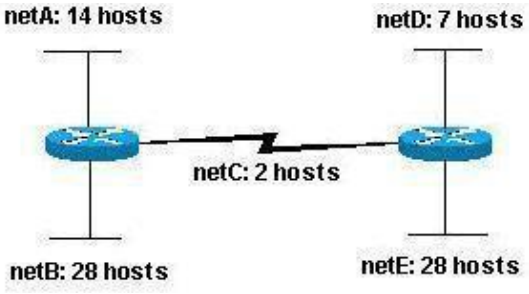


Figure 3

Looking at the network shown in Figure 3, you can see that you are required to create five subnets. The largest subnet must support 28 host addresses. Is this possible with a Class C network? and if so, then how?

You can start by looking at the subnet requirement. In order to create the five needed subnets you would need to use three bits from the Class C host bits. Two bits would only allow you four subnets (2^2).

Since you need three subnet bits that leaves you with five bits for the host portion of the address. How many hosts does this support? $2^5 = 32$ (30 usable). This meets the requirement.

Therefore you have determined that it is possible to create this network with a Class C network. An example of how you might assign the subnetworks is:

netA: 204.15.5.0/27 host address range 1 to 30 netB: 204.15.5.32/27 host address range 33 to 62 netC: 204.15.5.64/27 host address range 65 to 94

netD: 204.15.5.96/27 host address range 97 to 126
netE: 204.15.5.128/27 host address range 129 to 158

VLSM Example

In all of the previous examples of subnetting, notice that the same subnet mask was applied for all the subnets. This means that each subnet has the same number of available host addresses. You can need this in some cases, but, in most cases, having the same subnet mask for all subnets ends up wasting address space. For example, in the Sample Exercise 2 section, a class C network was split into eight equal−size subnets; however, each subnet did not utilize all available host addresses, which results in wasted address space. Figure 4 illustrates this wasted address space.

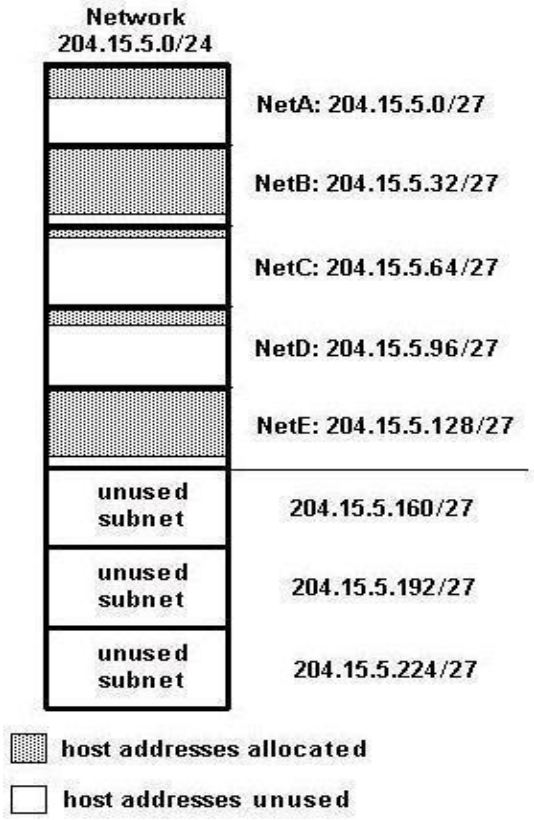


Figure 4

Figure 4 illustrates that of the subnets that are being used, NetA, NetC, and NetD have a lot of unused host address space. It is possible that this was a deliberate design accounting for future growth, but in many cases this is just wasted address space due to the fact that the same subnet mask is being used for all the subnets. Variable Length Subnet Masks (VLSM) allows you to use different masks for each subnet, thereby using address space efficiently.

VLSM Example

Given the same network and requirements as in Sample Exercise 2 develop a subnetting scheme with the use of VLSM, given:

netA: must support 14 hosts
netB: must support 28 hosts
netC: must support 2 hosts
netD: must support 7 hosts
netE: must support 28 host

Determine what mask allows the required number of hosts.

netA: requires a /28 (255.255.255.240) mask to support 14 hosts
netB: requires a /27 (255.255.255.224) mask to support 28 hosts
netC: requires a /30 (255.255.255.252) mask to support 2 hosts
netD*: requires a /28 (255.255.255.240) mask to support 7 hosts
netE: requires a /27 (255.255.255.224) mask to support 28 hosts

* a /29 (255.255.255.248) would only allow 6 usable host addresses therefore netD requires a /28 mask.

The easiest way to assign the subnets is to assign the largest first. For example, you can assign in this manner:

netB: 204.15.5.0/27 host address range 1 to 30
netE: 204.15.5.32/27 host address range 33 to 62
netA: 204.15.5.64/28 host address range 65 to 78
netD: 204.15.5.80/28 host address range 81 to 94
netC: 204.15.5.96/30 host address range 97 to 98

This can be graphically represented as shown in Figure 5:

Figure 5

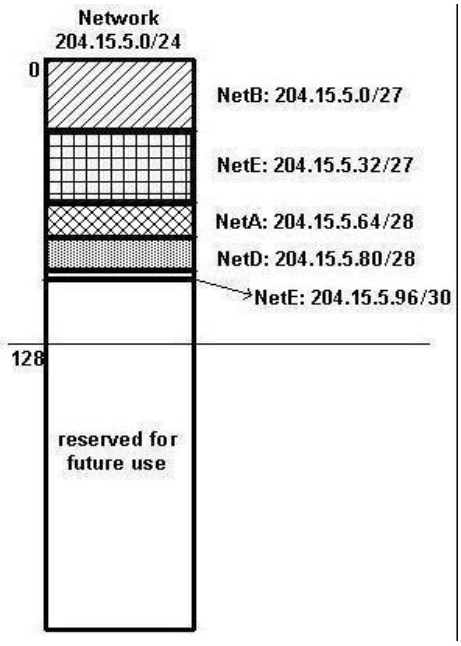


Figure 5 illustrates how using VLSM helped save more than half of the address space.

CIDR

Classless Interdomain Routing (CIDR) was introduced to improve both address space utilization and routing scalability in the Internet. It was needed because of the rapid growth of the Internet and growth of the IP routing tables held in the Internet routers.

CIDR moves way from the traditional IP classes (Class A, Class B, Class C, and so on). In CIDR , an IP network is represented by a prefix, which is an IP address and some indication of the length of the mask. Length means the number of left−most contiguous mask bits that are set to one. So network 172.16.0.0 255.255.0.0 can be represented as 172.16.0.0/16. CIDR also depicts a more hierarchical Internet architecture, where each domain takes its IP addresses from a higher level. This allows for the summarization of the domains to be done at the higher level. For example, if an ISP owns network 172.16.0.0/16, then the ISP can offer 172.16.1.0/24, 172.16.2.0/24, and so on to customers. Yet, when advertising to other providers, the ISP only needs to advertise 172.16.0.0/16.

Sample Config

Routers A and B are connected via serial interface.

Router A

```
hostname routera
!
ip routing
!
int e 0
ip address 172.16.50.1
255.255.255.0 !(subnet 50)
int e 1 ip address 172.16.55.1
255.255.255.0 !(subnet 55)
int t 0 ip address 172.16.60.1
255.255.255.0 !(subnet 60) int s 0
ip address 172.16.65.1 255.255.255.0 (subnet
65)    !S 0 connects to router B
router rip network
172.16.0.0
```

Router B

```
hostname routerb
!
ip routing
!
int e 0
ip address 192.1.10.200
255.255.255.240 !(subnet 192)
int e 1
ip address 192.1.10.66
255.255.255.240 !(subnet 64)
```

int s 0
ip address 172.16.65.2 (same subnet as router A's s
0) !Int s 0 connects to router A
router rip network
192.1.10.0
network 172.16.0.0

Host/Subnet Quantities Table

Class B # bits	Mask	Effectiv e Subnets	Effective Hosts
1	255.255.128.0	2	32766
2	255.255.192.0	4	16382
3	255.255.224.0	8	8190
4	255.255.240.0	16	4094
5	255.255.248.0	32	2046
6	255.255.252.0	64	1022
7	255.255.254.0	128	510
8	255.255.255.0	256	254
9	255.255.255.128	512	126
10	255.255.255.192	1024	62
11	255.255.255.224	2048	30
12	255.255.255.240	4096	14
13	255.255.255.248	8192	6
14	255.255.255.252	16384	2

Class C # bits	Mask	Effectiv e Subnets	Effective Hosts
1	255.255.255.128	2	126
2	255.255.255.192	4	62
3	255.255.255.224	8	30
4	255.255.255.240	16	14
5	255.255.255.248	32	6
6	255.255.255.252	64	2

*Subnet all zeroes and all ones included. These might not be supported on some legacy systems.
*Hos t all zeroes and all ones excluded.

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 05

Basic VLAN Configuration & Inter-Vlan Routing

Upon Completion of this Lab you will be able to learn:

- Cable a network according to the topology diagram
- Erase the startup configuration and reload a switch to the default state
- Perform basic configuration tasks on a switch
- Create VLANs
- Assign switch ports to a VLAN
- Add, move, and change ports
- Verify VLAN configuration
- Enable trunking on inter-switch connections
- Verify trunk configuration
- Save the VLAN configuration
- InterVlan Routing
- Switch Configuration for InterVlan Routing
- Introducing Router on stick for InterVlan Routing
- Trunk and Access Modes configuration of End Devices
- Checking the Broadcast Domain in InterVlans

What is a VLAN?

VLAN is a Virtual Local Area Network. In technical terms, a VLAN is a broadcast domain created by switches. Normally, it is a router creating that broadcast domain. With VLAN's, a switch can create the broadcast domain.

Because switches can talk to each other, some ports on switch A can be in VLAN 10 and other ports on switch B can be in VLAN 10. Broadcasts between these devices will not be seen on any other port in any other VLAN, other than 10. However, these devices can all communicate because they are on the same VLAN. Without additional configuration, they would not be able to communicate with any other devices, not in their VLAN. Devices on different VLAN's can communicate with a router or a Layer 3 switch.

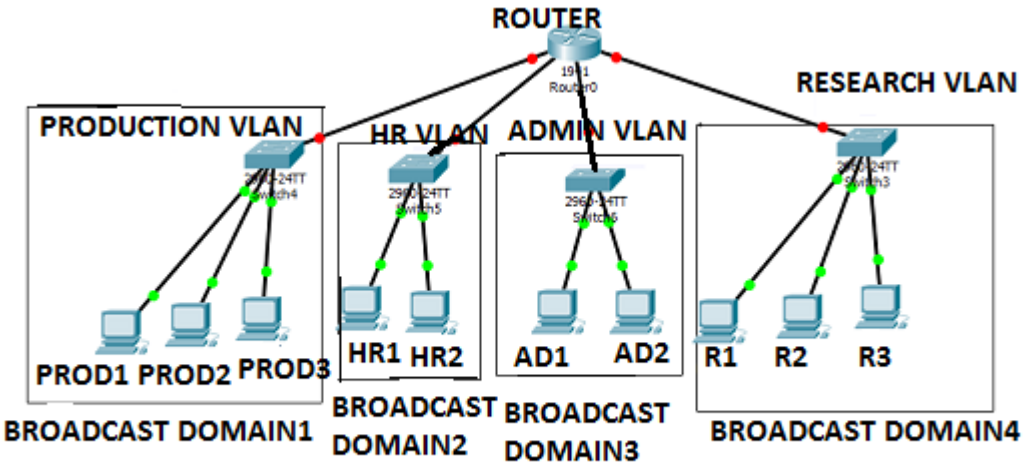
Configuring VLAN's can vary even between different models of Cisco switches. Your goals, no matter what the commands are, is to:

- *Create the new VLAN's*
- *Put each port in the proper VLAN*

Let's say we wanted to create VLAN's 5 and 10. We want to put ports 2 & 3 in VLAN 5 (Marketing) and ports 4 and 5 in VLAN 10 (Human Resources). On a Cisco 2950 switch, here is how you would do it:

At this point, only ports 2 and 3 should be able to communicate with each other and ports 4 & 5 should be able to communicate. That is because each of these is in its own VLAN. For the device on port 2 to

communicate with the device on port 4, you would have to configure a trunk port to a router / Layer 3



switch so that it can strip off the VLAN information, route the packet, and add back the VLAN information.

Addressing Table

Device (Hostname)	Interface	IP Address	Subnet Mask	Default Gateway
S1	VLAN 99	172.17.99.11	255.255.255.0	N/A
S2	VLAN 99	172.17.99.12	255.255.255.0	N/A
S3	VLAN 99	172.17.99.13	255.255.255.0	N/A
PC1	NIC	172.17.10.21	255.255.255.0	172.17.10.1
PC2	NIC	172.17.20.22	255.255.255.0	172.17.20.1
PC3	NIC	172.17.30.23	255.255.255.0	172.17.30.1
PC4	NIC	172.17.10.24	255.255.255.0	172.17.10.1
PC5	NIC	172.17.20.25	255.255.255.0	172.17.20.1
PC6	NIC	172.17.30.26	255.255.255.0	172.17.30.1

Initial Ports Assignments (switch 2 and switch 3)

Ports	Assignment	Network
Fa0/1 – 0/5	802.1q Trunks (Native VLAN 99)	172.17.99.0 /24
Fa0/6 – 0/10	VLAN 30 – Guest (Default)	172.17.30.0 /24
Fa0/11 – 0/17	VLAN 10 – Faculty/Staff	172.17.10.0 /24
Fa0/18 – 0/24	VLAN 20 – Students	172.17.20.0 /24

Task 1: Prepare the Network

Step 1: Cable a network that is similar to the one in the topology diagram.

You can use any current switch in your lab as long as it has the required interfaces shown in the topology. Note: If you use 2900 or 2950 switches, the outputs may appear different. Also, certain commands may be different or unavailable.

Step 2: Clear any existing configurations on the switches, and initialize all ports in the shutdown state.

It is a good practice to disable any unused ports on the switches by putting them in shutdown. Disable all ports on the switches:

Switch#config term
Switch(config)#interface range fa0/1-24
Switch(config-if-range)#shutdown
Switch(config-if-range)#interface range gi0/1-2
Switch(config-if-range)#shutdown

Task 2: Perform Basic Switch Configurations

Step 1: Configure the switches according to the following guidelines.

- Configure the switch hostname.
- Disable DNS lookup.
- Configure an EXEC mode password of **class**.
- Configure a password of **cisco** for console connections.

• Configure a password of **cisco** for vty connections.

Step 2: Re-enable the user ports on S2 and S3.

S2(config)#interface range fa0/6, fa0/11, fa0/18
S2(config-if-range)#switchport mode access
S2(config-if-range)#no shutdown
S3(config)#interface range fa0/6, fa0/11, fa0/18
S3(config-if-range)#switchport mode access
S3(config-if-range)#no shutdown

Task 3: Configure and Activate Ethernet Interfaces

Step 1: Configure the PCs. You can complete this lab using only two PCs by simply changing the IP addressing for the two PCs specific to a test you want to conduct. For ex ample, if you want to test connectivity between PC1 and PC2, then configure the IP addresses for those PCs by referring to the addressing table at the beginning of the lab. Alternatively, you can configure all six PCs with the IP addresses and default gateways.

Task 4: Configure VLANs on the Switch

Step 1: Create VLANs on switch S1.

Use the **vlan** *vlan-id* command in global configuration mode to add a VLAN to switch S1. There are four VLANS configured for this lab: VLAN 10 (faculty/staff); VLAN 20 (students); VLAN 30 (guest); and VLAN 99 (management). After you create the VLAN, you will be in vlan configuration mode, where you can assign a name to the VLAN with the **name** *vlan name* command. S1(config)#**vlan 10**
S1(config-vlan)#**name faculty/staff**

S1(config-vlan)#vlan 20
S1(config-vlan)#name students
S1(config-vlan)#vlan 30
S1(config-vlan)#name guest
S1(config-vlan)#vlan 99
S1(config-vlan)#name management
S1(config-vlan)#end
S1#

Step 2: Verify that the VLANs have been created on S1.

Use the **show vlan brief** command to verify that the VLANs have been created. S1#**show vlan brief**

```
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24, Gi0/1
                                                 Gi0/2
10   faculty/staff                     active
20   students                          active
30   guest                             active
99   management                        active
```

Step 3: Configure and name VLANs on switches S2 and S3.

Create and name VLANs 10, 20, 30, and 99 on S2 and S3 using the commands from Step 1. Verify the correct configuration with the **show vlan brief** command.

Step 4: Assign switch ports to VLANs on S2 and S3.

Refer to the port assignment table on page 1. Ports are assigned to VLANs in interface configuration mode, using the **switchport access vlan** *vlan-id* command. You can assign each port individually or you can use the **interface range** command to simplify this task, as shown here. The commands are shown for S3 only, but you should configure both S2 and S3 similarly. Save your configuration when done.

S3(config)#interface range fa0/6-10
S3(config-if-range)#switchport access vlan 30
S3(config-if-range)#interface range fa0/11-17
S3(config-if-range)#switchport access vlan 10
S3(config-if-range)#interface range fa0/18-24
S3(config-if-range)#switchport access vlan 20
S3(config-if-range)#end
S3#copy running-config startup-config

Destination filename [startup-config]? [enter] Building configuration… [OK]

Step 5: Determine which ports have been added.

Use the **show vlan id** *vlan-number* command on S2 to see which ports are assigned to VLAN 10. Which ports are assigned to VLAN 10?_____

Note: The **show vlan id** *vlan-name* displays the same output. You can also view VLAN assignment information using the **show interfaces** *interface* **switchport** command.

Step 6: Assign the management VLAN.

A management VLAN is any VLAN that you configure to access the management capabilities of a switch. VLAN 1 serves as the management VLAN if you did not specifically define another VLAN. You assign the management VLAN an IP address and subnet mask.

A switch can be managed via HTTP, Telnet, SSH, or SNMP. Because the out-of-the-box configuration of **a Cisco switch has VLAN 1 as the default VLAN**, VLAN 1 is a bad choice as the management VLAN. You do not want an arbitrary user who is connecting to a switch to default to the management VLAN.

Recall that you configured the management VLAN as VLAN 99 earlier in this lab. From interface configuration mode, use the **ip address** command to assign the management IP address to the switches.

S1(config)#interface vlan 99
S1(config-if)#ip address 172.17.99.11 255.255.255.0
S1(config-if)#no shutdown
S2(config)#interface vlan 99
S2(config-if)#ip address 172.17.99.12 255.255.255.0
S2(config-if)#no shutdown
S3(config)#interface vlan 99
S3(config-if)#ip address 172.17.99.13 255.255.255.0
S3(config-if)#no shutdown

Assigning a management address allows IP communication between the switches, and also allows any host connected to a port assigned to VLAN 99 to connect to the switches. Because VLAN 99 is

configured as the management VLAN, any ports assigned to this VLAN are considered management ports and should be secured to control which devices can connect to these ports.

Step 7: **Configure trunking and the native VLAN for the trunking ports on all switches.**

Trunks are connections between the switches that allow the switches to exchange information for all VLANS. By default, a trunk port belongs to all VLANs, as opposed to an access port, which can only belong to a single VLAN.

If the switch supports both ISL and 802.1Q VLAN encapsulation, the trunks must specify which method is being used.

Because the 2960 switch only supports 802.1Q trunking, it is not specified in this lab. A native VLAN is assigned to an 802.1Q trunk port. In the topology, the native VLAN is VLAN 99. An 802.1Q trunk port supports traffic coming from many VLANs (tagged traffic) as well as traffic that does not come from a VLAN (untagged traffic). The 802.1Q trunk port places untagged traffic on the native VLAN.

Untagged traffic is generated by a computer attached to a switch port that is configured with the native VLAN. One of the IEEE 802.1Q specifications for Native VLANs is to maintain backward compatibility with untagged traffic common to legacy LAN scenarios.

For the purposes of this lab, a native VLAN serves as a common identifier on opposing ends of a trunk link.

It is a best practice to use a VLAN other than VLAN 1 as the native VLAN.

Use the **interface range** command in global configuration mode to simplify configuring trunking.

```
S1(config)#interface range fa0/1-5
S1(config-if-range)#switchport mode trunk
S1(config-if-range)#switchport trunk native vlan 99
S1(config-if-range)#no shutdown
S1(config-if-range)#end
S2(config)# interface range fa0/1-5
S2(config-if-range)#switchport mode trunk
S2(config-if-range)#switchport trunk native vlan 99
S2(config-if-range)#no shutdown
S2(config-if-range)#end
S3(config)# interface range fa0/1-5
S3(config-if-range)#switchport mode trunk
S3(config-if-range)#switchport trunk native vlan 99
S3(config-if-range)#no shutdown
S3(config-if-range)#end
```

Verify that the trunks have been configured with the **show interface trunk** command.

S1#show interface trunk

```
Port        Mode         Encapsulation  Status        Native vlan
Fa0/1       on           802.1q         trunking      99
Fa0/2       on           802.1q         trunking      99

Port      Vlans allowed on trunk
Fa0/1       1-4094
Fa0/2       1-4094

Port        Vlans allowed and active in management domain
Fa0/1       1,10,20,30,99
Fa0/2       1,10,20,30,99


Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       1,10,20,30,99
Fa0/2       1,10,20,30,99
```

Step 8: Verify that the switches can communicate.

From S1, ping the management address on both S2 and S3.

S1#ping 172.17.99.12

Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 172.17.99.12, timeout is 2 seconds:

!!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/9

ms

S1#ping 172.17.99.13

Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 172.17.99.13, timeout is 2 seconds:

.!!!! Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Step 9: Ping several hosts from PC2.

Ping from host PC2 to host PC1 (172.17.10.21).

Is the ping attempt successful?_____Ping from host PC2 to the switch VLAN 99 IP address 172.17.99.12.

Is the ping attempt successful? _____

Because these hosts are on different subnets and in different VLANs, they cannot communicate without a Layer 3 device to route between the separate subnetworks. Ping from host PC2 to host PC5.

Is the ping attempt successful?_____Because PC2 is in the same VLAN and the same subnet as PC5, the ping is successful.

Step 10: Move PC1 into the same VLAN as PC2.

The port connected to PC2 (S2 Fa0/18) is assigned to VLAN 20, and the port connected to PC1 (S2 Fa0/11) is assigned to VLAN 10. Reassign the S2 Fa0/11 port to VLAN 20. You do not need to first remove a port from a VLAN to change its VLAN membership. After you reassign a port to a new VLAN, that port is automatically removed from its previous VLAN.

S2#configure terminal

Enter configuration commands, one per line. End with
CNTL/Z.

S2(config)#interface fastethernet 0/11
S2(config-if)#switchport access vlan 20
S2(config-if)#end

Ping from host PC2 to host PC1.

Is the ping attempt successful?_____Even though the ports used by PC1 and PC2 are in the same VLAN, they are still in different subnetworks, so they cannot communicate directly.

Step 11: Change the IP address and network on PC1.

Change the IP address on PC1 to 172.17.20.22. The subnet mask and default gateway can remain the same. Once again, ping fromhost PC2 to host PC1, using the newly assigned IP address. Is the ping attempt successful?_____Why was this attempt successful?

Task 5: Document the Switch Configurations

On each switch, capture the running configuration to a text file and save it for future reference.

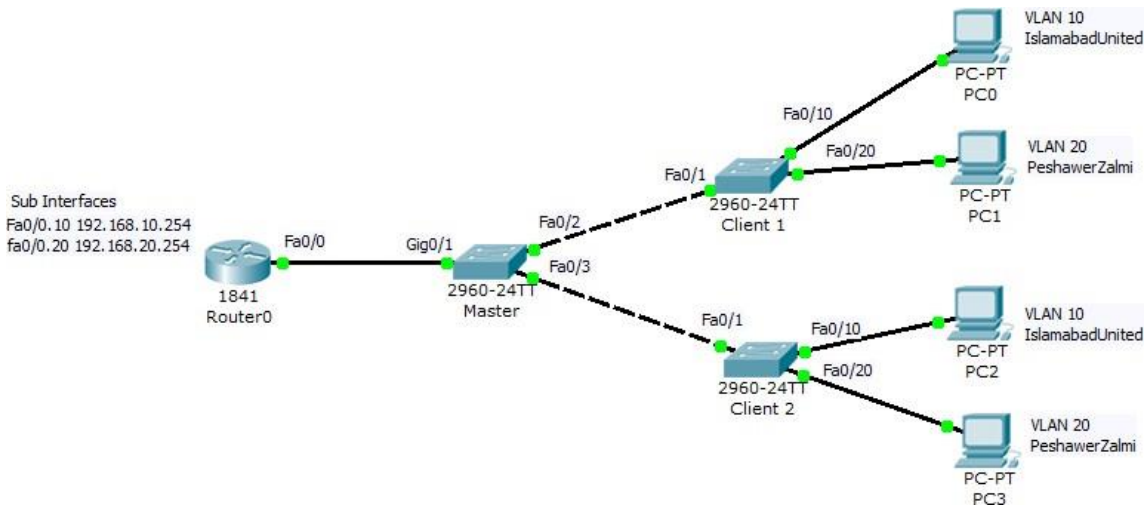
Task 6: Clean Up

Erase the configurations and reload the switches. Disconnect and store the cabling. For PC hosts that are normally connected to other networks (such as the school LAN or to the Internet), reconnect the appropriate cabling and restore the TCP/IP settings.

Inter VLAN Routing

As we've learned that each VLAN is a unique broadcast domain, so, computers on separate VLANs are, by default, not able to communicate. There is a way to permit these computers to communicate; it is called inter-VLAN routing.

First of all open the packet tracer and Make a scenario as shown in the above figure. Label Each PC, Router and Switch.



First, we need to create vlans in master and Client switches, just like we did in last lab. For that, go to switches and do the following:

Master:

```
Switch>en
Switch#config t
Switch(config)#hostname Master
Master(config)#vlan 10                                  //Creating VLANs
Master(config-vlan)#name IslamabadUnited
Master(config-vlan)#exit
Master(config)# vlan 20
Master(config-vlan)#name PeshawerZalmi
Master(config-vlan)#exit
Master(config)#vlan 99
Master(config-vlan)#name Management
Master(config-vlan)#exit
Master(config)#int vlan 99                  //Asssigning IP address to trunk port
Master(config-vlan)#ip address 192.168.99.1 255.255.255.0
Master(config-vlan)#exit
Master(config)#int range fa0/1-5            //Including fastethernet ports to VLAN 99
Master(config-if-range)#switchport trunk access vlan 99
Master(config-if-range)#end
Master#writing memory
```

Building configuration…
[OK]

Client 1:

```
Switch>en
Switch#config t
Switch(config)#hostname Client1
Client1(config)#vlan 10                     //Creating VLANs
Client1(config-vlan)#name IslamabadUnited
Client1(config-vlan)#exit
Client1(config)# vlan 20
Client1(config-vlan)#name PeshawerZalmi
Client1(config-vlan)#exit
Client1(config)#vlan 99
Client1(config-vlan)#name Management
Client1(config-vlan)#exit
Client1(config)#interface range fa0/10-14
                                  //Including fastethernet ports (10-14) to VLAN 10
Client1(config-if)#switchport access vlan 10
Client1(config-if)#exit
Client1(config)#interface range fa0/20-24
Client1(config-if-range)#switchport access vlan 20
                                  //Including fastethernet ports (20-24) to VLAN 20
Client1(config-if-range)#exit
Client1(config)#int vlan 99                 //Asssigning IP address to trunk port
Client1(config-vlan)#ip address 192.168.99.2 255.255.255.0
Client1(config-vlan)#exit
```

Client1(config)#int range fa0/1-5 //Including fastethernet ports (1-5) to VLAN 99
Client1(config-if-range)#switchport mode trunk
Client1(config-if-range)#switchport trunk access vlan 99
Client1(config-if-range)#end
Client1#writing memory
Building configuration…
[OK]

Client 2:

Switch>en
Switch#config t
Switch(config)#hostname Client2
Client2(config)#vlan 10
Client2(config-vlan)#name IslamabadUnited
Client2(config-vlan)#exit
Client2(config)# vlan 20
Client2(config-vlan)#name PeshawerZalmi
Client2(config-vlan)#exit
Client2(config)#vlan 99
Client2(config-vlan)#name Management
Client2(config-vlan)#exit
Client2(config)#interface range fa0/10-14
Client2(config-if)#switchport access vlan 10
Client2(config-if)#exit
Client2(config)#interface range fa0/20-24
Client2(config-if-range)#switchport access vlan 20
Client2(config-if-range)#exit
Client2(config)#int vlan 99
Client2(config-vlan)#ip address 192.168.99.3 255.255.255.0
Client2(config-vlan)#exit
Client2(config)#int range fa0/1-5
Client2(config-if-range)#switchport mode trunk
Client2(config-if-range)#switchport trunk access vlan 99
Client2(config-if-range)#end
Client2#writing memory
Building configuration…
[OK]

In any of the Client switches, show VLAN brief.

Another method for creating VLANs is through vtp Domain. In the master switch, you can create a vtp domain and share it with Client switches through a vtp password. Once you enter the password in Clients, VLANs created in Master switch are automatically loaded. But, you still need to add/include different fastethernet ports to those VLANs.

Master:

Master(config)#vtp domain cisco
Changing VTP domain name from NULL to cisco

Master(config)#vtp password network
Setting device VLAN database password to network

Clients:

Client1(config)#vtp password network
Setting device VLAN database password to network
Client1#show vlan brief

Now you can skip creating VLANs step in client switches.

Configuring inter-VLAN routing using router-on-a-stick:

Router-on-a-stick is a type of router configuration in which a single physical interface manages traffic between multiple VLANs on a network. The router interface has to be configured to operate as a trunk link and is connected to a switch port (SW1) which will have to be configured in trunk mode.

Now, for Inter VLAN Routing, we need to configure our router. Router has sub-interfaces that can be assigned with ip addresses and used as gateway for VLAN communication. Each subinterface is created using the interface *interface_id.Subinterface_id* in the global configuration mode.

Router:

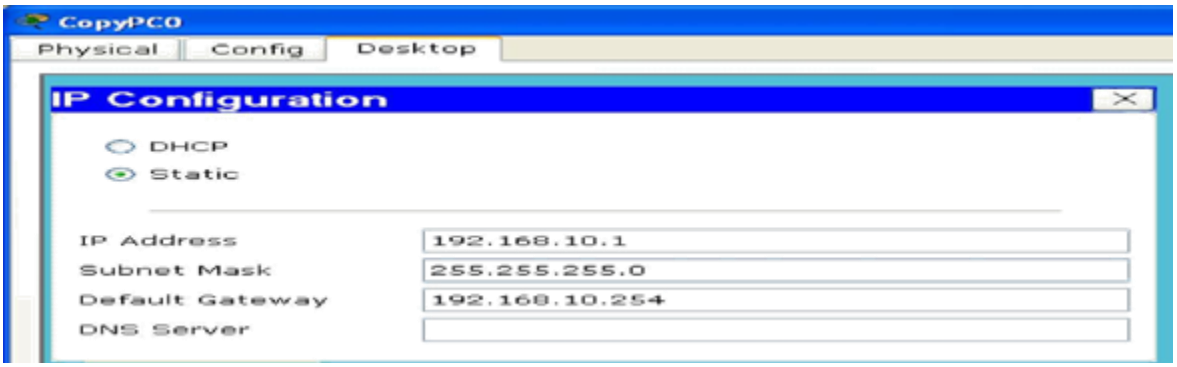
Router<en
Router#config t
Router(config)#int fa0/0
Router(config-if)#no shutdown
Router(config-if)#int fa0/0.10
Router(config-subif)#encapsulationdot1Q 10
Router(config-subif)#ip address 192.168.10.254 255.255.255.0
Router(config-subif)#exit
Router(config)# int fa0/0.20
Router(config-subif)#encapsulationdot1Q 20
Router(config-subif)#ip address 192.168.20.254 255.255.255.0
Router(config-subif)#exit
Router(config)#exit
Router#write memory
Building configuration…
[OK]

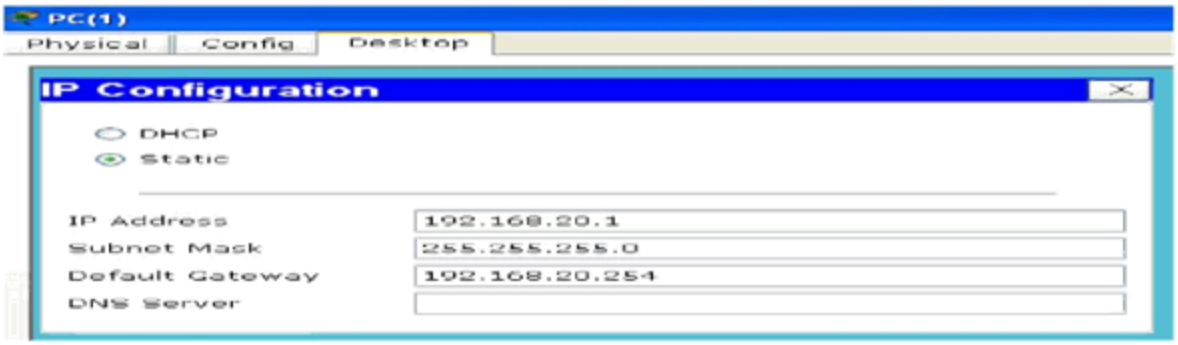
Now, subinterface fa0/0.10 is the gateway for vlan 10 and fa0/0.20 is gateway for vlan 20. For inter communication of these vlans, router takes packets from one vlan and forwards to vlan 20, sends back replies through the same connection and so on.

One last thing we have to do is, make port of Master switch which is connected to router as a trunk port.

Master(config)#int range gig0/1
Master(config-if-range)#switchport mode trunk
Master(config-if-range)#exit

Configure Each PC.


CopyPC0
Physical
Config
Desktop
IP Configuration
DHCP
Static
IP Address
192.168.10.1
Subnet Mask
255.255.255.0
Default Gateway
192.168.10.254
DNS Server


PC(1)
Physical
Config
Desktop
IP Configuration
DHCP
Static
IP Address
192.168.20.1
Subnet Mask
255.255.255.0
Default Gateway
192.168.20.254
DNS Server


PC 2
Physical
Config
Desktop
IP Configuration
DHCP
Static
IP Address
192.168.10.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.10.254
DNS Server


PC 3
Physical
Config
Desktop
IP Configuration
DHCP
Static
IP Address
192.168.20.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.20.254
DNS Server

For Checking Connectivity between Vlan 10 and Vlan 20, Open the command prompt of a PC on one Vlan and Ping the System on the other Vlan.

Note: For Failed pings, use capture/forward.

For Checking Connectivity between Vlan 10 and Vlan 20, Open the command prompt of a PC on one Vlan and Ping the System on the other Vlan.

TASKS:

Tasks related to the lab will be provided by the lab instructor.

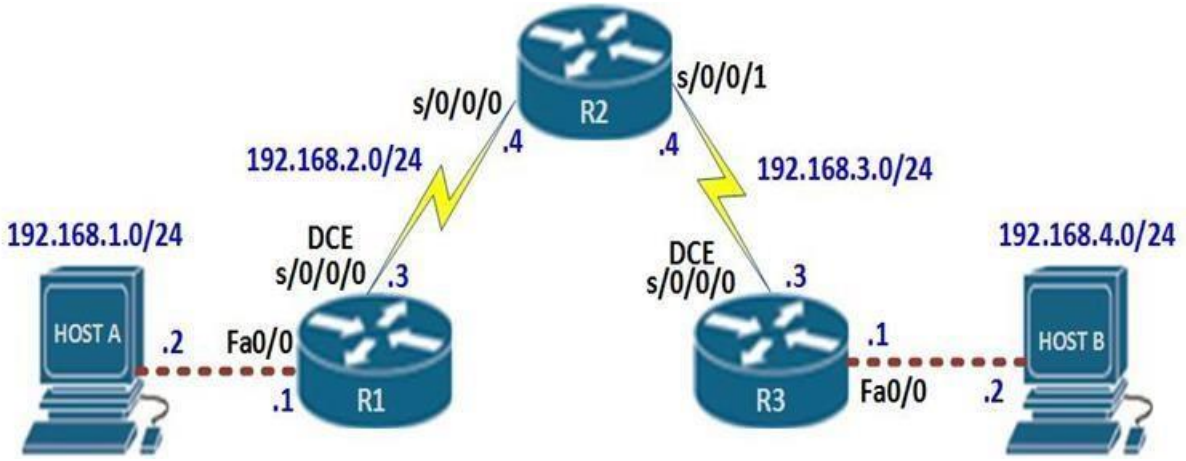
Lab 06

Static Routing

Upon Completion of this Lab, you will be able to learn these Objectives:

- Static routing and its uses
- How to configure indirectly connected routes using static routing
- Updating the routing table

As you may already know, the work of the router is to forward packets from the source device to the destination device. In between there may be several routers. The router uses a database known as the routing table to forward these packets.



The network above shows a small network consisting of 3 routers and 2 hosts. In this scenario, R1 can ping HOST A, R1 can ping R2 s0/0/0 interface but not interface s0/0/1.

R3 can ping HOST B, R3 can ping R2's s0/0/1 interface ONLY. HOST A and HOST B cannot communicate.

```
HOST B>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.4.2:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
HOST A >ping 192.168.1.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.4.1: Destination host unreachable.
Reply from 192.168.4.1: Destination host unreachable.
Reply from 192.168.4.1: Destination host unreachable.
Reply from 192.168.4.1: Destination host unreachable.

Ping statistics for 192.168.4.2:
   Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The routing table is the database that contains information about various networks, we have said that these remote networks may either be learnt through routing protocols or manually configured routes.

The output of the "show ip route" command on a router, shows the routes that a particular router can reach. By default, a router will only know of directly connected routes.

Directly connected routes in our scenario, from R1's perspective are the network connected to HOST A and the network between R1 and R2.

Since no other configuration has been made on these routers, R2 and R3, should only have directly connected routes.

The directly connected networks are the only networks that can be reached by a particular router. In our scenario, this means that;

- Host A can ping R1
- R1 can ping R2's s0/0/0 interface but not interface s0/0/1
- R2 can ping R1's s0/0/0 interface but not interface fa0/0 or HOST A
- R2 can ping R3's s0/0/0 interface but not interface fa0/0 or HOST B
- R3 can ping R2's s0/0/1 interface but not interface s0/0/0
- HOST B can ping R3.
- Neither hosts can ping each other
- R1 and R3 cannot ping each other.

Configuration of Static Routing

Static routes are one way we can communicate to remote networks. In production networks, static routes are mainly configured when routing from a particular network to a stub network.

stub networks are networks that can only be accessed through one point or one interface.
In the above scenario, the 192.168.1.0/24 and 192.168.4.0/24 networks are stub networks. This means that for hosts in these network segments only have one way to communicate with other hosts, which is R1 and R3 for the 192.168.1.0/24 and 192.168.4.0/24 networks respectively.

The command needed to configure a static route is shown below.

Router(config)# ip route (network-address) (subnet-mask) (next-hop ip address/ exit interface)

The table below explains the meaning of each of the parameters in the ip route command as well as an example of the command which would be used on R1 to configure a static route to R3's LAN network (192.168.4.0/24).

IP route	State that the route being configured is a static route	IP route
Network-address	The network address of the destination network. This is the network I am trying to reach.	192.168.4.0
Subnet-mask	The network address of the destination network that I am trying to reach	255.255.255.0
Nexthop IP address	This is the IP address of the router that is connecting me to the desired network	192.168.2.4
Exit interface	This is the exit point interface on my router that connects to the router that will take me to the desired network	s0/0/0

Therefore, to configure a static route on R1 for network 192.168.3.0/24 and 192.168.4.0/24, the commands to be issued on R1 is:

R1(config)# ip route 192.168.3.0 255.255.255.0 192.168.2.4
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.2.4
OR
R1(config)# ip route 192.168.3.0 255.255.255.0 s0/0/0
R1(config)# ip route 192.168.4.0 255.255.255.0 s0/0/0

Similarly, for R2:
R2(config)# ip route 192.168.1.0 255.255.255.0 s0/0/0
R2(config)# ip route 192.168.4.0 255.255.255.0 s0/0/1

And for R3:
R3(config)# ip route 192.168.2.0 255.255.255.0 s0/0/0
R3(config)# ip route 192.168.1.0 255.255.255.0 s0/0/0

Run the show ip route command on each of the routers to see the list of all routes the routers is aware of. When all the configurations have been made on all the three routers, communication between HOST A and HOST B should be possible.

Setting up a Default Route

Static routing solves one more network problem. It can redirect all unmatched packets to a certain port. This feature is extremely helpful in several situations. We can set a default route for internet connection or we can implement a security measurement to deal with all matched packet.

By default, Routers are configured to drop the packet if destination address is not found in routing table. Default route will override this behavior. If no match for destination network is found in routing table then it would be forwarded to the default route. Thus default route isa way to deal withall unmatchedpackets.

The syntax for configuring a static default route is:

Router(config)# ip route 0.0.0.0 0.0.0.0 [next-hop ip address/ exit interface]

In this scenario, to configure a default static route, the command sequence on R1 would be.

R1(config)# ip route 0.0.0.0 0.0.0.0 s0/0/0

Deletion of a Static Route

In static routing we have to manage all routes manually. If any route goes down, we have to remove that manually. Removing a route in static routing is easier than you think. All you need to do is just add a keyword no before the same command that we have used to configure the static route.

no ip route command is used to remove the route from routing table. Following commands will remove the route from their respective routes. For example,

R1(config)# ip route 192.168.3.0 255.255.255.0 s0/0/0

Advantage of static routing

- It is easy to implement.
- It is most secure way of routing, since no information is shared with other routers.
- It puts no overhead on resources such as CPU or memory.

Disadvantage of static routing

- It is suitable only for small network.
- If a link fails it cannot reroute the traffic.

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 07

Routing Information Protocol (RIP)

This lab teaches the basic configuration for connecting two routers. Moreover, this lab contains RIP (Routing Information Protocol) and its configuration.

Learning Objectives:

- Understanding RIP (Routing Information Protocol)
- Revision of configuring Routes Statically
- Configuring Routes via RIP
- Configuring RIP on a Cisco Router
- Configuring RIP version 2 on a VLSM network

Overview:

In this lab we will see how to connect two routers together and how to configure router so that they can communicate with each other plus networks connected to each other. We will also see what is RIP, how RIP work and how to configure RIP.

Routing Information Protocol (RIP) is a dynamic routing protocol which uses hop count as a routing metric to find the best path between the source and the destination network. It is a distance vector routing protocol which has AD value 120 and works on the application layer of OSI model. RIP uses port number 520.

Advertising:

The Cisco IOS software in every router sends routing information updates every 30 seconds to its neighbor routers; this process is termed advertising.

Hop Count:

Hop count is the number of routers occurring in between the source and destination network. The path with the lowest hop count is considered as the best route to reach a network and therefore placed in the routing table. RIP prevents routing loops by limiting the number of hopes allowed in a path from source and destination. The maximum hop count allowed for RIP is 15 and hop count of 16 is considered as network unreachable.

Features of RIP:

1. Updates of the network are exchanged periodically.
2. Updates (routing information) are always broadcast.
3. Full routing tables are sent in updates.
4. Routers always trust on routing information received from neighbor routers. This is also known as *Routing on rumours*.

RIP versions:

There are three vesions of routing information protocol – **RIP Version1**, **RIP Version2**.
RIP v1 is known as *Classful* Routing Protocol because it doesn't send information of subnet mask in its routing update.
RIP v2 is known as *Classless* Routing Protocol because it sends information of subnet mask in its routing update.

>> Use debug command to get the details :

***#** debug ip rip*

>> Use this command to show all routes configured in router, say for router R1 :

***R1#** show ip route*

>> Use this command to show all protocols configured in router, say for router R1 :

***R1#** show ip protocols*

RIP timers:

- **Update timer:** The default timing for routing information being exchanged by the routers operating RIP is 30 seconds. Using Update timer, the routers exchange their routing table periodically.
- **Invalid timer:** If no update comes until 180 seconds, then the destination router considers it as invalid. In this scenario, the destination router mark hop count as 16 for that router.
- **Hold down timer:** This is the time for which the router waits for neighbor router to respond. If the router isn't able to respond within a given time, then it is declared dead. It is 180 seconds by default.
- **Flush time:** It is the time after which the entry of the route will be flushed if it doesn't respond within the flush time. It is 60 seconds by default. This timer starts after the route has been declared invalid and after 60 seconds i.e time will be 180 + 60 = 240 seconds.
  - Note that all these times are adjustable. Use this command to change the timers:

- **R1(config-router)#** timers basic
- **R1(config-router)#** timers basic 20 80 80 90

Connecting Two Routers:

To connect two Routers, place two routers in your working space.

1. Click on one of the Routers.
2. Switch Off the Router.
3. Select the additional WIC card (NM-4A/S) containing serial ports and place it to the black space on the router.
4. Switch On the Router. (Repeat Step 1-4 for the other router)
5. Now select the DCE cable from Wires tab and connect routers using their serial ports.

Configurations:

First of all go under serial interface of the router and assign it an ip.

R1(Config)#interface serial 1/0
R1(Config-if)#ip address 192.168.2. 255.255.255.0
R1(Config-if)#noshutdown

Now set clock rate of the router(You'll do it for one of the two routers).

R1(Config-if)#clock rate 56000

Assign IP addresses to the serial interface of the second router, you are done.

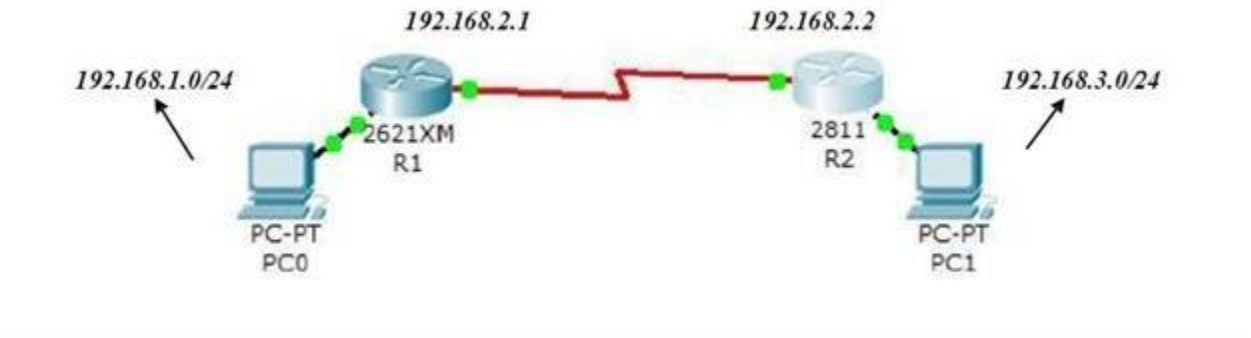
Configuring Static Route:

Go to R1 and in global configuration mode enter following command.

R1(Config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2

Now go to R2 and in global configuration mode enter following command.

```
R2(Config)#ip route 192.168.1.0 255.255.255.0  192.168.2.1
```



- **Configuring RIP:**

There are two main steps to configure RIP.

1. Enable RIP
2. Advertise its networks

- **Enable RIP:**

On router R1

```
R1#Conf t
R1(config)#router rip
```

Advertising Networks

```
R1(config-router)#network    192.168.1.0
R1 (config-router)#network 192.168.2.0
```

On router R2

```
R2#Conf t
R2(config)#router rip
```

Advertising Networks

```
R2(config-router)#network 192.168.3.0
R2 (config-router)#network 192.168.2.0
```

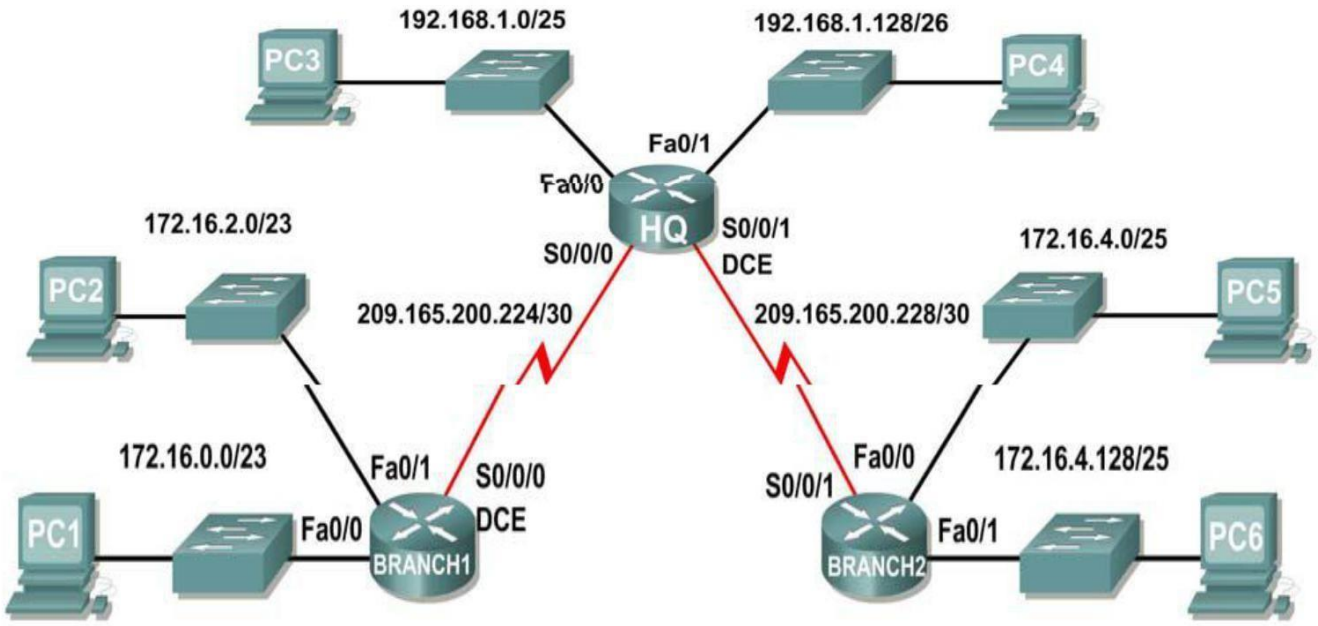
- **To Check Routing Table:**

```
R1#show ip route
```

RIP Version 2 Configuration:

RIPv2 is a classless protocol. RIPv2 multicasts the entire routing table to all adjacent routers at the address 224.0.0.9, as opposed to RIPv1 which uses broadcast (255.255.255.255). Unicast addressing is still allowed for special applications.

Topology Diagram



Addressing Table	Interface	IP Address	Subnet Mask	Default Gateway
Device	**Interface**	IP Address	Subnet Mask	Default gateway
HQ	**Fa0/0**	192.168.1.1	255.255.255.128	N/A
	Fa0/1	192.168.1.129	255.255.255.192	N/A
	S0/0/0	209.165.200.225	255.255.255.252	N/A
	S0/0/1	209.165.200.229	255.255.255.252	N/A
BRANCH1	**Fa0/0**	172.16.0.1	255.255.254.0	N/A
	Fa0/1	172.16.2.1	255.255.254.0	N/A
	S0/0/0	209.165.200.226	255.255.255.252	N/A
BRANCH2	**Fa0/0**	172.16.4.1	255.255.255.128	N/A
	Fa0/1	172.16.4.129	255.255.255.128	N/A
	S0/0/1	209.165.200.230	255.255.255.252	N/A
PC1	**NIC**	172.16.0.10	255.255.254.0	172.16.0.1
PC2	**NIC**	172.16.2.10	255.255.254.0	172.176.2.1
PC3	**NIC**	192.168.1.10	255.255.255.128	192.168.1.1
PC4	**NIC**	192.168.1.138	255.255.255.192	192.168.1.129
PC5	**NIC**	172.16.4.10	255.255.255.128	172.16.4.1
PC6	**NIC**	172.16.4.138	255.255.255.128	172.16.4.138

In this lab, you will begin by loading configuration scripts on each of the routers. These scripts contain errors that will prevent end-to-end communication across the network. You will need to troubleshoot each

router to determine the configuration errors, and then use the appropriate commands to correct the configurations. Whenyou have corrected all of the configuration errors, all of the hosts on the network should be able to communicate with each other. The network should also have the following requirements met:

- RIPv2 routing is configured on the BRANCH1 router.
- RIPv2 routing is configured on the BRANCH2 router.
- RIPv2 routing is configured on the HQ router.
- RIP updates must be disabled on the BRANCH1, BRANCH2, and HQ LAN interfaces

Task 1: Cable, Erase, and Reload the Routers.

Step 1: Cable a network.

Cable a network that is similar to the one in the Topology Diagram.

Step 2: Clear the configuration on each router.

Clear the configuration on each of the routers using the e**rase startup-config** command and then **reload** the routers. Answer **no** if asked to save changes.

Task 2: Load Routers with the Supplied Scripts

Step 1: Load the following script onto the BRANCH1 router:

```
hostname BRANCH1
!
interface FastEthernet0/0
ip address 172.16.0.1  255.255.254.0
duplex auto
speed auto
no shutdown
!
interface FastEthernet0/1
ip address 172.16.2.1  255.255.254.0
duplex auto
speed auto
no shutdown
!
interface Serial0/0/0
ip address 209.165.200.226  255.255.255.252
clock rate 64000
no shutdown
!
router rip
passive-interface FastEthernet0/0
```

```
passive-interface FastEthernet0/1
network 172.16.0.0
network 209.165.200.0
!
ip classless
line con 0
line vty 0 4
login
!
!
end
```

Step 2: Load the following script onto the BRANCH2 router.

```
hostname BRANCH2
!
!
!
interface FastEthernet0/0
ip address 172.16.4.129  255.255.255.128
duplex auto
speed auto
no shutdown
!
interface FastEthernet0/1
ip address 172.16.4.1  255.255.255.128
duplex auto
speed auto
no shutdown
!
interface Serial0/0/1
ip address 209.165.200.230  255.255.255.252
no shutdown
!
router rip
version 2
passive-interface  FastEthernet0/0
passive-interface  FastEthernet0/1
network 209.165.200.0
!
ip classless
line con 0
line vty 0 4
login
!
!
end
```

Step 3: Load the following script onto the HQ router.

```
hostname HQ
!
!
!
interface FastEthernet0/0
ip address 192.168.1.1  255.255.255.128
duplex auto
speed auto
no shutdown
!
interface FastEthernet0/1
ip address 192.168.1.129 255.255.255.192
duplex auto
speed auto
no shutdown
!
interface Serial0/0/0
ip address 209.165.200.225  255.255.255.252

no shutdown
!
interface Serial0/0/1
ip address 209.165.200.229  255.255.255.252
no shutdown
!
router rip
version 2
passive-interface  FastEthernet0/0
passive-interface  FastEthernet0/1
network 192.168.1.0
network 209.165.200.0
!
ip classless
line con 0
line vty 0 4
login

!
!
end
```

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 08

Open Shortest Path First Protocol (OSPF)

Learning Objectives:
Upon completion of this lab, you will be able to:
• Cable a network according to the Topology Diagram
• Erase the startup configuration and reload a router to the default state
• Perform basic configuration tasks on a router
• Configure and activate interfaces
• Configure OSPF routing on all routers
• Configure OSPF router IDs
• Verify OSPF routing using show commands
• Configure a static default route
• Propagate default route to OSPF neighbors
• Configure OSPF Hello and Dead Timers
• Configure OSPF on a Multi-access network
• Configure OSPF priority
• Understand the OSPF election process

OSPF is a Link State protocol that's considered may be the most famous protocol among the Interior Gateway Protocol (IGP) family, developed in the mid 1980's by the OSPF working group of the IETF.

When configured, OSPF will listen to neighbors and gather all link state data available to build a topology map of all available paths in its network and then save the information in its topology database, also known as its **Link-State Database** (**LSDB**). Using the information from its topology database. From the information gathered, it will calculate the best shortest path to each reachable subnet/network using an algorithm called **Shortest Path First** (**SFP**). OSPF will then construct **three tables** to store the following information:

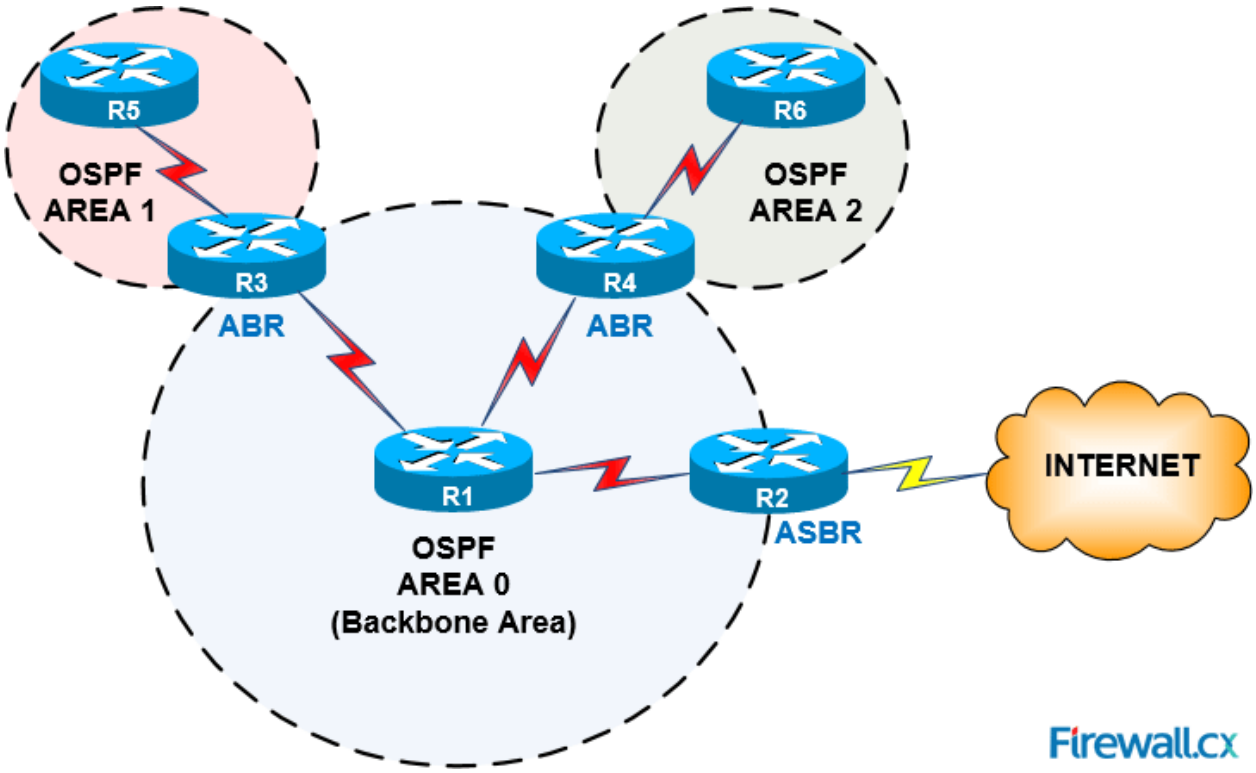
- **Neighbor Table:** Contains all discovered OSPF neighbors with whom routing information will be interchanged
- **Topology Table:** Contains the entire road map of the network with all available OSPF routers and calculated best and alternative paths.
- **Routing Table:** Contain the current working best paths that will be used to forward data traffic between neighbors.

OSPF AREAS

OSPF offers a very distinguishable feature named: **Routing Areas**. It means dividing routers inside a single autonomous system running OSPF, into areas where each area consists of a group of connected routers.

The idea of dividing the OSPF network into areas is to simplify administration and optimize available resources. Resource optimization is especially important for large enterprise networks with a plethora of network and links. Having many routers exchange the link state database could flood the network and reduce its efficiency – this was the need that led to the creation of concept Areas.

Areas are a logical collection of routers that carry the same **Area ID** or number inside of an OSPF network, the OSPF network itself can contain multiple areas, the first and main Area is called the backbone area **"Area 0"**, all other areas must connect to **Area 0** as shown in the diagram:



All routers within the same **Area** have the **same topology table** -**Link State Database**- but different **routing table** as OSPF calculates different best paths for each router depending on its location within the network topology while they will all share the same **Link State topology**.

The goal of having an **Area** is to localize the network as follow:

- The **Area boundaries** will give the opportunity of using **summarization**, as it's not possible to summarize network prefixes in normal link state protocols because routers are supposed to have the same map topology of the entire network coincide in all neighbors.

- **Area boundaries** will also help preventing fault containment by suppressing updates that take place when a change occurs in the network causing a flood of updates between routers. This also happens to be a weakness of link state protocols: When connecting large sized networks, it is very difficult to avoid link state database floods.

OSPF Router ID

Each OSPF router selects a router ID (RID) that has to be unique on your network. OSPF stores the topology of the network in its LSDB (Link State Database) and each router is identified with its unique router ID, if you have duplicate router IDs then you will run into reachability issues.

Because of this, two OSPF routers with the same router ID will not become neighbors but you could still have duplicated router IDs in the network with routers that are not directly connected to each other.

OSPF uses the following criteria to select the router ID:

1. Manual configuration of the router ID.
2. Highest IP address on a loopback interface.
3. Highest IP address on a non-loopback interface.

Manual setting ofrouter ID:

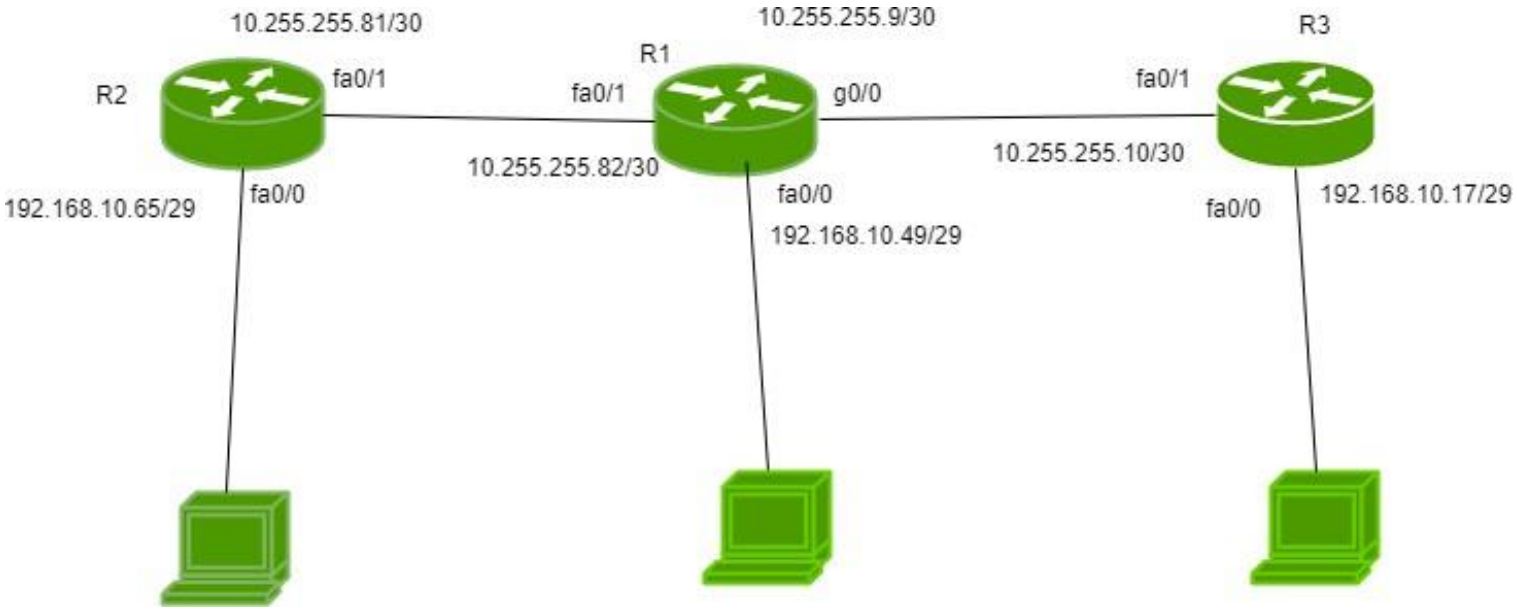
```
R1(config)#routerospf 1
R1(config-router)#router-id 1.1.1.1
```

Use the **router-id** command for this. Let's verify our work:

```
R1#show ip protocols | include Router ID
  Router ID 1.1.1.1
```

If you don't set a manual router ID, router learns the highest IP address itself, as soon as OSPF process is started.

OSPF Configuration:



There is a small topology in which there are 3 routers namely R1, R2, R3 are connected. R1 is connected to networks 10.255.255.80/30 (interface fa0/1), 192.168.10.48/29 (interface fa0/0) and 10.255.255.8/30 (interface gi0/0). R2 is connected to networks 192.168.10.64/29 (interface fa0/0), 10.255.255.80/30 (interface fa0/1). R3 is connected to networks 10.255.255.8/30 (int fa0/1), 192.168.10.16/29 (int fa0/0).

Now, configuring OSPF for R1.

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.10.48 0.0.0.7 area 1
R1(config-router)#network 10.255.255.80 0.0.0.3 area 1
R1(config-router)#network 10.255.255.8 0.0.0.3 area 1
```

Here, 1 is the OSPF instance or process I'd. It can be same or different (doesn't matter). You have to use wildcard mask here and area used is 1.

Now, configuring R2

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.10.64 0.0.0.7 area 1
R2(config-router)#network 10.255.255.80 0.0.0.3 area 1
```

Similarly, for R3

```
R3(config)#router ospf 1
R3(config-router)#network 192.168.10.16 0.0.0.7 area 1
R3(config-router)#network 10.255.255.8 0.0.0.3 area 1
```

You can check the configuration by typing command

```
R3#show ip protocols
```

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 09

Access control List (ACL)

Learning Objectives

.• Configure and verify ACLs to control traffic.
• Verify ACLs using the logging capabilities of the router

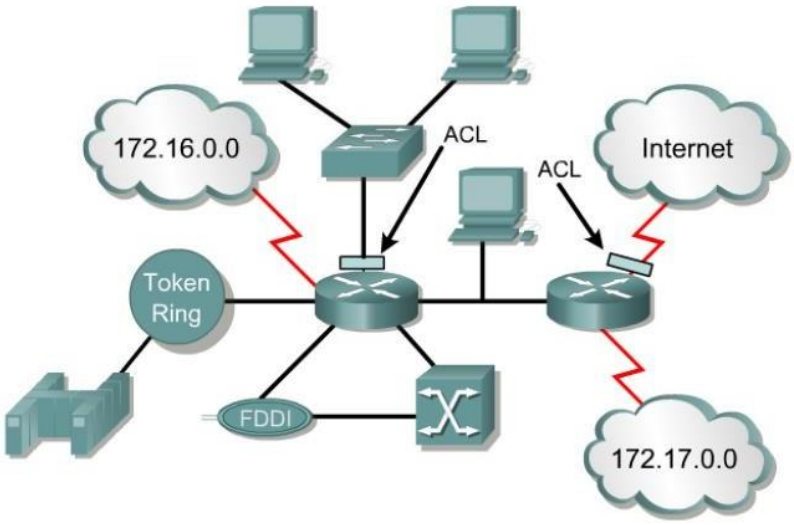
Overview

- Network administrators must figure out how to deny unwanted access to the network while allowing internal users appropriate access to necessary services
- Although security tools, such as passwords, callback equipment, and physical security devices are helpful, they often lack the flexibility of basic traffic filtering and the specific controls most administrators prefer.
- For example, a network administrator may want to allow user's access to the LAN the Internet, but not permit external users telnet access into the LAN.
- Routers provide basic traffic filtering capabilities, such as blocking Internet traffic, with access control lists (ACLs).
- An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols.
- This module will introduce standard and extended ACLs as a means to control network traffic, and how ACLs are used as part of a security solution.

What are ACLs?

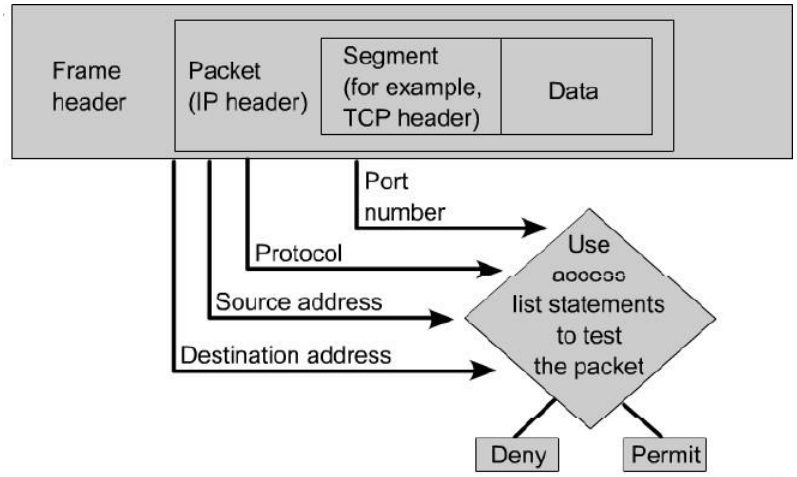
An access list is a sequential series of commands or filters.
These lists tell the router what types of packets to:

- accept or
- deny
- Acceptance and denial can be based on specified conditions.
- ACLs applied on the router's interfaces


172.16.0.0
ACL
ACL
Internet
Token
Ring
FDDI
172.17.0.0

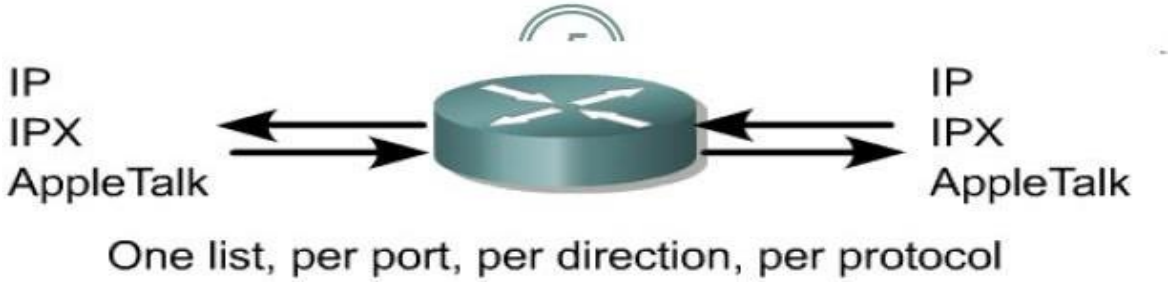
The router examines each packet to determine whether to forward or drop it, based on the conditions specified in the ACL.

- *Some* ACL decision points are:
- IP source address
- IP destination addresses
- UDP or TCP protocols
- upper-layer (TCP/UDP) port numbers



ACLs must be defined on a:

- per-protocol (IP, IPX, AppleTalk)
- per direction (in or out)
- per port (interface) basis.
- ACLs control traffic in one direction at a time on an interface.
- A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
- Finally every interface can have multiple protocols and directions defined.



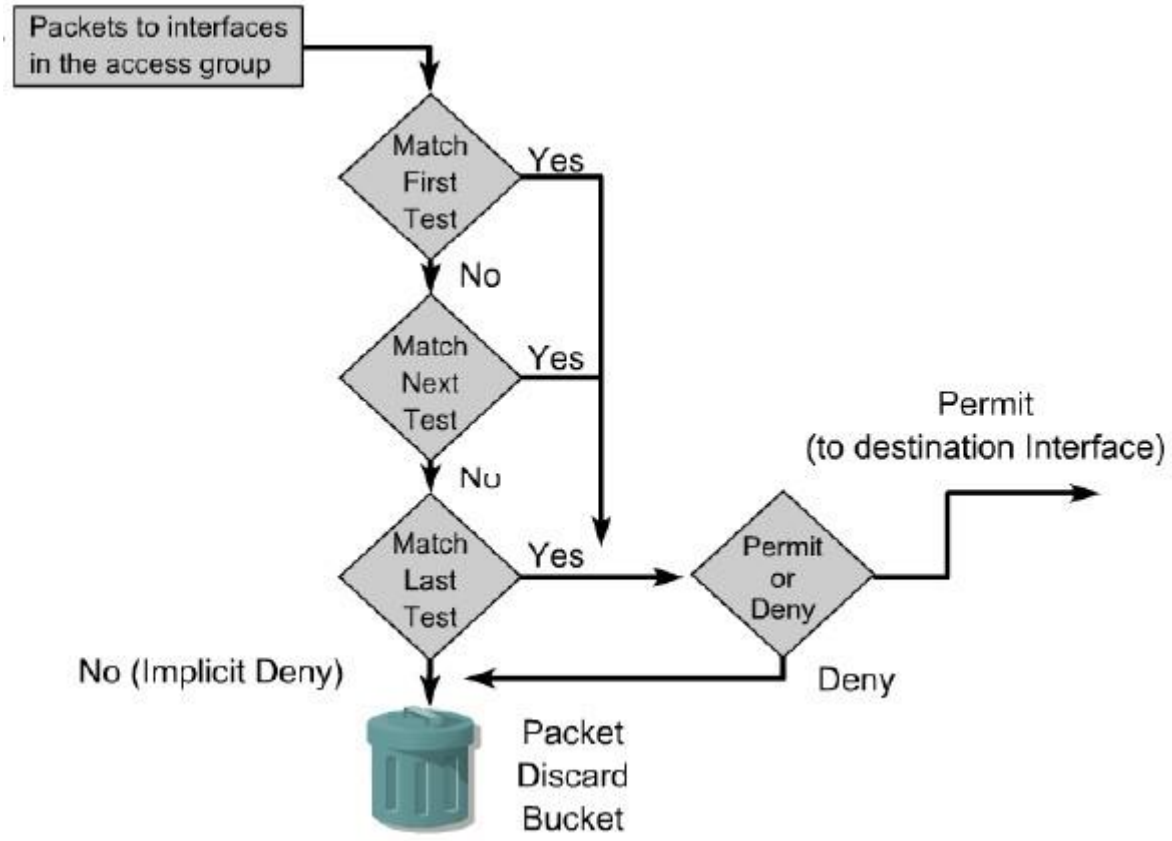
One list, per port, per direction, per protocol

With two interfaces and three protocols running, this router could have a total of 12 separate ACLs applied.

Working Of ACLs

An ACL is a group of statements that define whether packets are accepted or rejected coming into an interface or leaving an interface.

- ACL statements operate in sequential, logical order (top down).
- If a condition match is true, the packet is permitted or denied and the rest of the ACL statements are not checked.
- If all the ACL statements are unmatched, an implicit ***"deny any"*** statement is placed at the end of the list by ***default***. (not visible)
- When first learning how to create ACLs, it is a good idea to add the **implicit deny** at the end of ACLs to reinforce the dynamic presence of the command line.



Two types of ACLs

Standard IP ACLs

- Can only filter on source IP addresses
- Extended IP ACLs
- Can filter on:
- Source IP address
- Destination IP address
- Protocol (TCP, UDP)

- Port Numbers (Telnet – 23, http – 80, etc.)
- *and other parameters*

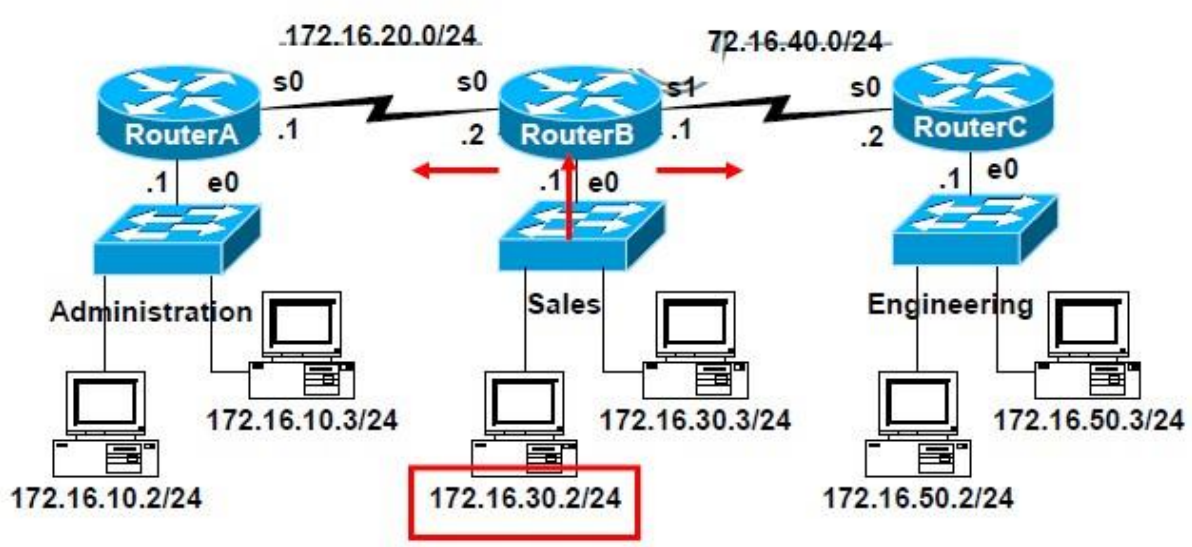
Creating Standard ACLs – 2 Steps

Step 1	Define the ACL by using the following command: `Router(config)#access-list access-list-number {permit | deny} {test-conditions}` A global statement identifies the ACL. Specifically, the 1-99 range is reserved for standard IP. This number refers to the type of ACL. In Cisco IOS Release 11.2 or newer, ACLs can also use an ACL name, such as education_group, rather than a number. The **permit** or **deny** term in the global ACL statement indicates how packets that meet the test conditions are handled by Cisco IOS software. **permit** usually means the packet will be allowed to use one or more interfaces that you will specify later.The final term or terms specifies the test conditions used by the ACL statement.
Step 2	Next, you need to apply ACLs to an interface by using the **access-group** command, as in this example: `Router(config-if)#{protocol} access-group access-list-number` All the ACL statements identified by `access-list-number` are associated with one or more interfaces. Any packets that pass the ACL test conditions can be permitted to use any interface in the access group of interfaces.

Step 1	Define the ACL by using the following command: `Router(config)#access-list access-list-number {permit | deny} {test-conditions}` A global statement identifies the ACL. Specifically, the 1-99 range is reserved for standard IP. This number refers to the type of ACL. In Cisco IOS Release 11.2 or newer, ACLs can also use an ACL name, such as education_group, rather than a number. The **permit** or **deny** term in the global ACL statement indicates how packets that meet the test conditions are handled by Cisco IOS software. **permit** usually means the packet will be allowed to use one or more interfaces that you will specify later.The final term or terms specifies the test conditions used by the ACL statement.

Protocol	Range
IP (Standard IP)	1-99
Extended IP	100-199
AppleTalk	600-699
IPX	800-899
Extended IPX	900-999
IPX Service Advertising Protocol	1000-1099

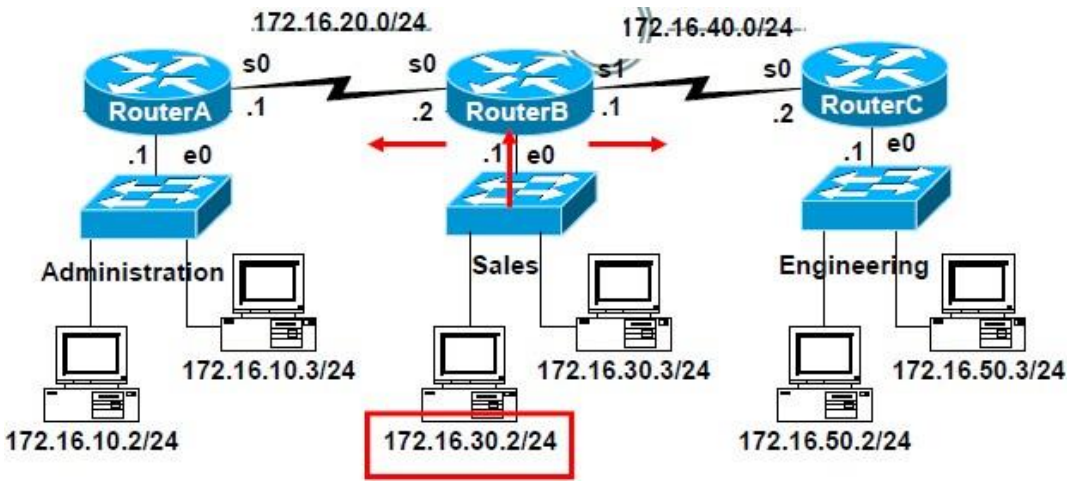
EXAMPLE



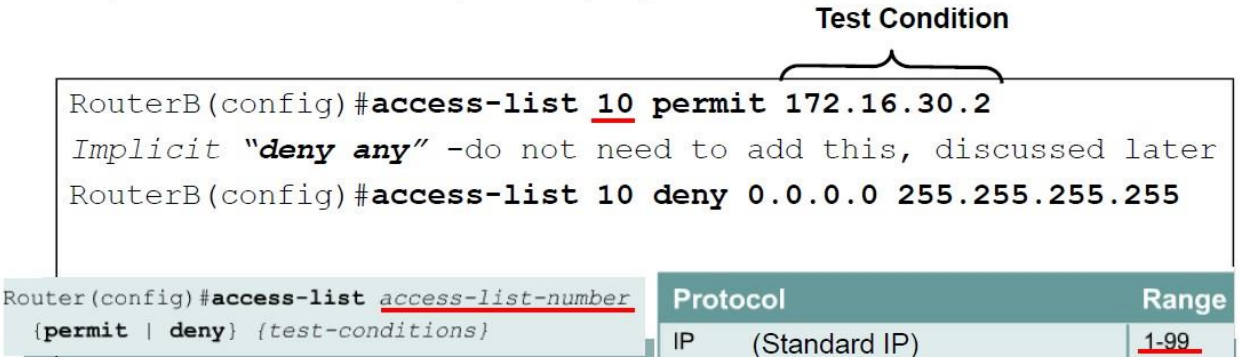
Task:

- Permit only the host 172.16.30.2 from exiting the Sales network.

- Deny all other hosts on the Sales network from leaving the 172.16.30.0/24 network.



Step 1 – ACL statements Implicit deny any, which is automatically added.



Applying ACLs

- You can define ACLs without applying them
- However, the ACLs will have no effect until they are applied to the router's interface.
- It is a good practice to apply the Standard ACLs on the interface closest to the destination of the traffic and Extended ACLs on the interface closest to the source.

Step 2 – Apply to an interface(s)

RouterB(config)#**access-list 10 permit 172.16.30.2**
*Implicit "**deny any**"* -do not need to add this, discussed later
RouterB(config)#**access-list 10 deny 0.0.0.0 255.255.255.255**
RouterB(config)# **interface e 0**
RouterB(config-if)# **ip access-group 10 in**

```
Router(config-if)#{protocol} access-group access-list-number
```

Step 2 – Or the outgoing interfaces

RouterB(config)#**access-list 10 permit 172.16.30.2**
*Implicit "**deny any**"* -do not need to add this, discussed later
RouterB(config)#**access-list 10 deny 0.0.0.0 255.255.255.255**
RouterB(config)# **interface s 0**
RouterB(config-if)# **ip access-group 10 out**
RouterB(config)# **interface s 1**
RouterB(config-if)# **ip access-group 10 out**
Because of the implicit deny any, this has an adverse effect of also denying packets from Administration from reaching Engineering, and denying packets from engineering to reach administration

```
RouterB(config)#access-list 10 permit 172.16.30.2
Implicit "deny any" -do not need to add this, discussed later
RouterB(config)#access-list 10 deny 0.0.0.0 255.255.255.255

RouterB(config)# interface s 0
RouterB(config-if)# ip access-group 10 out
RouterB(config)# interface s 1
RouterB(config-if)# ip access-group 10 out
```

TASKS:

Tasks related to the lab will be provided by the lab instructor.

Lab 10

Configuring NAT (Network Address Translation)

Lab exercise will result in the following outcomes. Please ensure you complete all the steps yourself and don't indulge in any kind of cross talk or interference in thework of other students.

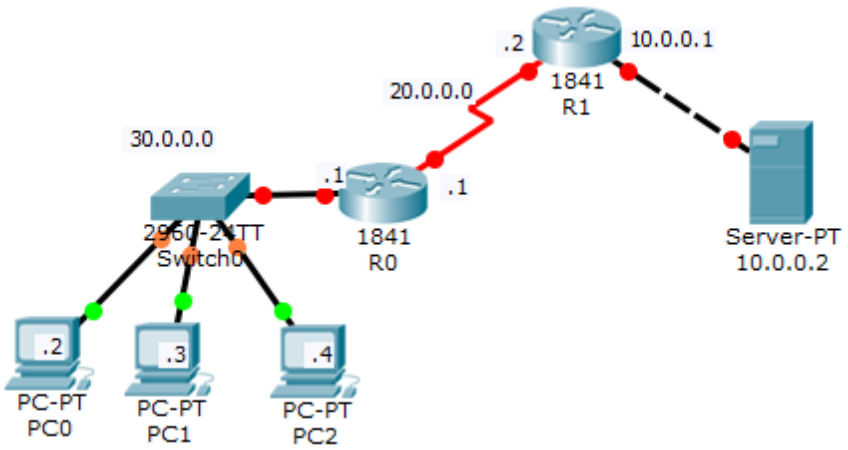
Learning Objectives:

- Introduction to NAT
- Differentiating between Public and Private IP Addresses
- Configuration Of Static NAT
- Configuration of Dynamic NAT
- Configuration of NAT via PAT

Configuration of static NAT is very straight forward. In this example we have a web server connected with Router 1. Our web server is using the IP address 10.0.0.2. But due to various reasons discussed in previous article our company want to use 50.0.0.1 IP address for this server. Now our task is to configure NAT on Router 1 which translate 10.0.0.2 [inside local web server address] to 50.0.0.1 [inside global ip address].

To configure static NAT follow this step by step guide

Either download this pre-configured topology or create your own topology as shown in this figure

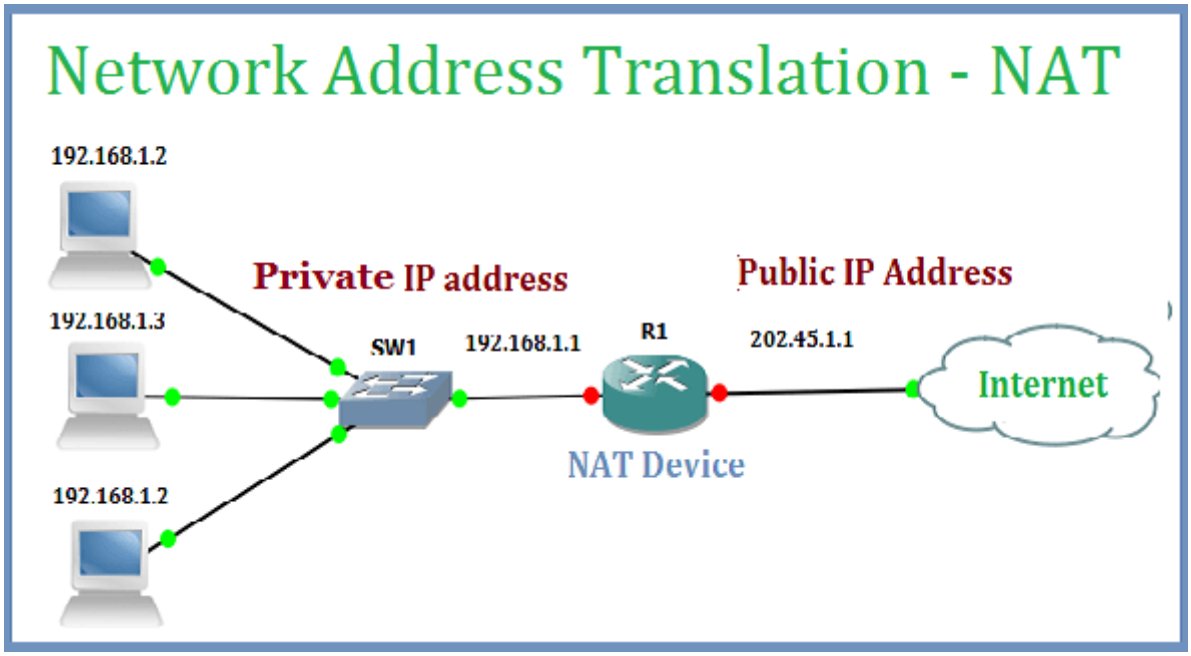


Configure ip address on PC and Server as given in figure

Now configure R1 exactly given here

Router>enable

```
Router#configure terminal
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
```



```
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.2 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.1
R1(config)#ip nat inside source static 10.0.0.2 50.0.0.1
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#
```

Now configure R0 exactly given here

```
Router>enable
Router#configure terminal
Router(config)#hostname R0
R0(config)#interface fastethernet 0/0
R0(config-if)#ip address 30.0.0.1 255.0.0.0
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#interface serial 0/0/0
R0(config-if)#ip address 20.0.0.1 255.0.0.0
R0(config-if)#clock rate 64000
R0(config-if)#bandwidth 64
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2
R0(config)#
```

As you have seen in configuration there is not direct route for 10.0.0.2. So PC from network of 30.0.0.0 will never know about it. They will access 50.0.0.1 as the web server IP. To test it double click on any computer and ping from 50.0.0.1 and you will get replay.

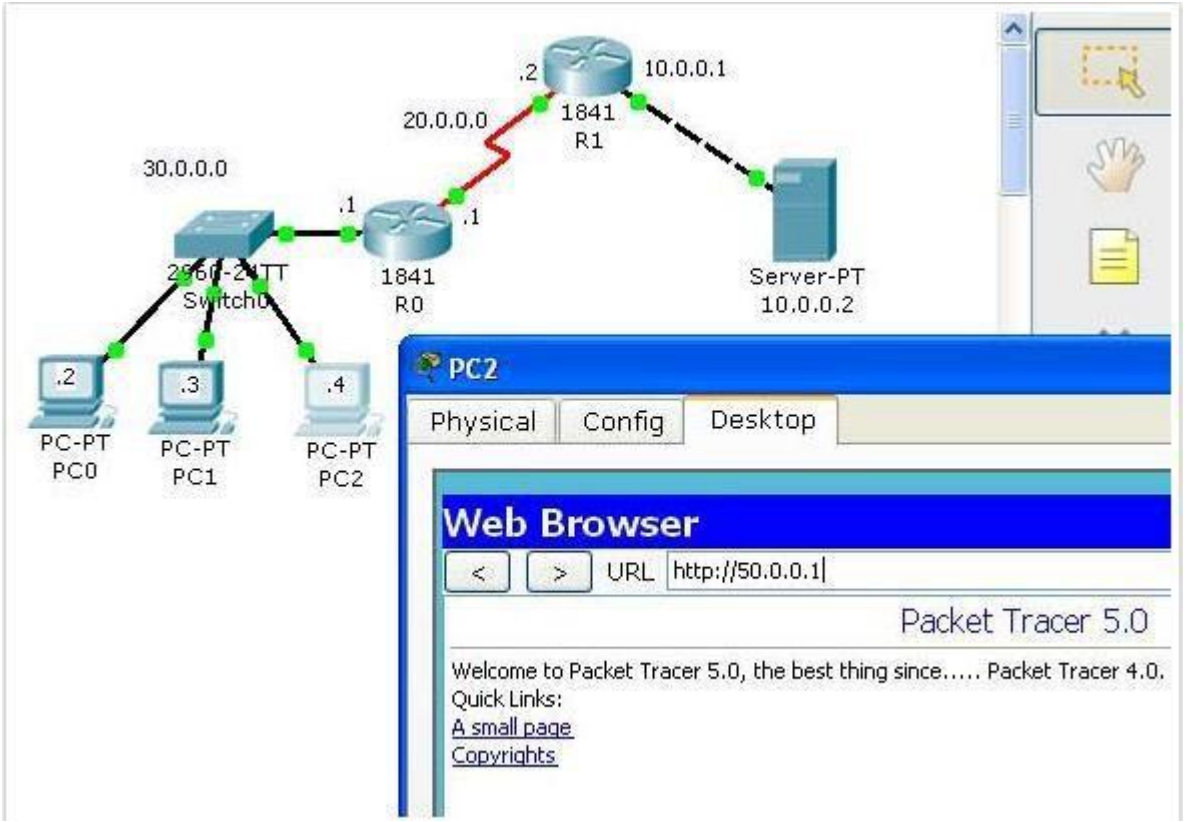
```
Packet Tracer PC Command Line 1.0
PC>ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:
Reply from 50.0.0.1: bytes=32 time=141ms TTL=126
Reply from 50.0.0.1: bytes=32 time=80ms TTL=126
Reply from 50.0.0.1: bytes=32 time=109ms TTL=126
Reply from 50.0.0.1: bytes=32 time=125ms TTL=126
Ping statistics for 50.0.0.1:
```

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 80ms, Maximum = 141ms, Average = 113ms

Now ping from 10.0.0.2 and you will get destination host unreachable error.

PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)

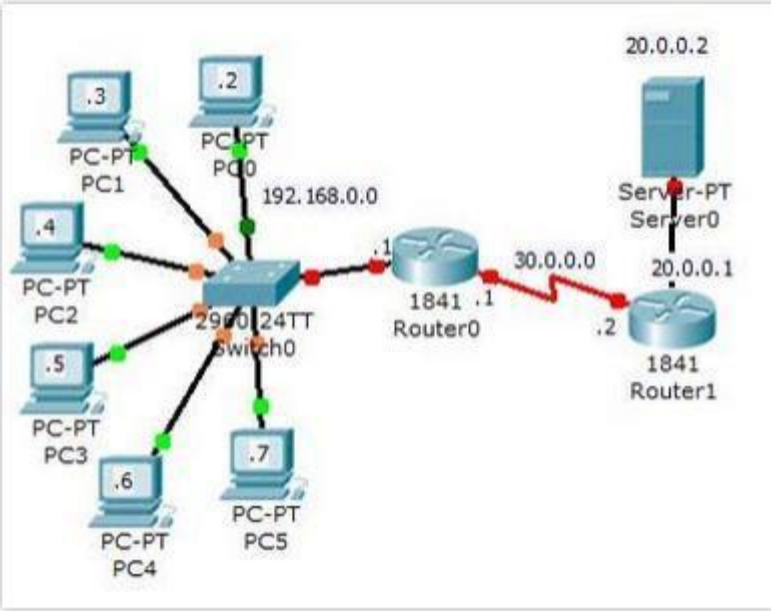
This demonstration show how the companies use NAT to hide their internal network from the outside of the world. Now open web browser from any PC in 30.0.0.0 network and brows the 50.0.0.1 site



As you can see in image you can easily access the 50.0.0.1
Step by Step Configuration of Dynamic NAT

With dynamic NAT, you must manually define two sets of addresses on your address translation device. One set defines which inside addresses are allowed to be translated (the local

addresses), and the other defines what these addresses are to be translated to (the global addresses).
For practice either download this pre created topology or create your own on packet tracer.



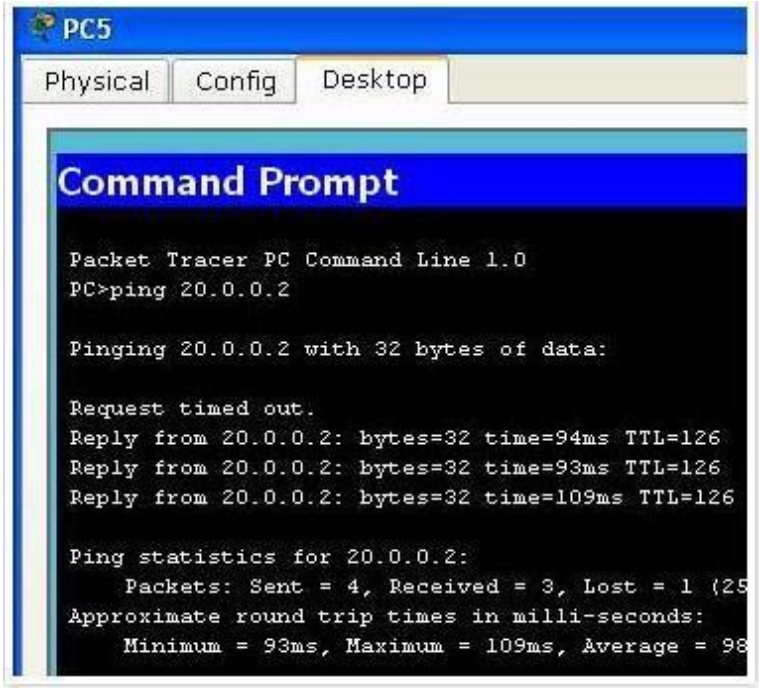
In this example our internal network is using 192.168.0.0 network. We have five public ip address 50.0.0.1 to 50.0.0.5 to use. **Router1 (1841 Router0)** is going to be NAT device. Double click on **Router1 (1841 Router0)** and configure it as given below:

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 192.168.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 30.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
R1(config)#ip nat pool test 50.0.0.1 50.0.0.5 netmask 255.0.0.0
R1(config)#ip nat inside source list 1 pool test
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/0
```

R1(config-if)#ip nat outside
R1(config-if)#exitR1(config)#exit

Now double click on R2(1841 Router1) and configure it as given below

Router>enable
Router#configure terminal
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
Router(config)#hostname R2
For testing of NAT go R1 and enable debug for NAT from privilege mode
R1#debug ip nat
Now go on pc and ping to 20.0.0.2



When ICMP ping packet reach to R1. It examines its source address against the access list 1. As this packet is generated form the network of 192.168.0.0 so it will pass the access list. Now router will check NAT pools for free address to translate with this address. Which you can check in the output of debag command in R1

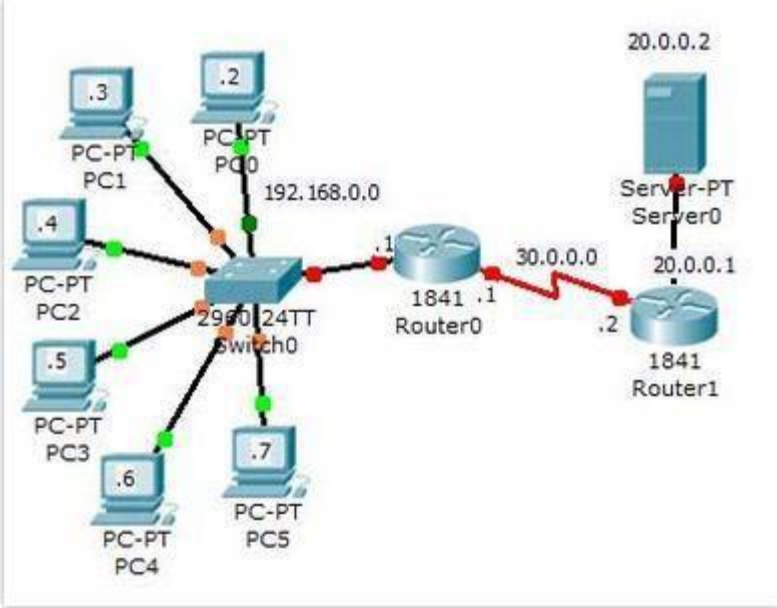
IP NAT debugging is on

NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
As you can see in output 192.168.0.5 is translate with 50.0.0.1 before leaving the router.
Now check for web access from any client pc

In real life its best practices to turn off debug after testing so go on Router 1 and turn off debug mode.
R1#no debug ip nat IP NAT debugging is off R1#

Step by Step Configuration of PAT

In dynamics Nat translations is made IP to IP. so you need as much global IP address as you have inside local address. That's an issue if you have few global IP address and hundred of inside local address to translate. In such a situation you need to use PAT.
For demonstration we are going to configure the same topology which we used in dynamic NAT but this time we are using only one global IP address 50.0.0.1



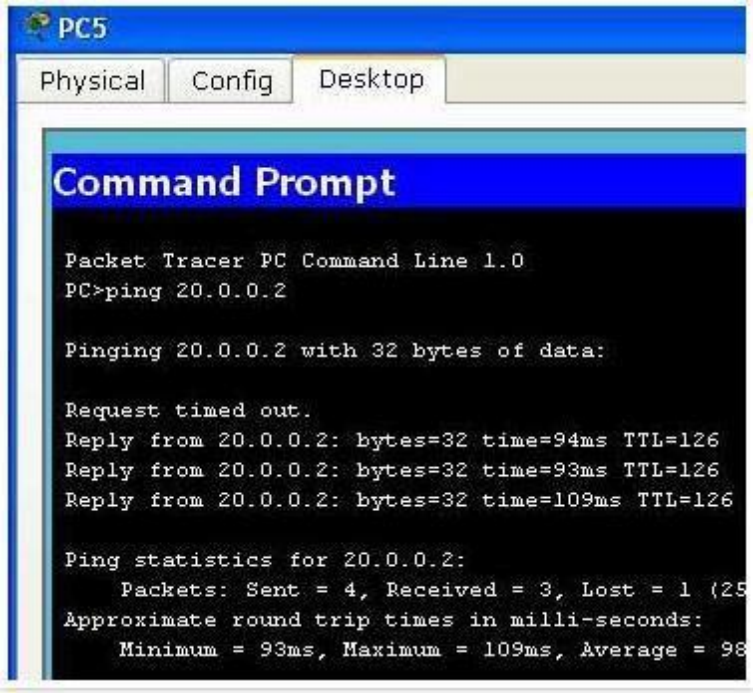
IP address of PC are already configured double click on R1 and configured it as given here
Now configure to R2 as given below

```
Router>enable
Router#configure terminal Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.0.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 30.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
R1(config)#ip nat pool test 50.0.0.1 50.0.0.1 netmask 255.0.0.0
R1(config)#ip nat inside source list 1 pool test overload
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#
```

Now configure to R2 as given below

```
Router>enable
Router#configure terminal
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#hostname R2
R2(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
```

Now go on pc and ping to 20.0.0.2



To verify PAT go on R1 and run show ip nat translations

```
R1#show ip nat translations
Pro Inside global Inside local Outside local Outside global
icmp 50.0.0.1:1 192.168.0.7:1 20.0.0.2:1 20.0.0.2:1
icmp 50.0.0.1:2 192.168.0.7:2 20.0.0.2:2 20.0.0.2:2
icmp 50.0.0.1:3 192.168.0.7:3 20.0.0.2:3 20.0.0.2:3
icmp 50.0.0.1:4 192.168.0.7:4 20.0.0.2:4 20.0.0.2:4
```

As you can see this time address translation is done with port address instead of IP

Lab Exercise will be provided by the instructor

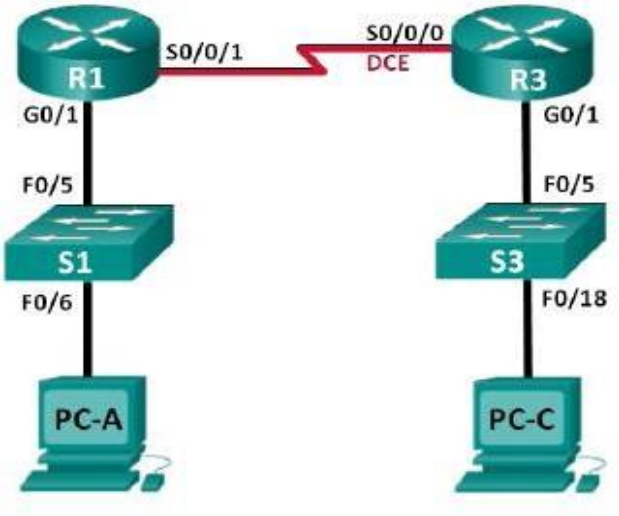
TASKS:

Tasks related to the lab will be provided by the lab instructor

LAB 11
IPv6 (Internet Protocol Version 6)

Configuring IPv6 Static and Default Routes

Topology



Addressing Table

Device	Interface	IPv6 Address / Prefix Length	Default Gateway
R1	G0/1	2001:DB8:ACAD:A::/64 eui-64	N/A
	S0/0/1	FC00::1/64	N/A
R3	G0/1	2001:DB8:ACAD:B::/64 eui-64	N/A
	S0/0/0	FC00::2/64	N/A
PC-A	NIC	SLAAC	SLAAC
PC-C	NIC	SLAAC	SLAAC

Learning Objectives

Part 1: Build the Network and Configure Basic Device Settings

- Enable IPv6 unicast routing and configure IPv6 addressing on the routers.
- Disable IPv4 addressing and enable IPv6 SLAAC for the PC network interfaces.
- Use **ipconfig** and **ping** to verify LAN connectivity.
- Use **show** commands to verify IPv6 settings.

Part 2: Configure IPv6 Static and Default Routes

- Configure a directly attached IPv6 static route.
- Configure a recursive IPv6 static route.
- Configure a default IPv6 static route.

Background / Scenario

In this lab, you will configure the entire network to communicate using only IPv6 addressing, including configuring the routers and PCs. You will use stateless address auto-configuration (SLAAC) for configuring the IPv6 addresses for the hosts. You will also configure IPv6 static and default routes on the routers to enable communication to remote networks that are not directly connected.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). The switches used are Cisco Catalyst 2960s with Cisco IOS Release 15.0(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Required Resources

• 2 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
• 2 Switches (Cisco 2960 with Cisco IOS Release 15.0(2) lanbasek9 image or comparable)
• 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
• Console cables to configure the Cisco IOS devices via the console ports
• Ethernet and serial cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will cable and configure the network to communicate using IPv6 addressing.

Step 1: Cable the network as shown in the topology diagram.

Step 2: Initialize and reload the routers and switches.

Step 3: Enable IPv6 unicast routing and configure IPv6 addressing on the routers.

a. Using Tera Term, console into the router labeled R1 in the topology diagram and assign the router the name R1.
b. Within global configuration mode, enable IPv6 routing on R1.

```
R1(config)# ipv6 unicast-routing
```

c. Configure the network interfaces on R1 with IPv6 addresses. Notice that IPv6 is enabled on each interface. The G0/1 interface has a globally routable unicast address and EUI-64

is used to create the interface identifier portion of the address. The S0/0/1 interface has a privately routable, unique-local address, which is recommended for point-to-point serial connections.

```
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:DB8:ACAD:A::/64 eui-64
R1(config-if)# no shutdown
R1(config-if)# interface serial 0/0/1
R1(config-if)# ipv6 address FC00::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
```

d. Assign a device name to router R3.

e. Within global configuration mode, enable IPv6 routing on R3.

```
R3(config)# ipv6 unicast-routing
```

f. Configure the network interfaces on R3 with IPv6 addresses. Notice that IPv6 is enabled on each interface. The G0/1 interface has a globally routable unicast address and EUI-64 is used to create the interface identifier portion of the address. The S0/0/0 interface has a privately routable, unique-local address, which is recommended for point-to-point serial connections. The clock rate is set because it is the DCE end of the serial cable.

```
R3(config)# interface gigabit 0/1
R3(config-if)# ipv6 address 2001:DB8:ACAD:B::/64 eui-64
R3(config-if)# no shutdown
R3(config-if)# interface serial 0/0/0
R3(config-if)# ipv6 address FC00::2/64
R3(config-if)# clock rate 128000
R3(config-if)# no shutdown
R3(config-if)# exit
```

Step 4: Disable IPv4 addressing and enable IPv6 SLAAC for the PC network interfaces.

a) On both PC-A and PC-C, navigate to the **Start** menu > **Control Panel**. Click the **Network and Sharing Center** link while viewing with icons. In the Network and Sharing Center window, click the **Change adapter settings** link on the left side of the window to open the Network Connections window.

b) In the Network Connections window, you see the icons for your network interface adapters. Double-click the Local Area Connection icon for the PC network interface that is connected to the switch. Click the **Properties** to open the Local Area Connection Properties dialogue window.

c) With the Local Area Connection Properties window open, scroll down through the items and uncheck the item **Internet Protocol Version 4 (TCP/IPv4)** check box to disable the IPv4 protocol on the network interface.

d) With the Local Area Connection Properties window still open, click the **Internet Protocol Version 6 (TCP/IPv6)** check box, and then click **Properties**.

e) With the Internet Protocol Version 6 (TCP/IPv6) Properties window open, check to see if the radio buttons for **Obtain an IPv6 address automatically** and **Obtain DNS server address automatically** are selected. If not, select them.

f) With the PCs configured to obtain an IPv6 address automatically, they will contact the routers to obtain the network subnet and gateway information, and auto-configure their IPv6 address information. In the next step, you will verify the settings.

Step 5: Use ipconfig and ping to verify LAN connectivity.

a. From PC-A, open a command prompt, type **ipconfig /all** and press Enter. The output should look similar to that shown below. In the output, you should see that the PC now has an IPv6 global unicast address, a link-local IPv6 address, and a link-local IPv6 default gateway address. You may also see a temporary IPv6 address and under the DNS server addresses, three site-local addresses that start with FEC0. Site-local addresses are private addresses that were meant to be backwards compatible with NAT. However, they are not supported in IPv6 and are replaced by unique-local addresses.

```
C:\Users\User1> ipconfig /all

Windows IP Configuration

<Output omitted>

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) 82577LC Gigabit Network Connection
Physical Address.....1C-C1-DE-91-C3-5D
DHCP Enabled.....No
Autoconfiguration Enabled.....Yes
IPv6 Address..... : 2001:db8:acad:a:7c0c:7493:218d:2f6c(Preferred)
Temporary IPv6 Address..... : 2001:db8:acad:a:bc40:133a:54e7:d497(Preferred)
Link-local IPv6 Address.....: fe80::7c0c:7493:218d:2f6c%13(Preferred)
Default Gateway.....: fe80::6273:5cff:fe0d:1a61%13
DNS Servers..... : fec0:0:0:ffff::1%1
fec0:0:0:ffff::2%1
```

fec0:0:0:ffff::3%1
NetBIOS over Tcpip ..... Disabled

Part 2: Configure IPv6 Static and Default Routes

In Part 2, you will configure IPv6 static and default routes three different ways. You will confirm that the routes have been added to the routing tables, and you will verify successful connectivity between PC-A and PC-C.

You will configure three types of IPv6 static routes:

- **Directly Connected IPv6 Static Route** – A directly connected static route is created when specifying the outgoing interface.
- **Recursive IPv6 Static Route** – A recursive static route is created when specifying the next-hop IP address. This method requires the router to execute a recursive lookup in the routing table in order to identify the outgoing interface.
- **Default IPv6 Static Route** – Similar to a quad zero IPv4 route, a default IPv6 static route is created by making the destination IPv6 prefix and prefix length all zeros, ::/0.

Step 1: Configure a directly connected IPv6 static route.

In a directly connected IPv6 static route, the route entry specifies the router outgoing interface. A directly connected static route is typically used with a point-to-point serial interface. To configure a directly attached IPv6 static route, use the following command format:

Router(config)# **ipv6 route** *<ipv6-prefix/prefix-length> <outgoing-interface-type> <outgoing-interface-number>*

a. On router R1, configure an IPv6 static route to the 2001:DB8:ACAD:B::/64 network on R3, using the R1 outgoing S0/0/1 interface.

```
R1(config)# ipv6 route 2001:DB8:ACAD:B::/64 serial 0/0/1
R1(config)#
```

b. View the IPv6 routing table to verify the new static route entry.

```
R1# show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C 2001:DB8:ACAD:A::/64 [0/0]
via GigabitEthernet0/1, directly connected
L 2001:DB8:ACAD:A:6273:5CFF:FE0D:1A61/128 [0/0]
via GigabitEthernet0/1, receive
S 2001:DB8:ACAD:B::/64 [1/0]
via Serial0/0/1, directly connected
```

C FC00::/64 [0/0]
via Serial0/0/1, directly connected
L FC00::1/128 [0/0]
via Serial0/0/1, receive
L FF00::/8 [0/0]
via Null0, receive

Step 2: Configure a recursive IPv6 static route.

In a recursive IPv6 static route, the route entry has the next-hop router IPv6 address. To configure a recursive IPv6 static route, use the following command format:

Router(config)# **ipv6 route** *<ipv6-prefix/prefix-length> <next-hop-ipv6-address>*

a. On router R1, delete the directly attached static route and add a recursive static route.

```
R1(config)# no ipv6 route 2001:DB8:ACAD:B::/64 serial 0/0/1
R1(config)# ipv6 route 2001:DB8:ACAD:B::/64 FC00::2
R1(config)# exit
```

b. On router R3, delete the directly attached static route and add a recursive static route.

```
R3(config)# no ipv6 route 2001:DB8:ACAD:A::/64 serial 0/0/0
R3(config)# ipv6 route 2001:DB8:ACAD:A::/64 FC00::1
R3(config)# exit
```

c. View the IPv6 routing table on R1 to verify the new static route entry.

```
R1# show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C 2001:DB8:ACAD:A::/64 [0/0]
via GigabitEthernet0/1, directly connected
L 2001:DB8:ACAD:A:6273:5CFF:FE0D:1A61/128 [0/0]
via GigabitEthernet0/1, receive
S 2001:DB8:ACAD:B::/64 [1/0]
via FC00::2
C FC00::/64 [0/0]
via Serial0/0/1, directly connected
L FC00::1/128 [0/0]
via Serial0/0/1, receive
L FF00::/8 [0/0]
                via Null0, receive
```

Device Configurations

Router R1

```
R1#show run
Building configuration...
Current configuration : 1222 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ipv6 unicast-routing
ipv6 cef
!
ip cef
        multilink bundle-name authenticated
!
!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
ipv6 address 2001:DB8:ACAD:A::/64 eui-64
```

```
ipv6 enable
!
interface Serial0/0/0
no ip address
shutdown
clock rate 2000000
!
interface Serial0/0/1
no ip address
ipv6 address FC00::1/64
ipv6 enable
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ipv6 route ::/0 Serial0/0/1
!
control-plane
!
line con 0
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport input all
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
line vty 0 4
login

transport input all
!
scheduler allocate 20000 1000
!
end
```

Router R3

```
R3#show run
Building configuration...
Current configuration : 1241 bytes
!
```

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
ipv6 unicast-routing
ipv6 cef
!
ip cef
multilink bundle-name authenticated
!
!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
no ip address
shutdown
duplex auto
speed auto
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
ipv6 address 2001:DB8:ACAD:B::/64 eui-64
ipv6 enable
!
interface Serial0/0/0
no ip address
ipv6 address FC00::2/64
ipv6 enable
clock rate 256000
!
interface Serial0/0/1
no ip address
```

```
shutdown
clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ipv6 route ::/0 Serial0/0/0
!
control-plane
!
line con 0
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport input all
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
line vty 0 4
login
transport input all
!
scheduler allocate 20000 1000
!
End
```

Lab 12

Open Ended Lab

Introduction:
An open-ended lab is where students are given the freedom to develop their own experiments, instead of merely following the already set guidelines from a lab manual or elsewhere. Making labs open-ended pushes students to think for themselves and think harder. The students here have to devise their own strategies and back them with explanations, theory and logical justification. This not only encourages students to come up with their experiments, but requires them to defend themselves and their experiment, if questioned.

Expected Outcomes:

- Problem analysis
- Topology configuration
- Needed security checks.
- Simulation

OEL Problem 1: Problem will be given by instructor
Aims:

- Use the router configuration
- Switch configuration
- Reliability
- Access control
- Complete network configuration

Sample Problem
Design and implement security on a data center. The datacenter has 3 servers, each of which host different services. The services on each of the servers are DNS, FTP and HTTPS. Each of the servers has unique public IP addresses. There are also 2 network administrators on the network who manages and troubleshoot any issues on the server.

- The datacenter servers and admins should be on different network.
- The servers should be accessible from the internet using public IP address.
- The servers should also have private IP address using which the admins access.

Problem Based Learning (PBL)

Introduction:

Problem-Based Learning (PBL) is a teaching method in which complex real-world problems are used as the vehicle to promote student learning of concepts and principles as opposed to direct

presentation of facts and concepts. In addition to course content, PBL can promote the development of critical thinking skills, problem-solving abilities, and communication skills. It can

also provide opportunities for working in groups, finding and evaluating research materials, and life-long learning (Duch et al, 2001).

OBJECTIVE

This is a problem based learning lab in which there is a freedom for devising one's own approach and method, by using the previous knowledge taught in lab, to solve the given problems.

PRE-LAB READING

Free to use knowledge from the previous labs and Courses.

GUIDELINES AND INSTRUCTIONS

Students are required to follow the following guidelines for this particular task.

1. Choose a Problem Based Learning (PBL) and submit your proposal by the end of 9th week with your lab instructor.
2. In case of any queries or issues discuss the problem with instructor before starting it. Submit in 11th week:
3. EER-diagram
   - Translation of the EER-diagram into a relational model (tables)
   - For each table identify functional dependencies, candidate keys and primary key as well as the normal form of the table. If the table is not in BCNF, motivate why you have not normalized it into this form.
4. Concise report submission in 15th week before 5:00 PM.
5. Final demonstration and viva in 15th week.

PROBLEM STATEMENT

NETWORK DESIGN PROPOSAL FOR AIRPORT

Project Description

The project is to design a proposal for setting up a network in an airport. The airport has three departments.

1. Airport authority
2. Flight service providers
3. Guests.

The airport authority maintains a server which handles the flight management controls. The flight service providers should have access only to the specific server in the airport authority network and not to any other systems. The guest users should have wireless access to a high speed internet connection, which should be shared among all the users in all the departments. The wireless access should be using a common password. The guest users should not have access to the other two departments. The users should obtain IP addresses automatically. The airport authority has 20 users, the flight service providers have 40 users and the maximum numbers of guests are estimated to be 100.

Networking Requirement

1. The active networking components (Routers, switches, wireless access points etc) with quantity.
2. The IP network design for each department.
3. Creating and mapping IP networks with vlans.
4. Analysis, identification and explanation of methodologies to use for access restriction and internet sharing.
5. Dynamic IP addressing design for all the networks.
6. Identify the configuration and features, wherever appropriate, which is required on the active components to setup the network.
7. Network topology diagram

Report Contents

1. Introduction
2. Networking Requirement
3. Network Design strategy
4. VLAN and IP Network Design
5. Requirement analysis of active networking components (Routers, switches, Access points, DHCP Server)
6. Network implementation plan
7. Network Topology Diagram
8.
    a. Network Configuration and guidelines a. Switch configuration (VLAN, Trunking)
    b. Router configuration (VLAN sub interface, Access lists) c. DHCP configuration (Scope creation with screen shot) d. Access point, server configuration guidelines
9. Hardware inventory list.

Report Format – PDF

Page max– 16

References

- CCNA Routing and Switching guide by Todd Lammle
- iliso.com/nat-packet-tracer-lab.htm
- tune.pk/.../setting-dhcp-dns-ftp-mail-server-dan-web-server-di-packet-tra...
- computernetworkingnotes.com/...vlan-stp-vtp.../configure-vlan-vtp.html
- computernetworkingnotes.com/switching.../basic-switch-configurations.h...
- tune.pk/.../packet-tracer-tutorial-connecting-and-configuring-2-routers-u...