



Birzeit University

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Manual for  
ENCS4130 Computer Networks Laboratory

July 2022

# Table of Experiments

## Network Review

EXP. No. 1. LAN Setup and Monitoring

EXP. No. 2. Router Configuration and Static Routing

EXP. No. 3. Dynamic Routing 1 (Distance Vector Routing Protocols) RIP & IGRP

EXP. No. 4. Dynamic Routing 2 (Link State Routing Protocols) OSPF

EXP. No. 5. Dynamic Routing 3 (Path Vector) BGP

EXP. No. 6. Access Lists

EXP. No. 7. Switching and VLANs 1 - Router on Stick

EXP. No. 8. Switching and VLANs 2 - Switch Virtual Interface

EXP. No. 9. Internet Protocol Version 6 (IPv6) Configuration

EXP. No. 10. Packet Sniffing and Domain Name System (DNS)



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **Network Review**

### **1. Objectives**

- ❖ Review network layering (OSI model).
- ❖ Introduce you to the network lab devices.
- ❖ Review the network subnetting

### **2. Introduction**

#### ***2.1. Open Systems Interconnection model (OSI model):***

It is a model that characterizes and standardizes the communication functions of telecommunication or computing system regardless to its underlying internal structure and technology. The model partitions a communication system into abstraction layers. The original version of the model had seven layers (see Table 0-1).

Table 0-1 OSI model by layer

Layer	Examples	Functions	Data to be sent	
Application Layer 7	<b>FTP DNS SMTP HTTP</b>	<b>Services used with end users' applications</b>	<b>Data</b>	<b>Hosts Layers (between hosts)</b>
Presentation Layer 6	<b>JPG, GIF SSL (HTTPS)</b>	<b>Formats the data to be viewed Encryption/decryption (security)</b>	<b>Data</b>	
Session Layer 5	<b>H322 that used for VOIP</b>	<b>Manage end-to-end connection between hosts</b>	<b>Data</b>	
Transport Layer 4	<b>TCP UDP</b>	<b>Ensure delivery of entire message</b>	<b>Segments</b>	
Network Layer 3	<b>IP RIP</b>	<b>Routing→path Forwarding→interface</b>	<b>Packets</b>	<b>Media Layers (over Network)</b>
Data Link Layer 2	<b>Ethernet MAC ARP</b>	<b>Physical addressing (MAC) Flow control</b>	<b>Frames</b>	
Physical Layer 1	<b>(Transmission media) Ethernet DSL</b>	<b>Signal Transmission</b>	<b>Bits</b>	








## ***2.2. Why layering***

- Troubleshooting: easier.
- Change: change in one-layer, other layers are not affected.
- Design: division into layers makes the solution much simple.
- Learning: understanding the network communication as layers is easier.

## ***2.3. Network Devices:***

There are many types of network devices used in building network topology. Some of them are shown in Table 0-2:

Table 0-2 Different Types of Network Devices

Device	Layer	Function	
Hub	<b>Layer 1 (Physical)</b>	<b>Dummy device (receiving information and send it to all connected devices)</b>	 Hub
Repeater	<b>Layer 1 (Physical)</b>	<b>Used to replicate the signal (make the signal stronger)</b>	 Repeater
DSL splitter	<b>Layer 1 (Physical)</b>	<b>Analog low-pass filter Used to split the signals between analog devices (such as analog modems) and a plain old telephone service (POTS) line</b>	 Splitter
Switch	<b>Layer 2 (Data link)</b>	<b>Self-learning (Receiving and sending frames to the correct destination)</b>	 Switch
Bridge	<b>Layer 2 (Data link)</b>	<b>Divides the LANs into Segments to: Reduce the traffic. Manage each segment separately It stores the MAC address for all devices in each segment and Broadcasts the received packets into the correct segment.</b>	 Bridge
Router	<b>Layer 3 (Network)</b>	<b>Routing: path from source to destination Forwarding: sending packets to the correct interface within the router.</b>	 Router
Multilayer switch (third layer switch)	<b>Both Layer 2 and Layer 3 (Data link and Network)</b>	<b>This device works as a switch if the sent data are in the same network and as a router if sent data from different networks</b>	 Multilayer Switch

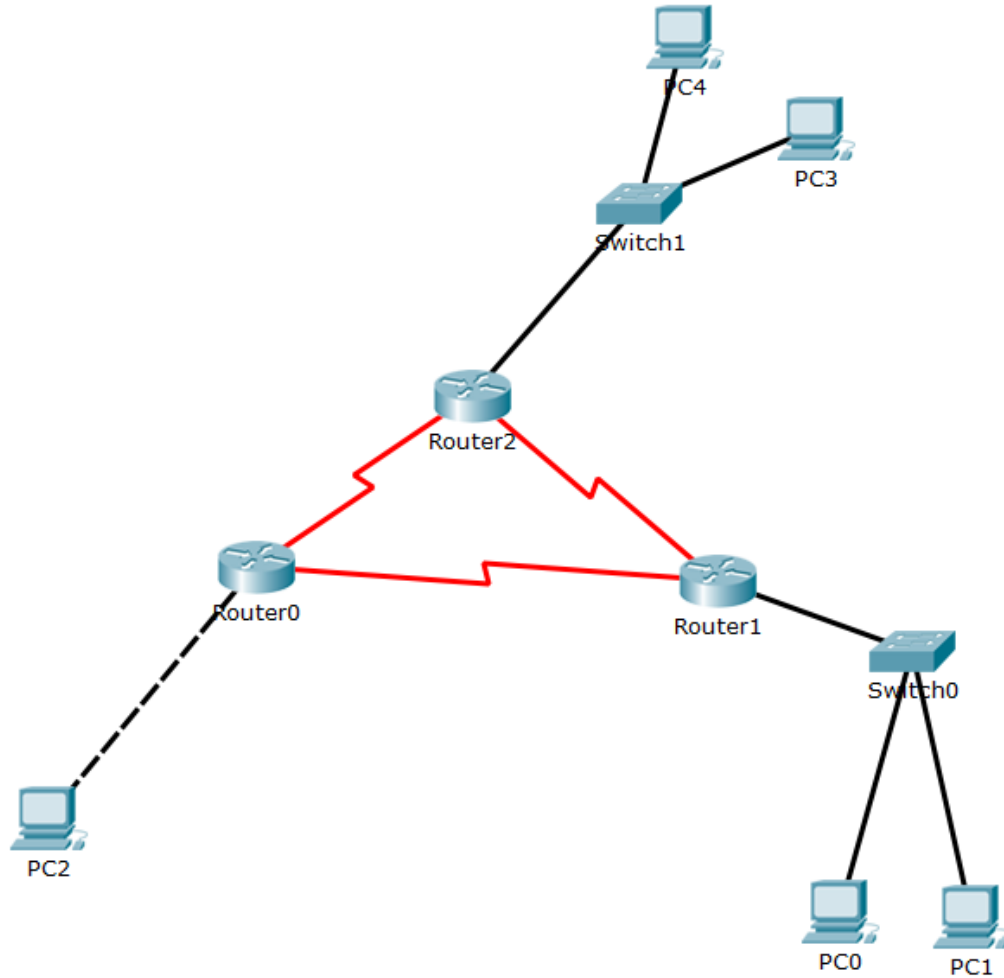
## 2.4. IP subnetting:

Each Network Interface Card (NIC or Network card) present in a PC is assigned one Network address called as IP address [or Network address]. This IP address is assigned by the administrator of the network. No two interfaces can have the same IP address on the same network. There is a burned-in address on the NIC called as Physical Address [or MAC address or Hardware address]. The MAC address of a network card indicates the vendor of that card and a unique serial number. IP addresses are divided into different classes. These classes determine the maximum number of hosts per network ID. Only three classes are used for network connectivity.

*Table 0-3 Classes of Networks*

<b>Address Class</b>	<b>IP Range</b>	<b>Bits for Subnet Mask</b>	<b>Subnet Mask</b>
Class A	<b>1.0.0.1 – 126.255.255.254</b>	<b>Left most 8 bits</b>	<b>255.0.0.0</b>
Class B	<b>128.0.0.1 – 191.255.255.254</b>	<b>Left most 16 bits</b>	<b>255.255.0.0</b>
Class C	<b>192.0.0.1-223.255.255.254</b>	<b>Left most 24 bits</b>	<b>255.255.255.0</b>

**2.5. How many networks are there in Figure 0-1**



*Figure 0-1 Network Topology*

6 networks



**2.6. Subnetting example TODO:**

Given the following the topology divide the given range 192.168.0.0/24 on the Networks A, B, C, D, E using minimum number of IPs.

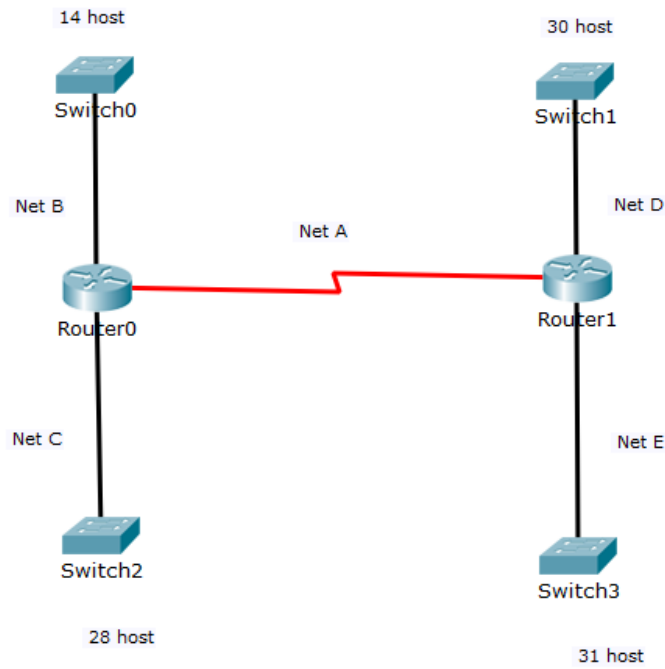


Figure 0-2 Network topology

Network Symbol	Network ID	Subnet mask	Wild card mask
A			
B			
C			
D			
E			





**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 1. LAN Setup and Monitoring**

### **1. Objectives**

- ❖ Study the types of Ethernet cabling and when and how to use them.
- ❖ Installing a Peer-to-Peer local area network (Workgroup LAN) using crossover cable.
- ❖ Learn to create a simple LAN with two PCs using an Ethernet switch and two
- ❖ straight-through cables to connect the workstations
- ❖ Learn to configure and verify the network connectivity.
- ❖ Implementing some applications like file sharing between workstations
- ❖ Learn about various network related commands.

### **2. Lab Requirements**

- ❖ 2 PCs with Network Interface Card (NIC) for each.
- ❖ One Ethernet hub and one Ethernet switch.
- ❖ Single crossover cable
- ❖ Two CAT5 straight-wired cables
- ❖ Cable tester.

### 3. Introduction

#### 3.1. Network Cables

There are many types of network cables used in the real-world applications. Some of them are given below:

- Unshielded twisted pair: As the name indicates, the wires are twisted with one another and there is no shield.



*Figure 1-1 Unshielded twisted pair*

- Shielded twisted pair: Shield with twisted pair.



*Figure 1-2 Shielded twisted pair*

- Coaxial cable: Like our TV cables.

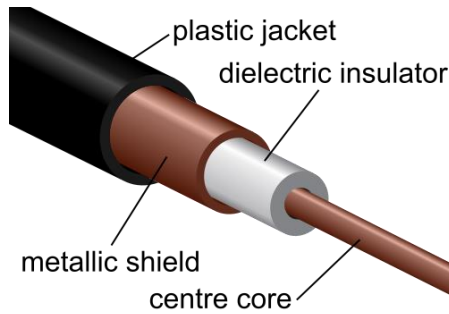


Figure 1-3 Coaxial cable

- Fiber-optic cable.

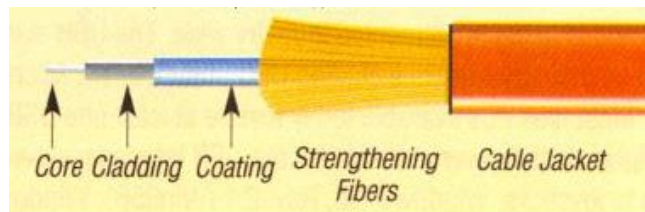


Figure 1-4 Fiber optic cable

Table 1-1 Comparison between network cables

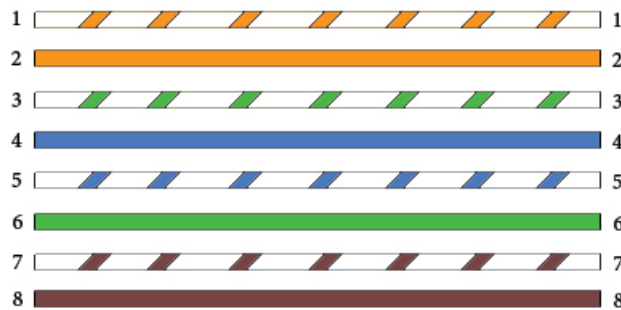
Type	Max distance	Bandwidth	Noise effect	Cost
Twisted pair	100 m	Up to 1-100 MHz	High noise	Cheap
Coaxial	100 m	Up to 3 GHz	Medium noise	Moderate
Fiber Optics	100 km	Up to THz	Less Noise	Expensive

### 3.3. Cable Connection for Network Devices

Since there is a bunch of different types of devices specified at the different layers of the OSI model, it is also very important to understand the many types of cables and connectors used for connecting all those devices to a network. We will go over cabling devices, discussing how to connect to a router or switch along with Ethernet LAN technologies. Ethernet cabling is an important discussion, especially if you are planning on build LAN.

➤ Straight Through Cable

Here, the connections are same on both the ends (*RJ45*) of the cable. This type of cable is used when we connect dissimilar devices [switch and router, router and hub, switch and PC, etc]. The wires and their respective pin numbers are shown in Figure 1-5.



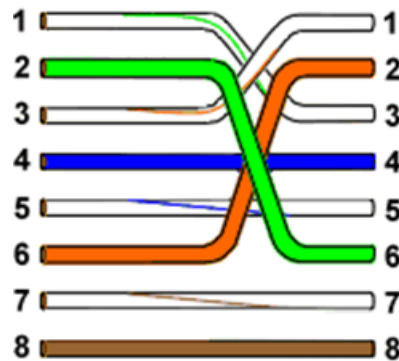
*Figure 1-5 Straight through cable*

Note that only pins 1, 2, 3, and 6 are used. Just connect 1 to 1, 2 to 2, 3 to 3, and 6 to 6, and you will be up and networking. However, remember that this would be an Ethernet- only cable and would not work with Voice, Token Ring, ISDN, etc.

➤ Cross-Over Cable

Here, the connections are different with a specific pattern in the *RJ45*. This type of cable is used when we connect similar devices [router and router, switch and switch, PC and PC, etc]

and with some exceptions [switch and hub, Router and PC]. The wires and their respective pin numbers are shown in Figure 1-6.

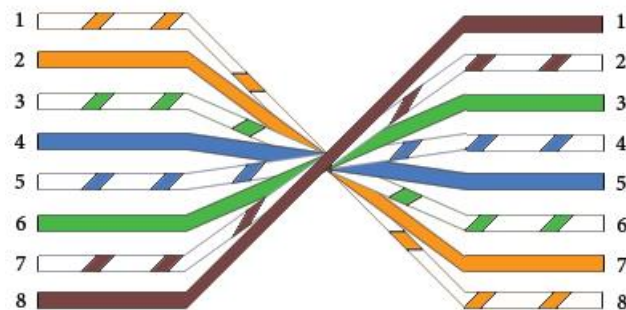


*Figure 1-6 Cross-Over cable*

The same four wires are used in this cable as in the straight-through cable, but we just connect different pins together. Notice that instead of connecting 1 to 1, etc. here we connect pins 1 to 3 and 2 to 6 on each side of the cable.

#### ➤ Roll-over Cable

Here, the connections are made in reverse order. This type of cable is used to connect the router/switch to the PC via console port for management purposes. As shown in Figure 1-7.



*Figure 1-7 Roll-Over cable*

## 4. Procedure

In this experiment, you will make a straight through cable and will use that cable to connect two PCs to create a simple Peer-to-Peer network. The instructions for this lab focus on the Windows 10 operating system.

### 4.1. Making Straight Through cable

In order to make a straight through cable we will need: Twisted pair cable, Two RJ-5, RJ45 crimping tool, Cable tester.

- Remove the outer plastic part from the twisted pair cable using the RJ45 crimping tool.
- Sort the inner cables as same as on both ends of the cable. All pins should have the same color wire on the same pin at both ends of the cable. (Pin 1 should match pin 1 and pin 8 should match pin 8 etc.)
- Cut of the cables in order to make them as the same length before inserting them to the RJ-45.
- Insert the cables inside the RJ-45 on both ends and make sure the cables are reaching the end of the RJ45.
- Use the RJ45 crimping tool to crimp the RJ45 with the cables. Note that when you crimp the RJ45 you will not be able to remove the cables.
- Verify the Cables by using the cable tester to make sure that all cables are connected in the correct order.

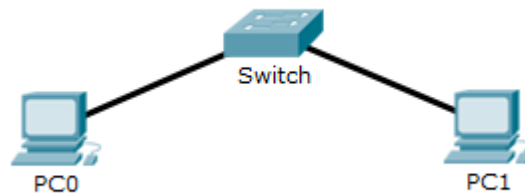
▪ **This part of the Lab is given by the instructor, please follow the steps above and any steps given to you by the instructor.**

▪ **Note that newly Operating systems and computers nowadays detect the cable types, so now there is no need to care a lot about the cable type.**



## 4.2. Connecting two PCs

The two PCs will be connected with a switch between them see Figure 1-8. Using a switch allows for more than just two workstations to be connected depending on the number of ports on the switch.



*Figure 1-8 Simple network topology*

### 4.2.1. Verify that the Network Interface Card (NIC) is installed for the two PCs

- From device manager verify that the NIC is installed in each PC.
- Click on the **Start** button at the lower left of the computer screen and select **“command prompt”**
- Type **“ping 127.0.0.1”** or **“ping localhost”**. This is the diagnostic or loopback address, and if you get a successful ping, your IP stack is then considered to be initialized and your Network Interface Card (NIC) card is functioning. If it fails, then there is a problem with the NIC card. This does not mean that a cable is plugged into the NIC, only that the IP protocol stack on the host can communicate to the NIC.

### 4.2.2. Plug in and Connect the Equipment

Plug the straight through cable from PC1 into port 1 of the switch and the cable from PC2 into port 2 of the switch. After the PCs have booted, check the green link light on the back of each NIC and the green lights on ports 1 and 2 of the switch to verify that they are communicating. This also verifies a good physical connection between the switch and the NICs in the PCs (OSI Layers 1 and 2). If the link light is not on it usually indicates a bad cable connection, an incorrectly wired cable or the NIC or switch may not be functioning correctly.

### 4.2.3. Configure an IP address and subnet mask for each computer manually

- At the desktop window, find the icon for the network labeled. Right click on this icon and select "**Open Network & Internet Settings**" as shown in Figure 1-9. This will open the network settings for the internet.

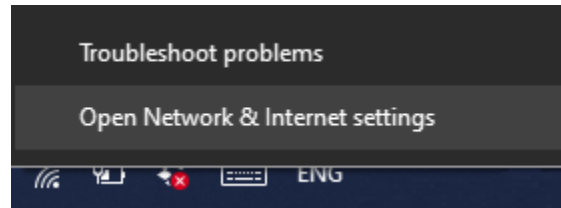


Figure 1-9 Open Network & Internet Settings

- Choose network and sharing center for the list as shown in Figure 1-10. This will open a new window for configuring the network settings

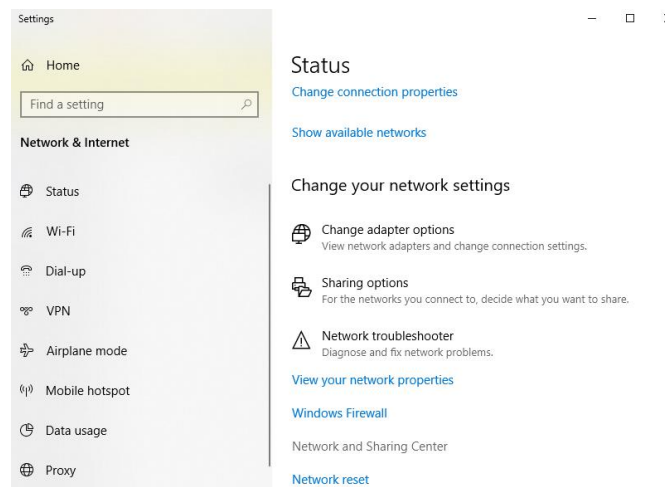


Figure 1-10 Network & Internet Settings

- From the Network and sharing center choose Ethernet as shown in Figure 1-10.



Figure 1-11 Network and Sharing center

- Choose properties for the ethernet settings as shown in, the choose “Internet Protocol Version 4 (TCP/IPv4)” and select “Use the following IP address”
- Set the IP address to be 192.168.0.1 and the Subnet mask to be 255.255.0.0. Clear the Default Gateway and DNS Server fields and click on OK for both windows. (what is the class of this IP address Class A or B or C).
- Verify that the IP Address for the computer has indeed changed. To do this, execute the **ipconfig/all** command again.
- Once each of you has set up the configuration correctly, it is time to verify that all computers are on the same network and can indeed communicate with each other.
- Repeat the steps for the second PC but set the IP to 192.168.0.2.

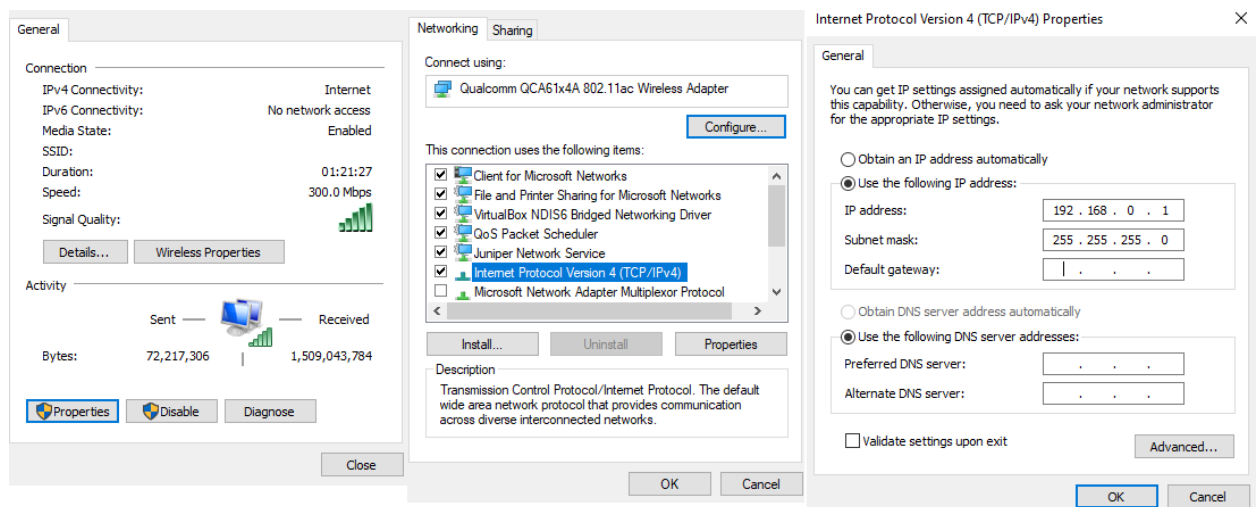


Figure 1-12 Setting IP address

#### 4.2.4. Verify Connectivity in Your Network with Ping

Use the **ping** command to check for basic TCP/IP connectivity. This will verify that you have a good OSI Layers 1 through 3 connections. Click on Start then “Command Prompt”. Enter the **ping** command followed by the IP address of the other workstation.

#### 4.3. Various Network Related Commands

To know and learn about various network related commands [**ping, tracert, netstat, at, net, route, arp**] and few definitions cum settings:

➤ **IPCONFIG Command**

This command is used to get IP configurations present in your PC.

➤ **PING Command**

Ping is a basic Internet program that lets you verify that a particular IP address exists and can accept requests. The verb ping means the act of using the ping utility or command. Ping is used diagnostically to ensure that a host computer you are trying to reach is actually operating. Various options available in the ping command (see windows help).

- Mention the difference between fragmenting and non-fragmenting packets.
- Test the reach ability towards a Ritaj server with fragmenting option enabled and limit the number of echos to 5.

➤ **TRACERT Command**

If someone would like to know how he goes from his house to his office, he could just tell the list of the crossroads where he passes. The same way we can ask the data sent over from your computer to the web server which way does it go, through which devices? We ask it by using the utility called **tracert**. In most computers today you can use this tool from the command line:

In UNIX machines it is called **tracert**, in MS Windows machines it is called **tracert**. Various options available in the ping command (see windows help).

- Find the route from your PC to [ritaj.birzeit.edu]
- Using the answers of the above, determine what is the first device your packet reaches to move from our network lab.

#### ➤ **Enhanced Ping**

**TJPing** tool is an excellent, widely acclaimed ping/lookup/tracert utility for Win95/98/Me/NT/2000/XP. It is fully configurable, multithreaded, and is very fast. All configuration options, hosts, and interface settings are remembered from session to session. Users can log all results to the file of their choice.

- Repeat the exercises provided to you in **ping** and **tracert** commands and store the result in a file for further reference.

#### ➤ **NETSTAT Command**

This command is used to get information about the open connections on your system (ports, protocols being used, etc.), incoming and outgoing data and also the ports of remote systems to which you are connected.

- Open a browser connection to http server [www.birzeit.edu] and write down the outcome of the command '**netstat -an**'.

## 5. Todo

How do you connect 2 computers without using hub or switch? Test the network during the lab.



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 2. Router Configuration and Static Routing**

### **1. Objectives**

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

### **2. Lab Requirements**

- ❖ Two Cisco routers.
- ❖ Two PCs.
- ❖ Two Cisco switches.
- ❖ Two CAT5 straight-wired cable.
- ❖ One Serial cable. (male and female).

### **3. Pre-lab**

- ❖ Install at home a simulator from CISCO called Packet Tracer version 6.2.

## 4. Introduction

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

### 4.1. Cisco Routers:

#### 4.1.1. The Cisco Router User Interface

The Cisco Internetwork Operating System (IOS) is the kernel of Cisco routers and most switches. A kernel is the basic part of an operating system that allocates resources and manages things, such as low-level hardware interfaces and security.

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

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

To access an interface the following command is used

```
interface <TYPE> <SLOT>/<PORT>
```

### 4.1.2. Connecting to a Cisco Router

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

#### ➤ The console port

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

#### ➤ Auxiliary port

You can also connect to a Cisco router through an auxiliary port which is really the same thing as a console port, so it follows that you can use it as one. But this auxiliary port also allows you to configure modem commands so that a modem can be connected to the router. This is a cool feature it lets you dial up a remote router and attach to the auxiliary port if the router is down and you need to configure it “out-of-band” (which means, basically, “out-of-the-network”).

#### ➤ Telnet

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

## 4.2. Routing

The term *routing* is used for taking a packet from one device and sending it through the network to another device on a different network. Routers do not really care about hosts—they only used to get packets to a network through a routed network, then the hardware address of the host is used to deliver the packet from a router to the correct destination host. If your network has



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

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

### ***4.3. Static Routing & Dynamic Routing***

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

#### ***4.3.1. Dynamic Routing***

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

#### ***4.3.2. Static Routing***

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

- Static routing has the following benefits:

- There is no overhead on the router CPU, which means you could possibly buy a cheaper router than if you were using dynamic routing.
  - There is no bandwidth usage between routers, which means you could possibly save money on WAN links.
  - It adds security because the administrator can choose to allow routing access to certain networks only.
- Static routing has the following disadvantages:
- The administrator must really understand the internetwork and how each router is connected in order to configure routes correctly.
  - If a network is added to the internetwork, the administrator has to add a route to it on all routers—by hand.
  - It is not feasible for large networks because maintaining it would be a full-time job in itself.

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

```
ip route <destination_network> <mask> <next-hop_address>
```

This list describes each command in the string:

- **ip route:** The command used to create the static route.
- **destination\_network:** The network you are placing in the routing table.
- **Mask:** The subnet mask being used on the network.
- **next-hop\_address:** The address of the next-hop router that will receive the packet and forward it to the remote network. This router interface's on a directly connected network. You must be able to ping the router interface before you add the route. If you type in the wrong nexthop address, or the interface to that router is down, the static route will show up in the router's configuration, but not in the routing table.

## 5. Procedure

In this lab, we will connect two routers and two PCs on different network. This will require configuring routing protocols between the routers. We will configure static routing which will be used as a routing protocol.

The IP address as follows: 192.X.10.0 → where X is : for example, student ID is 1224530, X = 30, and so the network will be 192.30.10.0/S.M

### 5.1. Building the Topology:

#### 5.1.1. Open Cisco packet tracer and set up the topology shown in Figure 2-1.

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

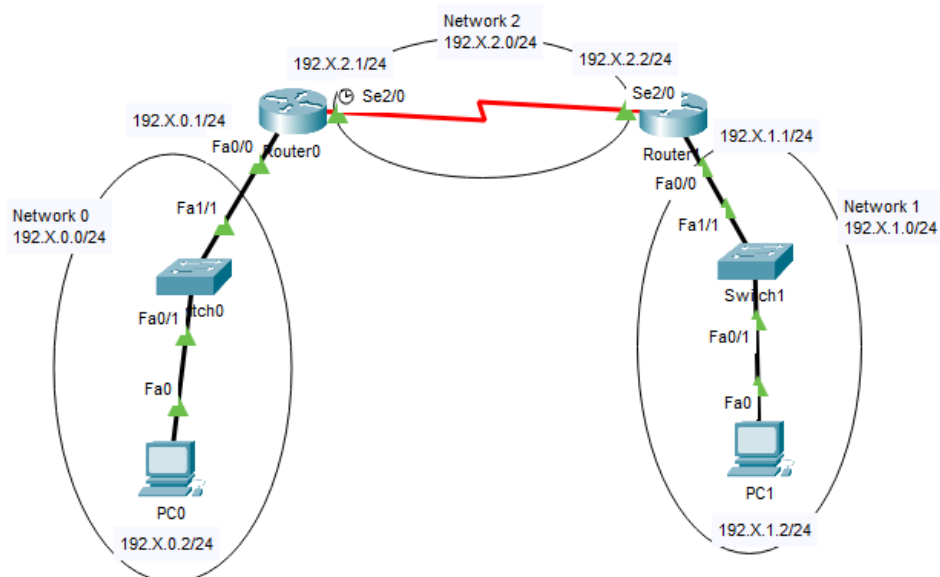


Figure 2-1 Topology

### 5.1.2. Configuring IPs for the PCs

- Click on the PC0 and go to desktop tab
- Choose IP configuration to add an IP address for the PC as shown in Figure 2-2

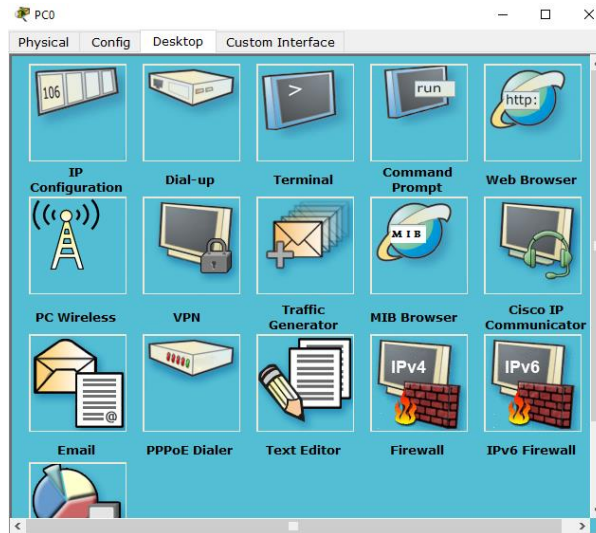


Figure 2-2 PC Desktop

- Add the following IP address (192. X.0.2/24) as shown in Figure 2-3.

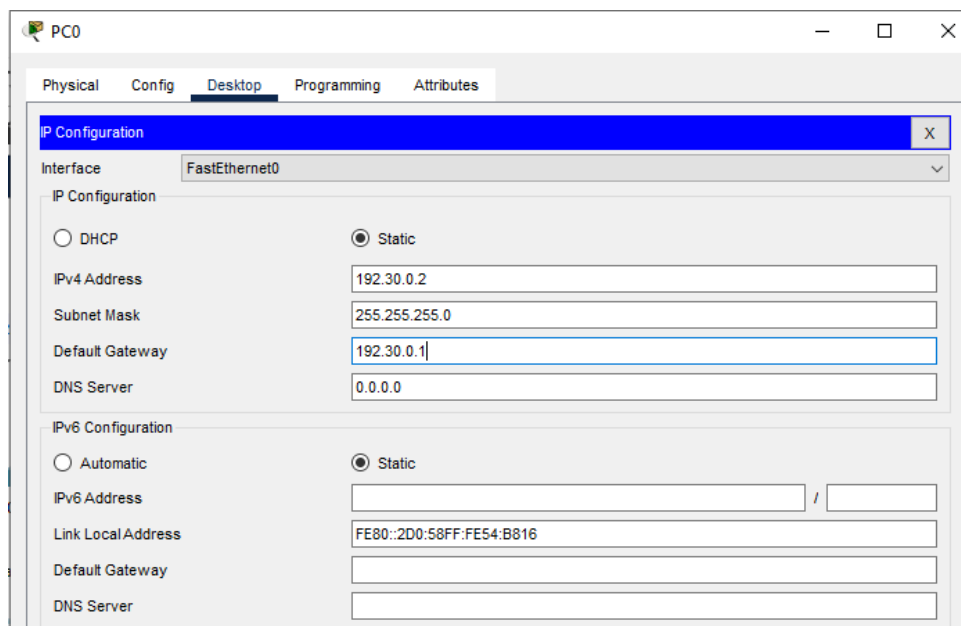


Figure 2-3 PC0 IP address

➤ Repeat the previous steps for PC1 with the IP address **192. X.1.2/24**

▪ **Note that subnet mask /24 is equivalent to 255.255.255.0**

### 5.1.3. Configuring IPs for the Routers

The router IPs and subnet masks for each interface are shown in the Table 2-1.

Table 2-1 Routers IPs

Router	Interface	IP address	Subnet mask
Router 0	Fa0/0	192.X.0.1	255.255.255.0
	Se2/0	192. X.2.2	255.255.255.0
Router 1	Fa0/0	192. X.1.1	255.255.255.0
	Se2/0	192. X.2.1	255.255.255.0

A. Click on router 0 to configure the router.

B. Go to CLI tab as shown in Figure 2-4.

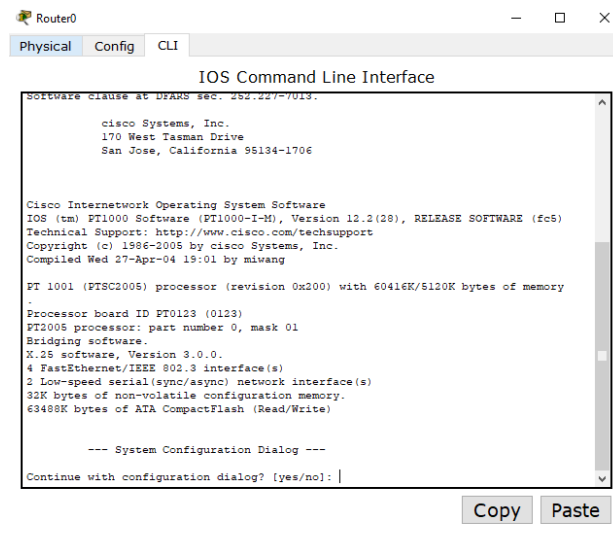


Figure 2-4 Router 0 CLI

### C. Setup Mode and Command Line interface mode (CLI)

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

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

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

```
Continue with configuration dialog? [yes/no]: no
Press RETURN to get started!
Router>
```

### D. Logging into the Router

After the interface status messages appear and you press Enter, the **Router>** prompt will appear. This is called user exec mode (user mode) and is mostly used to view statistics, but it is also a stepping-stone to logging into privileged mode. You can only view and change the configuration of a Cisco router in privileged exec mode (privileged mode), which you get into with the enable command. Here is how you would do that:

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

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

```
Router#disable
Router>
```

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

```
Router>logout
```

After trying all these commands put the Cisco router in privileged exec mode (privileged mode). Using the enable command.

### E. Overview of Router Modes

To configure from a CLI, you can make global changes to the router by typing configure terminal (or **config t** for short), which puts you in global configuration mode and changes what is known as the running-config. A global command (a command run from global config) is one that is set once and affects the entire router.

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

```
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

At this point, you make changes that affect the router as a whole, hence the term global configuration mode. To change the running-config—the current configuration running in dynamic RAM (DRAM) you use the configure terminal.

### F. Router Interfaces

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

```
interface <TYPE> <SLOT>/<PORT>
```

as seen here:

```
Router(config)#interface fastethernet 0/0
Router(config-if)#
```

Interface configuration is one of the most important router configurations, because without interfaces, a router is a totally useless. In addition, interface configurations must be exact to enable communication with other devices. Some of the configurations used to configure an interface are Network layer addresses, media type, bandwidth, and other administrator commands. Different routers use different methods to choose the interfaces used on them.

To exit interface you can type the command:

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

Now it is time to choose the interface you want to configure. Once you do that, you will be in interface configuration for that specific interface. (choose the interface Fa0/0).

### G. Bringing Up an Interface

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

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

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
```

Now the interface is up, we can add an IP address to that interface.

### H. Configuring the first IP Address on an interface

Even though you do not have to use IP on your routers, it is most often what people use. To configure IP addresses on an interface, use the command

```
ip address <IP-ADDRESS> <SUBNET-MASK>
```



from interface configuration mode for example to add the IP address 192.X.0.1/24 to the interface Fa0/0 use the command:

```
Router(config-if)#ip address 192.X.0.1 255.255.255.0
```

Make sure that you are in the correct interface before adding the IP address.

- I. **Now repeat Steps F – H for Router 0 and the second interface (Se2/0) with IP shown in Table 2-1.**
- J. **Now repeat Steps A – I for Router 1 using the correct IP addresses shown in Table 2-1.**

#### ***5.1.4. Configuring Static Routing***

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

```
ip route <destination_network> <mask> <next-hop_address>
```

To configure Router 0 to network 192.X.1.0/24 the following command is used:

**destination\_network** the network Router 0 is not connected to (192.X.1.0).

**mask** the subnet mask being used on the destination network /24 (255.255.255.0)

**next-hop\_address** is the address of Se2/0 on Router 1 (192.X.2.1)

```
Router(config)#ip route 192.X.1.0 255.255.255.0 192.X.2.1
```

- **Now repeat this step 5.1.4 for router 1 with the correct addresses.**

#### ***5.1.5. Showing all router configuration to make sure everything is good***

Go the router setup and write the command

```
Router#show running-config
```

This will show you all the configuration for the router you have made.

### 5.1.6. Showing the routing table

To show the routing table for each router type

```
Router#show ip route
```

This will show you all the routing paths for the router

## 5.2. Verifying Your Configuration

Once all the routers' routing tables are configured, they need to be verified. The best way to do this, besides using the **show ip route** command, is with the **ping** program. By pinging from routers PC0 and PC1, the whole internetwork will be tested end-to-end.

Open PC0 and go to desktop>command\_prompt and type

```
Ping <IP-ADDRESS>
```

```
Ex: Ping 192.X.1.2
```

If the ping is working correctly then everything is configured well.

## 5.3. Other Important Configurations:

### 5.3.1. Editing and Help Features

You can use the Cisco advanced editing features to help you configure your router. If you type in a question mark (?) at any prompt. Here is a shortcut: To find commands that start with a certain letter, use the letter and the question mark with no space between them:

```
Router#c?
```

```
clear clock configures connect copy
```

```
Router#c
```

By typing **c?**, we received a response listing all the commands that start with *c*. To find the next command in a string, type the first command and then a question mark:

```
Router#clock ?
```

```
set Set the time and date
```

```
Router#clock set ?
hh:mm:ss Current Time
```

```
Router#clock set 10:30:10 ?
<1-31> Day of the month
```

```
MONTH Month of the year
```

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

```
MONTH Month of the year
```

```
Router#clock set 10:30:10 28 january ?
<1993-2035> Year
```

```
Router#clock set 10:30:10 28 january 2020 ?
<cr>
```

```
Router#
```

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

*Table 2-2 Enhanced Editing Command*

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

### 5.3.2. *Gathering Basic Routing Information*

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

```
Router#sh version
```

### 5.3.3. *Serial Interface Commands*

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

```
Router#config t
Router(config)#int s1
Router(config-if)#clock rate ?
Router(config-if)#clock rate 64000
```

Here is an example of using the bandwidth command:

```
Router(config-if)#bandwidth <BANDWIDTH-IN-KILOBITS>
Router(config-if)#bandwidth 64
```

### 5.3.4. *Hostnames*

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

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#hostname RouterB
RouterB(config)#
```

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

### 5.3.5. Passwords

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

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

- A. Set your enable secret password by typing `enable secret cisco` (the third word `cisco` should be your own personalized password) and pressing Enter.
- B. Now let us see what happens when you log all the way out of the router and then log in. Log out by pressing **Ctrl+Z**, then type `exit` and press Enter. Go to privileged mode. Before you are allowed to enter privileged mode, you will be asked for a password. If you successfully enter the secret password, you can proceed.
- C. Remove the secret password. Go to privileged mode, type **config t**, and press Enter. Type `no enable secret` and press Enter. Log out and then log back in again, and now you should not be asked for a password.
- D. To set the Telnet or VTY password, type `line vty 0 4` and then press Enter. The **0 4** is the range of the five available virtual lines used to connect with Telnet. If you have an enterprise IOS, the number of lines may vary. Use the question mark to determine the last line number available on your router.
- E. One more command you need to set for your VTY password is `password`. Type `password cisco` to set the password. (`cisco` is your password.)
- F. Here is an example of how to set the VTY passwords:

```
Router#config t
Router(config)#line vty 0 4
Router(config-line)#password cisco
Router(config-line)#login
```

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

### ***5.3.6. Viewing and Saving Configurations***

If you run through setup mode, you will be asked if you want to use the configuration you just created. If you say Yes, then it will copy the configuration running in DRAM, (known as the running-config), into NVRAM, and name the file startup-config. You can manually save the file from DRAM to NVRAM by using the copy runningconfig startupconfig command (you can use the shortcut copy run start also):

```
Router#copy run start
```

Also you can type command:

```
Router#write
```

This will save your configuration even when shutting down your router.

## **6. Todo:**

This part will be given to you by the instructor.



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 3. Dynamic Routing 1 (Distance Vector Routing Protocols)**

### **RIP & EIGRP**

#### **1. Objectives**

- ❖ Learn how to configure and verify IP routing with Cisco routers.
- ❖ Dynamic routing RIP and EIGRP

#### **2. Lab Requirements**

- ❖ Three Cisco routers.
- ❖ Five PCs.
- ❖ Two Cisco switches.
- ❖ Several CAT5 straight-wired cables.
- ❖ Two Serial cable. (male and female).

#### **3. Introduction**

There are two main routing classes used in data communication networks. The first class is called distance vector routing protocol and the second one is called link state routing protocol.

Routing Information Protocol (RIP) and Enhanced Interior Gateway Routing Protocol (EIGRP), which will be used in this experiment, are distance vector routing protocols. An example of the other category is Open Shortest Path First (OSPF). In distance vector routing Protocols at the beginning each node (router) has only routing information about its direct neighbors. Each router broadcast periodically its routing information to its neighbors. This way, eventually, each node will get information about the entire network. When a node goes down, the direct neighbors will update their routing information and then update their neighbors using the periodic broadcasts and so on, until all nodes in the network knows about this change.

### ***3.1. Dynamic Routing***

Dynamic routing is when protocols are used to find networks and update routing tables on routers. True—this is easier than using static or default routing, but it will cost you in terms of router CPU processes and bandwidth on the network links. A routing protocol defines the set of rules used by a router when it communicates routing information between neighbor routers. The two routing protocols we will introduce about in this Lab are Routing Information Protocol (RIP) and Enhanced Interior Gateway Routing Protocol (EIGRP).

### ***3.2. Routing Protocol Basics***

There are some important things you should know about routing protocols before getting deeper into RIP. Specifically, you need to understand administrative distances, the three different kinds of routing protocols, and routing loops:

#### ***3.2.1. Administrative Distances***

The administrative distance (AD) is used to rate the trustworthiness of routing information received on a router from a neighbor router. An administrative distance is an integer from 0 to 255, where 0 is the most trusted and 255 means no traffic will be passed via this route. If a router receives two updates listing the same remote network, the first thing the router checks is the AD. If one of the advertised routes has a lower AD than the other, then the route with the lowest AD will be placed in the routing table. If both advertised routes to the same network have the same



AD, then routing protocol metrics (such as hop count or bandwidth of the lines) will be used to find the best path to the remote network. The advertised route with the lowest metric will be placed in the routing table. But if both advertised routes have the same AD as well as the same metrics, then the routing protocol will load-balance to the remote network (which means that it sends packets down each link).

Table 3-1 shows the default administrative distances that a Cisco router uses to decide which route to take to a remote network.

*Table 3-1 Administrative Distance for different routing protocols*

<b>Route Source</b>	<b>Administrative Distance (AD)</b>
Connected interface (directly)	0
Static route	1
IGRP	100
EIGRP	90
RIP	120

If a network is directly connected, the router will always use the interface connected to the network. If an administrator configures a static route, the router will believe that route over any other learned routes. You can change the administrative distance of static routes, but, by default, they have an AD of 1. If you have a static route, a RIP-advertised route, and an IGRP-advertised route listing the same network, then by default, the router will always use the static route unless you change the AD of the static route.

### ***3.2.2. Distance-Vector Routing Protocols***

The distance-vector protocols find the best path to a remote network by judging distance. Each time a packet goes through a router that has called a hop. The route with the least number of hops to the network is determined to be the best route. The vector indicates the direction to the remote network. Both RIP and IGRP are distance-vector routing protocols.

They send the entire routing table to directly connect neighbors. The distance-vector routing algorithm passes complete routing table contents to neighboring routers, which then

combine the received routing table entries with their own routing tables to complete the router's routing table. This is called routing by rumor, because a router receiving an update from a neighbor router believes the information about remote networks without finding out for itself.

It is possible to have a network that has multiple links to the same remote network, and if that is the case, the administrative distance is checked first. If the AD is the same, the protocol will have to use other metrics to determine the best path to use to that remote network.

### ***3.3. Routing Information Protocol (RIP)***

It uses only hop count to determine the best path to a network. If RIP finds more than one link to the same remote network with the same hop count, it will automatically perform a round-robin load balancing. RIP can perform load balancing for up to six equal-cost links (four by default) and it uses classful subnetting.

To configure RIP routing, just turn on the protocol with the:

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

That tell the RIP routing protocol which networks to advertise. Now to configure our router internetwork with RIP.

```
Router(config-router)#network <ID-OF-CONNECTED-NETWORKS>
```

### ***3.4. Enhanced Interior Gateway Routing Protocol (EIGRP)***

EIGRP is a Cisco-proprietary distance-vector routing protocol. This means that all your routers must be Cisco routers to use EIGRP in your network. Cisco created this routing protocol to overcome the problems associated with RIP.

EIGRP has a maximum hop count of 255 with a default of 100. This is helpful in larger networks and solves the problem of 15 hops being the maximum possible in a RIP network. EIGRP also uses a different metric than RIP. EIGRP uses bandwidth and delay of the line by default as a metric for determining the best route to an internetwork. This is called a composite metric.

Reliability, load, and maximum transmission unit (MTU) can also be used, although they are not used by default.

The main difference between RIP and EIGRP configuration is that when you configure IGRP, you supply the autonomous system number. All routers must use the same number in order to share routing table information. Here is a list of EIGRP characteristics that you won't find in RIP:

- EIGRP can be used in large Internetworks
- EIGRP uses an Autonomous System number for activation
- EIGRP gives a full route table update every 90 seconds
- EIGRP uses bandwidth and delay of the line as metric (lowest composite metric)

Here is how to turn on EIGRP routing:

```
Router(config)#router EIGRP <AS>  
Router(config-router)#network <ID-OF-CONNECTED-NETWORKS>
```

## 4. Procedure

In this lab, we will connect three routers and several PCs on different networks. This will require configuring routing protocols between the routers. We will configure dynamic routing (Rip) which will be used as a routing protocol.

### 4.1. Building the topology

Build the topology shown in Figure 3-1

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

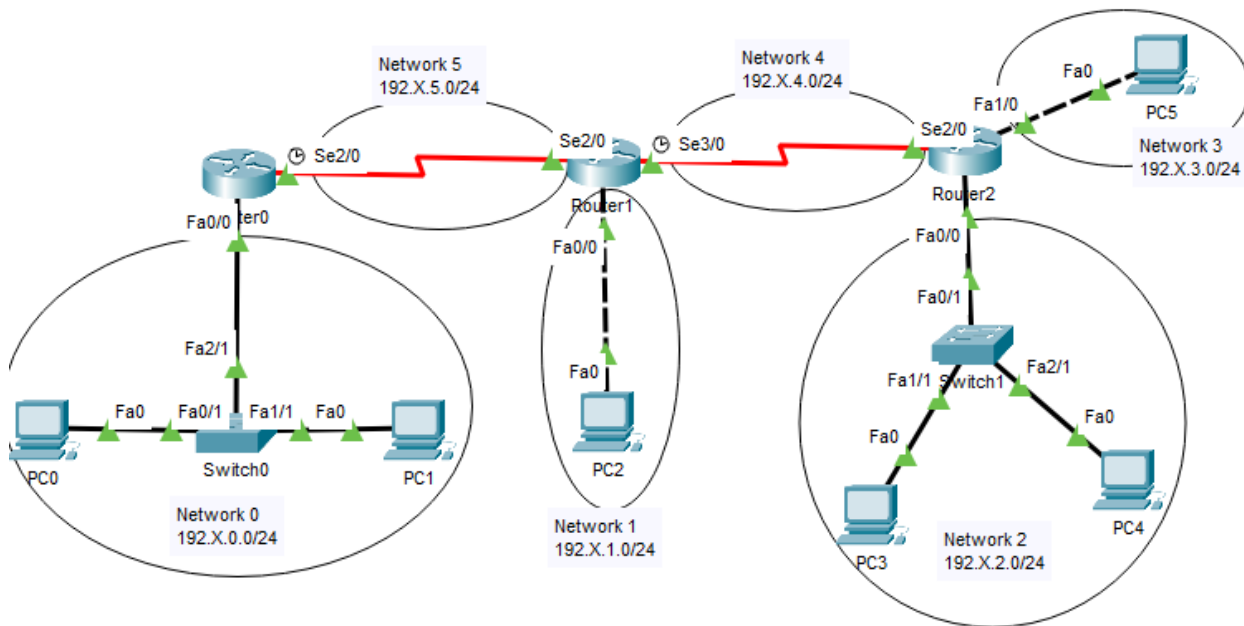


Figure 3-1 Topology

Use the following IPs shown in Table 3-2 for the configuration. In order to configure the IPs for the PCs and Routers see EXP. No. 2 sections 5.1.2 & 5.1.3

*Table 3-2 IPs for all networks and devices*

<b>Network Number And Network ID</b>	<b>Device</b>	<b>Port</b>	<b>IP</b>	<b>Subnet Mask</b>
Network 0 192.X.0.0/24	Router 0	Fa0/0	192.X.0.1	255.255.255.0
	PC0	Fa0	192.X.0.2	255.255.255.0
	PC1	Fa0	192.X.0.3	255.255.255.0
Network 1 192.X.1.0/24	Router 1	Fa0/0	192.X.1.1	255.255.255.0
	PC2	Fa0	192.X.1.2	255.255.255.0
Network 2 192.X.2.0/24	Router 2	Fa0/0	192.X.2.1	255.255.255.0
	PC3	Fa0	192.X.2.2	255.255.255.0
	PC4	Fa0	192.X.2.3	255.255.255.0
Network 3 192.X.3.0/24	Router 2	Fa1/0	192.X.3.1	255.255.255.0
	PC5	Fa0	192.X.3.2	255.255.255.0
Network 4 192.X.4.0/24	Router 1	Se3/0	192.X.4.1	255.255.255.0
	Router 2	Se2/0	192.X.4.2	255.255.255.0
Network 5 192.X.5.0/24	Router 0	Se2/0	192.X.5.1	255.255.255.0
	Router 1	Se2/0	192.X.5.2	255.255.255.0

## 4.2. Configuring RIP Routing

To configure RIP routing, just turn on the protocol with the:

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

That tell the RIP routing protocol which networks to advertise. Let us configure our three routers with RIP routing and practice that.

```
Router(config-router)#network <ID-OF-CONNECTED-NETWORKS>
```

#### Router 0:

```
Router(config)#router rip  
Router(config-router)#network 192.X.0.0  
Router(config-router)#network 192.X.5.0
```

#### Router 1:

```
Router(config)#router rip  
Router(config-router)#network 192.X.1.0  
Router(config-router)#network 192.X.4.0  
Router(config-router)#network 192.X.5.0
```

#### Router 2:

```
Router(config)#router rip  
Router(config-router)#network 192.X.2.0  
Router(config-router)#network 192.X.3.0  
Router(config-router)#network 192.X.4.0
```

RIP has an administrative distance of 120. Static routes have an administrative distance of 1 by default. The routing tables will not be propagated with RIP information if static routing is configured. So, the first thing you need to do is to delete the static routes off each router if you have any.

Note the fact that you need to type in every directly connected network that you want RIP to advertise. But because they are not directly connected, we are going to leave out some networks for the RIP which is its job to find them and populate the routing table.

Two or three commands, and you are done—sure makes your job a lot easier than when using static routes. However, keep in mind the extra router CPU process and bandwidth that you are consuming.

RIP and EIGRP use the classful address when configuring the network address. Because of this, all subnet masks must be the same on all devices in the network (this is called classful routing).

### 4.3. Verifying the RIP Routing Tables

Each routing table should now have the routers' directly connected routes as well as RIP injected routes received from neighboring routers. This output shows us the contents of the Router routing table:

```
Router#sh ip route
R 192.X.1.0 [120/1] via 192.168.5.2, 00:00:23, Serial2/0
R 192.X.2.0 [120/2] via 192.168.5.2, 00:00:23, Serial2/0
R 192.X.3.0 [120/2] via 192.168.5.2, 00:00:23, Serial2/0
R 192.X.4.0 [120/1] via 192.168.5.2, 00:00:23, Serial2/0
C 192.X.0.0 is directly connected, FastEthernet0/0
C 192.X.5.0 is directly connected, Serial2/0
```

The R means that the networks were added dynamically using the RIP routing protocol. The [120/1] is the administrative distance of the route (120) along with the number of hops to that remote network (1).

### 4.4. Configuring EIGRP Routing

The command used to configure EIGRP is the same as the one used to configure RIP routing with one important difference: you use an autonomous system (AS) number. All routers within an autonomous system must use the same AS number, or they will not communicate with routing information. Here is how to turn on EIGRP routing:

```
Router(config)#router EIGRP <AS>
Router(config-router)#network <ID-OF-CONNECTED-NETWORKS>
```

Router 0:

```
Router(config)#router eigrp 10
Router(config-router)#network 192.X.0.0
Router(config-router)#network 192.X.5.0
```

Router 1:

```
Router(config)#router eigrp 10

Router(config-router)#network 192.X.1.0
Router(config-router)#network 192.X.4.0
```

```
Router(config-router)#network 192.X.5.0
```

#### Router 2:

```
Router(config)#router igrp 10  
Router(config-router)#network 192.X.2.0  
Router(config-router)#network 192.X.3.0  
Router(config-router)#network 192.X.4.0
```

### 4.5. Verifying the EIGRP Routing Tables

It is important to verify your configurations once you have completed them, or at least once, you think you have completed them. The following list includes the commands you can use to verify the routed and routing protocols configured on your Cisco routers:

- show ip route
- show protocols
- show ip protocols
- debug ip rip
- debug eigrp packets
- debug ip eigrp notifications
- debug ip eigrp neighbor

## 5. Todo

This part will be given to you by the instructor.





**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 4. Dynamic Routing 2 (Link State Routing Protocols)**

### **Open Shortest Path First (OSPF)**

#### **1. Objectives**

- ❖ Learn how to configure and verify IP routing with Cisco routers.
- ❖ Dynamic routing OSPF

#### **2. Lab Requirements**

- ❖ Four Cisco routers.
- ❖ Six PCs.
- ❖ Three Cisco Switches.
- ❖ Several CAT5 straight-wired cables.
- ❖ Four Serial cable (Male and female).

#### **3. Introduction**

Open shortest path first (OSPF) is an Interior Gateway Protocol. Designed expressly for IP networks, OSPF supports variable length subnet masks (VLSM), making it a classless routing protocol. OSPF also allows packet authentication and uses IP multicast when sending/receiving packets.

OSPF routing protocol has two primary characteristics. The first is that the protocol is open. The second is that it is based on SPF algorithm (Dijkstra algorithm) OSPF is the routing protocol of choice when:

- There are routers from vendors other than Cisco in the network.
- The network requires segmentation into areas or zones.

OSPF is a link-state routing protocol. That calls for sending of link-state advertisements (LSAs) to all other routers within the same area. As OSPF routers accumulate link-state information, they use the SPF algorithm to calculate the shortest path to each node.

As a link-state routing protocol, OSPF contrasts with RIP, which is distance-vector routing protocol. Routers running RIP send all of their routing tables in routing-update messages to their neighbors.

OSPF uses bandwidth as metric (cost). It uses a reference bandwidth of **100 Mbps** for cost calculation. The formula to calculate the cost is reference bandwidth divided by interface bandwidth. Thus, a 100Mbps link has a metric of 1; a 10Mbps link has a metric of 10; a 1Gbps (or faster) link also has a cost of 1 because the cost cannot be lower than 1. The cost for each link in the path is added together to form a metric for the route.

### 3.1. Route Summarization

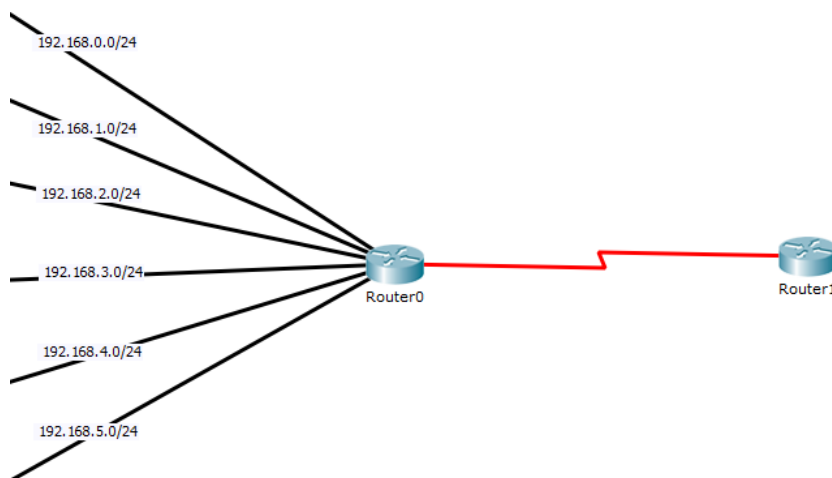


Figure 4-1 How to use Router summarization

Route summarization is the process of replacing a series of routes with a summary route and a mask. This lessens the size of routing update packet itself and makes the routing table smaller, yet still allow for complete IP connectivity when done correctly. In, the 6 more specific routes in router 0 as shown in Figure 4-1 (i.e. 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0 ..... and 192.168.5.0/24) can be replaced by two summary routes which are 192.168.0.0/22 and 192.168.4.0/23.

Not that we cannot replace the 6 networks using 21 subnet mask and id 192.168.0.0/21 with one subnet because this network include smaller subnets that are not connected to router 1 as 192.168.6.0/24 and 192.168.7.0/24.

In OSPF, to summarize routes from one area to another, you can use this command in OSPF routing process mode:

```
Router(config-router)#area AREA-ID range <SUMMARY-ADDRESS> <SUBNET-MASK>
```

### 3.2. Routing Hierarchy

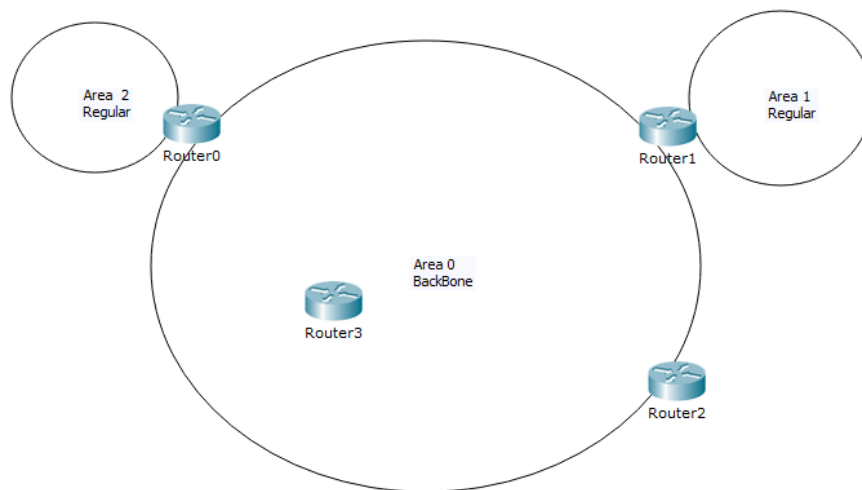
Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the autonomous system (AS), which is a collection of networks under a common administration that share a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to another ASs.

An AS can be divided into a number of areas, which are groups of contiguous networks and attached hosts. Routers with multiple interfaces can participate in multiple areas. These routers, which are called Area Border Routers (ABRs), maintain separate topological databases for each area.

An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned. Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; inter-area routing occurs when they are in different areas.

An OSPF backbone which is called **area 0** is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers shows an area design diagram.

The backbone area forms the central hub of an OSPF network. All other areas are connected to it, and inter-area routing happens via routers connected to the backbone area and to their own non-backbone areas. The backbone area distributes all routing information between the non-backbone areas. The backbone must be adjacent to all other areas, but does not need to be physically contiguous. Connectivity can be established and maintained through virtual links. All OSPF areas must connect to the backbone area. This connection, however, can be through a virtual link.



*Figure 4-2 Area Design*

You can notice that all areas connect back to AREA 0. This implies that both R0 and R1 are Area Border Routers (ABRs).

### 3.3. Understanding OSPF Neighbor Relationships

Hello messages are sent on chosen interfaces once every 10 seconds on broadcast/point to point networks. These messages contain all sort of information:

*Table 4-1 Information in OSPF Messages*

Router ID	<b>Hello and dead timers</b>	<b>Network mask</b>
<b>Area id</b>	Neighbors	Router priority
DR/BDR IP addresses		Authentication password

The parameters in bold must match between the routers to form the OSPF neighbor relationship.

### 3.4. Enabling OSPF

As with other routing protocols, enabling OSPF requires that you create an OSPF routing process, specify the range of IP addresses to be associated with the routing process, and assign area IDs to be associated with that range of IP addresses. Use the following commands, starting in global configuration mode:

#### 3.4.1. Configuring OSPF on the Router

```
Router(config)# router ospf <PROCESS-ID>
```

This command starts the OSPF routing process with your process number. The process number is an arbitrary number. It is recommended that the number match on all routers but it is not required.

#### 3.4.2. Adding networks to the OSPF protocol

```
Router(config-router)# network <ID-ADDRESS> <WILDCARD-MASK> area <AREA-ID>
```

This command defines an interface on which OSPF runs and defines the area ID for that interface. Once OSPF is configured, you can check the status using these commands

```
show ip route,  
show ip ospf neighbor  
show ip protocols
```

### **3.5. Router ID**

The OSPF router id identifies the router to OSPF neighbors. It is in IP format. The default value is highest physical interface at startup. However, loopback interfaces beat physical interfaces.

There is a command to hardcode the router id value that beats all:

```
Router(config-router)#router-id <A.B.C.D>
```

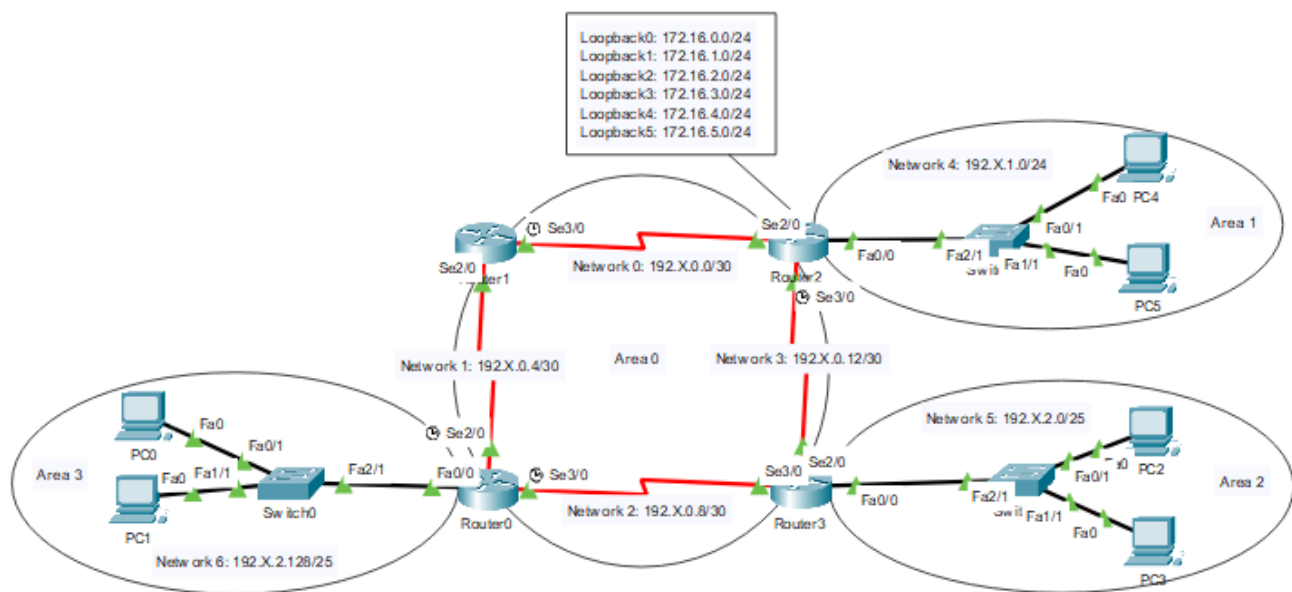
## 4. Procedure

In this lab, we will connect four routers and several PCs on different networks and Loopback networks. This will require configuring routing protocols between the routers. We will configure dynamic routing (OSPF) which will be used as a routing protocol.

### 4.1. Building the Topology

Build the topology shown in Figure 4-3.

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type



*Figure 4-3 Topology*



Table 4-2 Networks IPS

Area/ Summarization	Network	Device	Interface	IP	Subnet Mask	Wildcard Mask
Area 0	Network 0 192.X.0.0/30	Router 2	Se2/0	192.X.0.1	255.255.255.252	0.0.0.3
		Router 1	Se3/0	192.X.0.2	255.255.255.252	0.0.0.3
	Network 1 192.X.0.4/30	Router 0	Se2/0	192.X.0.5	255.255.255.252	0.0.0.3
		Router 1	Se2/0	192.X.0.6	255.255.255.252	0.0.0.3
	Network 2 192.X.0.8/30	Router 0	Se3/0	192.X.0.9	255.255.255.252	0.0.0.3
		Router 3	Se3/0	192.X.0.10	255.255.255.252	0.0.0.3
	Network 3 192.X.0.12/30	Router 2	Se3/0	192.X.0.13	255.255.255.252	0.0.0.3
		Router 3	Se2/0	192.X.0.14	255.255.255.252	0.0.0.3
Area 1	Network 4 192.X.1.0/24	Router 2	Fa0/0	192.X.1.1	255.255.255.0	0.0.0.255
		PC4	Fa0	192.X.1.2	255.255.255.0	0.0.0.255
		PC5	Fa0	192.X.1.3	255.255.255.0	0.0.0.255
Area 2	Network 5 192.X.2.0/25	Router 3	Fa0/0	192.X.2.1	255.255.255.128	0.0.0.127
		PC2	Fa0	192.X.2.2	255.255.255.128	0.0.0.127
		PC3	Fa0	192.X.2.3	255.255.255.128	0.0.0.127
Area 3	Network 6 192.X.2.128/25	Router 0	Fa0/0	192.X.2.129	255.255.255.128	0.0.0.127
		PC0	Fa0	192.X.2.130	255.255.255.128	0.0.0.127
		PC1	Fa0	192.X.2.131	255.255.255.128	0.0.0.127
Summarization 172.16.0.0/22	172.16.0.0/24	Router 2	Loopback0	172.16.0.1	255.255.255.0	0.0.0.255
	172.16.1.0/24	Router 2	Loopback1	172.16.1.1	255.255.255.0	0.0.0.255
	172.16.2.0/24	Router 2	Loopback2	172.16.2.1	255.255.255.0	0.0.0.255
	172.16.3.0/24	Router 2	Loopback3	172.16.3.1	255.255.255.0	0.0.0.255
Summarization 172.16.4.0/23	172.16.4.0/24	Router 2	Loopback4	172.16.4.1	255.255.255.0	0.0.0.255
	172.16.5.0/24	Router 2	Loopback5	172.16.5.1	255.255.255.0	0.0.0.255

Use the following IPs shown in Table 4-2 for the configuration. In order to configure the IPs for the PCs and Routers, see EXP. No. 2 sections 5.1.2 & 5.1.3.

To configure the loopback IPs, you first need to create a loopback interface and then add the IP address to the loopback. The configuration below is adding the IP address 172.16.0.1 to loopback 0:

```
Router(config)#interface loopback 0
%LINK-5-CHANGED: Interface Loopback0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed
state to up
Router(config-if)#ip address 172.16.0.1 255.255.255.0
```

## 4.2. Configuring OSPF Routing

The command used to configure OSPF is different from the one used to configure RIP routing; here you use a Process ID number which can take numbers between <1 - 65535> for the OSPF routing protocol. To enable OSPF routing on a router the following command will be executed:

```
Router(config)#router ospf <PROCESS-ID>
```

That tells the OSPF routing protocol which networks to advertise. That is it. Let us configure our internetworks with OSPF routing and practice that. The command used for adding a network is:

```
Router(config-router)#network <NETWORK-ID> <OSPF-WILDCARD-BITS> area <AREA-ID>
```

For example, to set the OSPF on Router 3, we use the following commands:

```
Router(config)#router ospf 1
Router(config-router)#network 192.X.0.8 0.0.0.3 area 0
Router(config-router)#network 192.X.0.12 0.0.0.3 area 0
Router(config-router)#network 192.X.2.0 0.0.0.127 area 2
```

- **Now you must complete the OSPF configuration for all routers.**

### 4.3. Changing the Cost

You can see the routing tables for each router by executing the command (`sh ip route`), we also can see that to go to a Network is through interface example (serial2/0) and the cost using this path.

You also can change the cost from an interface by changing the bandwidth of that interface so to reduce the cost for going from PC0 to PC4 through Routers R0→R1→R2 we can change the cost from R0→R1 and from R1→R2 by changing the bandwidth for interface Se2/0 for R0 and Se3/0 for R1. The routing table for router0 after changing the bandwidth will be affected. To change the cost on an interface to 5 we can set the bandwidth to be 20000Kbits using the formula.

Therefore, to set the cost to 5 the bandwidth should be 20Mbits which is equivalent to 20000Kbits. To set the bandwidth on an interface first we should access the targeted interface and change the bandwidth using the following command:

```
Router(config-if)#bandwidth <BANDWIDTH-IN-KILOBITS>
```

For example, to change the cost to 5 for interface Se2/0 for router0 we use the following commands:

```
Router(config)#interface se2/0  
Router(config-if)#bandwidth 20000
```

The end-to-end cost is the summation of the cost through all interfaces.

- **Note that the serial interface to the switch has a cost of 1.**

#### 4.4. Summarization

To add the loopback networks to the OSPF on router 2 we can add them one by one for example:

```
Router(config)#router ospf 1
Router(config-router)#network 172.16.0.0 0.0.0.255 area 1
Router(config-router)#network 172.16.1.0 0.0.0.255 area 1
Router(config-router)#network 172.16.2.0 0.0.0.255 area 1
Router(config-router)#network 172.16.3.0 0.0.0.255 area 1
Router(config-router)#network 172.16.4.0 0.0.0.255 area 1
Router(config-router)#network 172.16.5.0 0.0.0.255 area 1
```

But as we can realize that we can combine the first 4 networks with one large network that include them all which is 172.16.0.0/22, so we can replace the first four networks with this ID resulting in:

```
Router(config)#router ospf 1
Router(config-router)#network 172.16.0.0 0.0.3.255 area 1
Router(config-router)#network 172.16.4.0 0.0.0.255 area 1
Router(config-router)#network 172.16.5.0 0.0.0.255 area 1
```

Also, the last two networks can be combined to 172.16.4.0/23 and resulting in:

```
Router(config)#router ospf 1
Router(config-router)#network 172.16.0.0 0.0.3.255 area 1
Router(config-router)#network 172.16.4.0 0.0.1.255 area 1
```

So this is why very useful to reduce the CPU execution and not dealing with 6 networks but with only two.

#### 4.5. Important Questions

- Why do we need for loopback interfaces?
- What is the router-id for OSPF? And why do we need it?
- Hardcode the router-id for R1, R2, and R3 as 1.1.1.1, 2.2.2.2, and 3.3.3.3 respectively. And Verify that.

## 5. Problem

Given the following the topology shown in Figure 4-4.

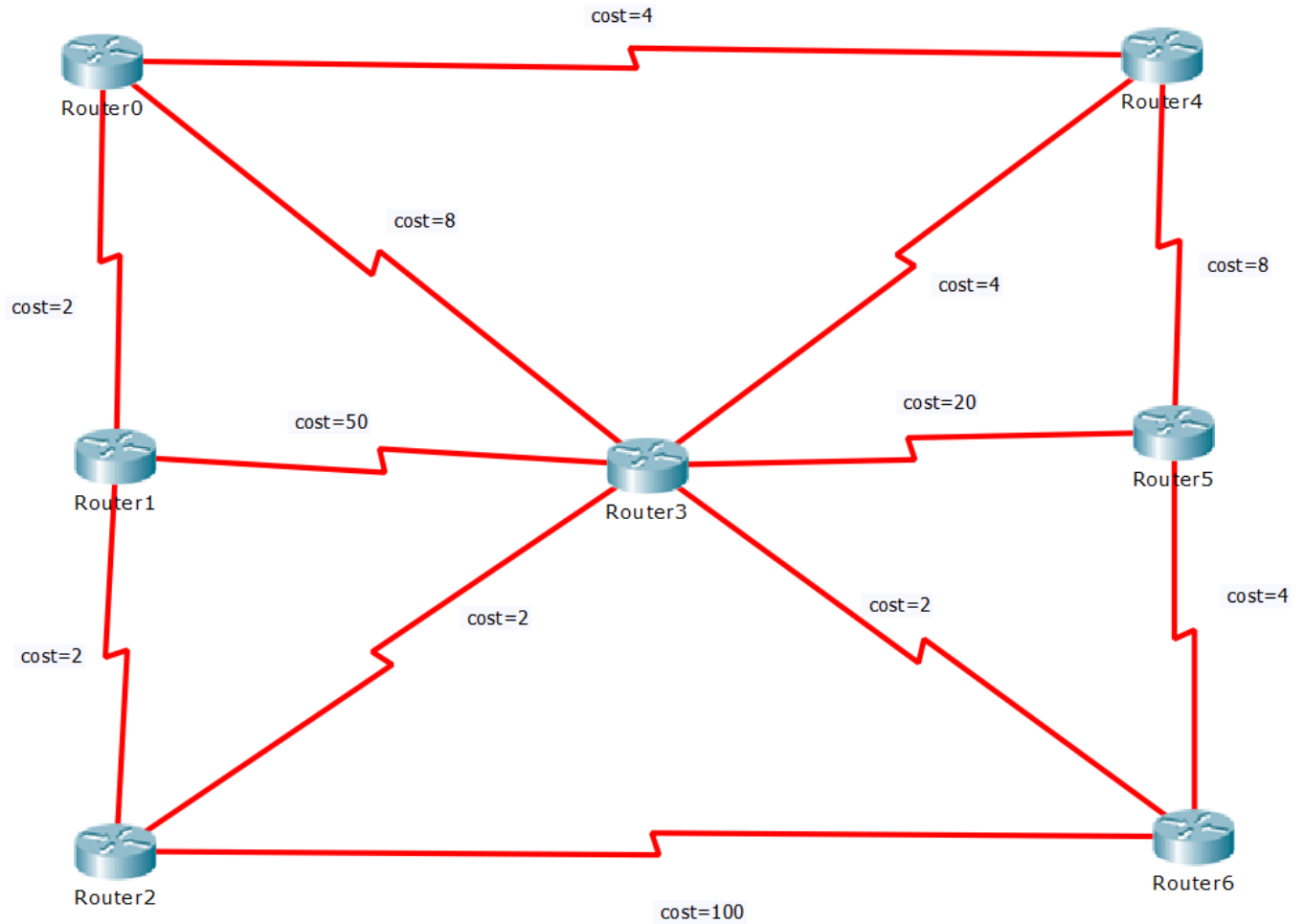


Figure 4-4 Problem Topology

### ➤ Part One:

1. Find the shortest path from Router 0 to Router 6 using Dijkstra's algorithm. Show your steps.
2. What is the cost of the shortest path from Router 0 to Router 6?

➤ Part Two:

Build and configure the above topology using Packet Tracer software based on the following requirements.

- Requirements:

To help guide this initial configuration, you've assembled a list of requirements:

1. For addressing the above network use the class C address 192.A.B.0 and use it to create networks (subnets) of 2 hosts each. A, and B represent the last four digits of your university ID.  
For example: if your university ID 1140302  
then (A = 03 = 3) and (B = 02 = 2)
2. Enable OSPF routing. Assume all routers are in area 0 (backbone)
3. Configure Router 6 with a loopback IP address 7.7.7.7/24. Advertise this network into OSPF process.
4. Don't forget to configure bandwidth values between links. These values should reflect the costs that are shown in the network diagram.
5. If a packet is sent from Router 0 to Router 7 (i.e. loopback 7.7.7.7). What routers it passes through until it reaches its destination? Use the traceroute command to test that.
6. Run the show IP route command on Router 0. From the output result. What is the cost (metric) to get from Router 0 to Router 6? Explain that.
7. What is the router-id for Router 0, and Router 6? Verify your answers

## 6. Todo

This part will be given to you by the instructor



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 5. Dynamic Routing 3 (Path Vector) BGP**

### **1. Objectives**

- ❖ Learn how to configure and verify IP routing with Cisco routers.
- ❖ Introducing to exterior gateway protocol and interior gateway protocols
- ❖ Introducing to Autonomous systems
- ❖ Dynamic routing BGP

### **2. Lab Requirements**

- ❖ Four Cisco routers.
- ❖ Six PCs.
- ❖ One Cisco switch.
- ❖ Several CAT5 straight-wired cables.
- ❖ Two Serial cable. (male and female).

## 3. Introduction

### 3.1. Border Gateway Protocol (BGP)

BGP is a standardized *exterior* gateway protocol (EGP), as opposed to RIP, OSPF, and EIGRP which are *interior* gateway protocols (IGP's). BGP Version 4 (BGPv4) is the current standard deployment.

BGP is considered a “Path Vector” routing protocol. BGP was not built to route within an Autonomous System (AS), but rather to route between AS's. BGP maintains a separate routing table based on shortest AS Path and various other attributes, as opposed to IGP metrics like distance or cost.

BGP is the routing protocol of choice on the Internet. Essentially, the Internet is a collection of interconnected Autonomous Systems.

BGP Autonomous Systems are assigned an Autonomous System Number (ASN), which is a 16-bit number ranging from 1 – 65535. A specific subset of this range, 64512 – 65535, has been reserved for private (or internal) use.

### 3.2. Using BGP

Contrary to popular opinion, BGP is not a necessity when multiple connections to the Internet are required. Fault tolerance or redundancy of outbound traffic can easily be handled by an IGP, such as OSPF or EIGRP.

BGP is also completely unnecessary if there is only one connection to an external AS (such as the Internet). There are over 100,000 routes on the Internet, and interior routers should not be needlessly burdened.

BGP should be used under the following circumstances:

- Multiple connections exist to external AS's (such as the Internet) via different providers.
- Multiple connections exist to external AS's through the same provider but connect via a separate CO or routing policy.



- The existing routing equipment can handle the additional demands.

BGP's true benefit is in controlling how traffic enters the local AS, rather than how traffic exits it.

To configure a BGP connection use the following command:

```
Router (config)# router bgp <AS-NUMBER>
```

Where the AS-NUMBER is the autonomous system number where the router is.

### 3.3. BGP Peers/Neighbors

For BGP to function, BGP routers (called speakers) must form neighbor relationships (called peers).

There are two types of BGP neighbor relationships:

- **iBGP** Peers – BGP neighbors within the same autonomous system.
- **eBGP** Peers – BGP neighbors connecting separate autonomous systems.

▪ **Note: Do not confuse an IGP, such as OSPF and RIO, with iBGP**

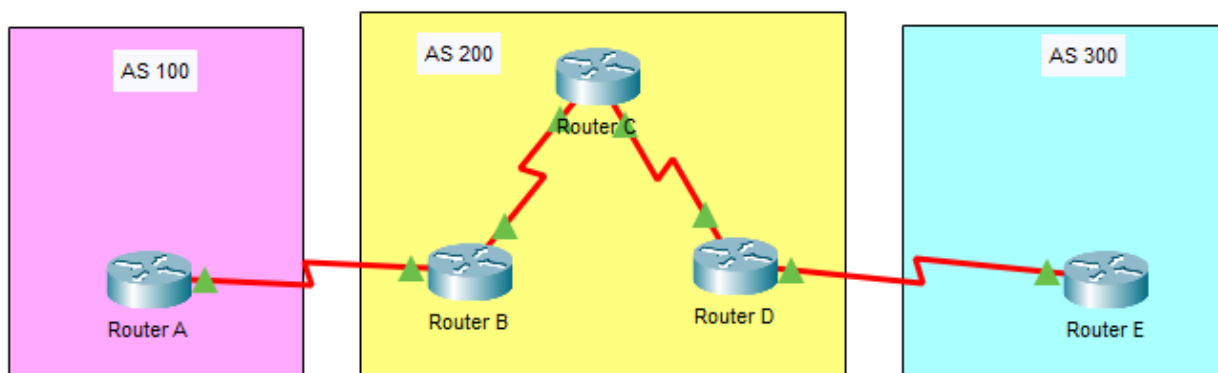


Figure 5-1 AS Connecting with BGP Links

In Figure 5-1, Router1, Router2, Router3 and Router4 are in AS 200 would form an iBGP peer relationship. Router0 is in AS 100 and Router5 in AS 200 would form an eBGP peering.

Once BGP peers form a neighbor relationship, they share their full routing table. Afterwards, only changes to the routing table are forwarded to peers. A Cisco router running BGP can belong to only one AS. The IOS will only allow one BGP process to run on a router.

The Administrative Distance for routes learned outside the Autonomous System (eBGP routes) is 20, while the AD for iBGP and locally originated routes is 200.

To configure a neighbor relationship with a router in separate AS (eBGP Peer) using the command:

```
Router(config-router)# neighbor <IP-ADDRESS-NEXT-INTERFACE> remote-
as <AS-OF-REMOTE-NEIGHBOR>
```

Where IP-ADDRESS-NEXT-INTERFACE is the address of the interface on other peer. And the AS\_OF\_REMOTE\_NEIGHBOR is the autonomous system number of the next AS.

### 3.4. BGP Peers Messages

BGP forms its peer relationships through a series of messages listed below

- **OPEN message**, it is sent between peers to initiate the session. The OPEN message contains several parameters:
  - BGP Version – must be the same between BGP peers
  - Local AS Number
  - BGP Router ID
- **KEEPALIVE messages**, these are sent periodically (every 60 seconds by default) to ensure that the remote peer is still available. If a router does not receive a **KEEPALIVE** from a peer for a Hold-time period (by default, 180 seconds), the router declares that peer dead. To globally adjust the **KEEPALIVE** and Hold-time timers for all neighbors:

```
Router(config-router)# timers bgp <KEEP-ALIVE> <HOLD-TIME>
```

- **UPDATE messages**, these are used to exchange routes between peers.

- **NOTIFICATION messages** are sent when there is a fatal error condition. If a NOTIFICATION message is sent, the BGP peer session is torn down and reset.

### 3.5. BGP Finite-State Machine (FSM):

As a BGP peer session is forming, it will pass through several states.

- Idle:** The initial BGP state.
- Connect:** BGP waits for a TCP connection with the remote peer. If successful, an OPEN message is sent. If unsuccessful, the session is placed in an Active state.
- Active:** BGP attempts to initiate a TCP connection with the remote peer. If successful, an OPEN message is sent. If unsuccessful, BGP will wait for a ConnectRetry timer to expire, and place the session back in a Connect State.
- OpenSent:** BGP has both established the TCP connection and sent an OPEN Message and is awaiting a reply OPEN Message. Once it receives a reply OPEN Message, the BGP peer will send a KEEPALIVE message.
- OpenConfirm:** BGP listens for a reply KEEPALIVE message.
- Established:** The BGP peer session is fully established. UPDATE messages containing routing information will now be sent.

If a peer session is stuck in an Active state, potential problems can include: no IP connectivity, an incorrect neighbor statement, or an access-list filtering TCP port 179.

## 4. Procedure

In this lab, we will connect four routers and several PCs on different networks. This will require configuring routing protocols between the routers. We will configure dynamic routing (OSPF) which will be used as a routing protocol inside the same Autonomous System and BGP between the different Autonomous Systems.

### 4.1. Building the topology

Build the topology shown in Figure 5-2.

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

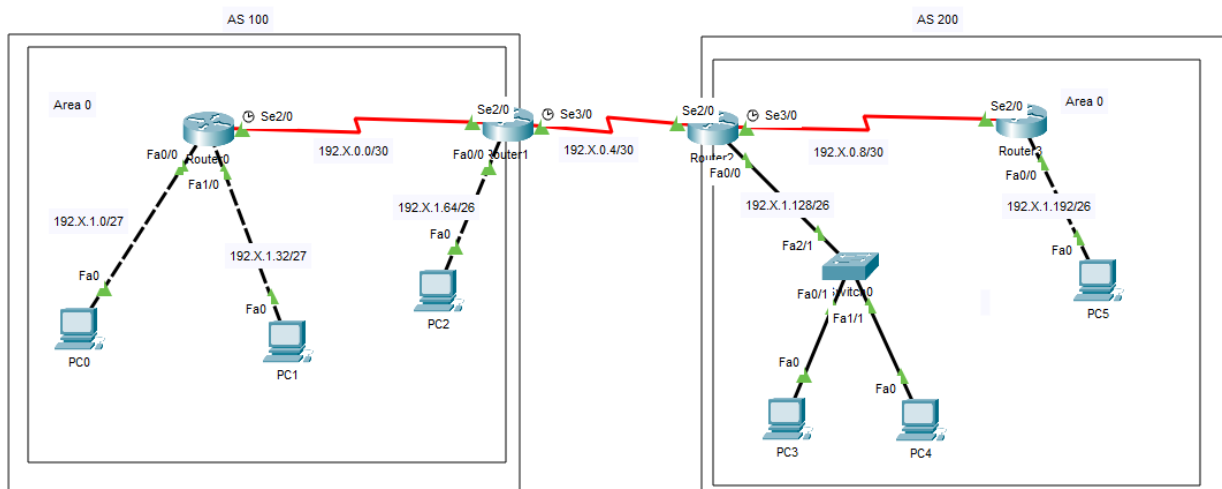


Figure 5-2 Topology

Table 5-1 Networks IPS

Area/AS & BGP Links	Network	Device	Interface	IP	Subnet Mask	Wildcard Mask
Area 0 / AS 100	192.X.0.0/30	Router 0	Se2/0	192.168.0.1	255.255.255.252	0.0.0.3
		Router 1	Se2/0	192.168.0.2	255.255.255.252	0.0.0.3
	192. X.1.0/27	Router 0	Fa0/0	192.168.1.1	255.255.255.224	0.0.0.31
		PC0	Fa0	192.168.1.2	255.255.255.224	0.0.0.31
	192. X.1.32/27	Router 0	Fa1/0	192.168.1.33	255.255.255.224	0.0.0.31
		PC1	Fa0	192.168.1.34	255.255.255.224	0.0.0.31
	192. X.1.64/26	Router 1	Fa0/0	192.168.1.65	255.255.255.192	0.0.0.63
		PC2	Fa0	192.168.1.66	255.255.255.192	0.0.0.63
Area 0 / AS 200	192.X.0.8/30	Router 2	Se3/0	192.X.0.9	255.255.255.252	0.0.0.3
		Router 3	Se2/0	192.X.0.10	255.255.255.252	0.0.0.3
	192.X.1.128/26	Router 2	Fa0/0	192.X.1.129	255.255.255.192	0.0.0.63
		PC 3	Fa0	192.X.1.130	255.255.255.192	0.0.0.63
		PC 4	Fa0	192.X.1.131	255.255.255.192	0.0.0.63
	192.X.1.192/26	Router 3	Fa0/0	192.X.1.193	255.255.255.192	0.0.0.63
		PC 5	Fa0	192.X.1.194	255.255.255.192	0.0.0.63
BGP Links	192.X.0.4/30	Router 1	Se3/0	192.X.0.5	255.255.255.252	0.0.0.3
		Router 2	Se2/0	192.X.0.6	255.255.255.252	0.0.0.3

Use the following IPs shown in Table 5-1 for the configuration. In order to configure the IPs for the PCs and Routers, see EXP. No. 2 sections 5.1.2 & 5.1.3.

## 4.2. Configuring OSPF Routing

We will configure OSPF routing protocol for the two antonyms systems (100 and 200) separately, you will not include the BGP link in the OSPF configuration for both routers 1 and 2. In order to configure OSPF see EXP. No. 4 section 4.2.

## 4.3. Configuring BGP Routing

BGP configuration have to be done just on Router 1 and Router 2. The first step in configuring BGP is to enable the BGP process, and specify the router's Autonomous System (AS):

```
Router (config)# router bgp <AS-NUMBER>
```

Where the AS-NUMBER is the autonomous system number where the router is. For example, to enable BGP on router 1 do the following command:

```
Router (config)# router bgp 100
```

Here BGP is now enabled on router 1. The next step is to configure a neighbor relationship with a router in separate AS (eBGP Peer) using the command:

```
Router(config-router)# neighbor <IP-ADDRESS-NEXT-INTERFACE> remote-  
as <AS-OF-REMOTE-NEIGHBOR>
```

Where IP-ADDRESS-NEXT-INTERFACE is the address of the interface on other peer. And the AS\_OF\_REMOTE\_NEIGHBOR is the autonomous system number of the next AS. For example, to configure BGP on router 1 do the following command:

```
Router(config-router)# neighbor 192.X.0.6 remote-as 200
```

▪ **Now repeat step 4.3 for router 2 with the correct values.**

#### 4.4. Define the BGP Over the OSPF

To allow the OSPF to communicate with the BGP a redistribute command is used to define the BGP protocol over the OSPF protocol:

```
Router(config)# router ospf <PROCESS-ID>
Router(config-router)# redistribute bgp <AS-NUMBER> subnets
```

Where the PROCESS-ID is the OSPF ID you configured in section 4.2 and the AS-NUMBER is the autonomous number for the BGP of configured on the same router. For example, to redistribute the BGP over the OSPF on router 1 use the following commands assuming you have chosen the process id for the OSPF to be 1.

```
Router(config)# router ospf 1
Router(config-router)# redistribute bgp 100 subnets
```

- **Now repeat section 4.4 for router 1 with the correct values.**

#### 4.5. Define the OSPF Over the BGP

To allow the BGP to communicate with the OSPF a redistribute command is used to define the OSPF protocol over the BGP protocol:

```
Router(config)# router bgp <AS-NUMBER>
Router(config-router)# redistribute ospf <PROCESS-ID>
```

Where the PROCESS-ID is the OSPF ID you configured in section 4.1 and the AS-NUMBER is the autonomous number for the BGP of configured on the same router. For example, to redistribute the OSPF over the BGP on router 1 use the following commands assuming you have chosen the process id for the OSPF to be 1.

```
Router(config)# router bgp 100
Router(config-router)# redistribute ospf 1
```

- **Now repeat section 4.5 for router 1 with the correct values.**

#### ***4.6. Configuring BGP Timers***

To set the keepalive and hold time for router 1 to 30 and 90 respectively use the following commands:

```
Router(config)# router bgp 100
Router(config-router)# timers bgp 30 90
```

These by default will change the timers on router, if the configured Hold-time timers between two peers are different, the peer session will still be established, and the smallest timer value will be used.

#### ***4.7. Viewing BGP Neighbors***

To view the status of all BGP neighbors you can use the following commands:

```
Router# show ip bgp
Router# show ip bgp summary
Router# show ip route
```

#### ***4.8. Viewing the BGP Routing Table***

Recall that BGP maintains its own separate routing table. This table contains a list of routes that can be advertised to BGP peers.

To view the BGP routing table on Router 1 run the following commands on the first router:

```
Router# show ip bgp
```

### **5. Todo**

This part will be given to you by the instructor.





**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 6. Access Lists**

### **1. Objectives**

- ❖ Learn how to configure and verify Access Lists with Cisco routers.
- ❖ Introducing to Standard ACL and Extended ACL

#### **PRE-LAB:**

Each student should prepare the topology in order to start with the new configurations of this experiment. (IPs and OSPF configurations).

### **2. Lab Requirements**

- ❖ Three Cisco routers.
- ❖ Six PCs.
- ❖ Three Cisco switches.
- ❖ Several CAT5 straight-wired cables.
- ❖ Two Serial cable. (male and female).

### 3. Introduction

The technical name for an access list is Access Control List (ACL). The individual entries in an access control list are called access control entries. The term access control list isn't often used in practice; you'll typically hear these lists referred to simply as access lists or ACLs.

Access lists filter network traffic by controlling whether routed packets are forwarded or blocked at the router's interfaces. The router examines each packet to determine whether to forward or drop the packet, based on the criteria specified within the access lists.

Access list criteria could be the source address of the traffic, the destination address of the traffic, the upper-layer protocol, or other information.

There are many reasons to configure access list, for example, you can use access lists to restrict contents of routing updates, or to provide traffic flow control. But one of the most important reasons to configure access lists is to provide security for your network; this is the reason focused on in this experiment.

To create an access list, you specify the protocol to filter, you assign a unique name or number to the access list, and you define packet filtering criteria. A single access list can have multiple filtering criteria statements.

#### 3.1. *Named and numbered ACL*

Access lists on many Cisco devices can be either named or numbered. Named access lists are referenced with a name such as BZU. Numbered access lists are the older method, where each ACL is defined by a number such as 10, 101 105...

#### 3.2. *Wildcard Masks*

Masks are used with IP addresses in ACLs to specify what should be permitted and denied. Masks in order to configure IP addresses on interfaces start with 255 and have the large values on the left side, for example, IP address 209.165.202.129 with a 255.255.255.224 mask. Masks for IP ACLs are the reverse, for example, mask 0.0.0.255. This is sometimes called an inverse mask or a wildcard mask. When the value of the mask is broken down into binary (0s and 1s), the results

determine which address bits are to be considered in processing the traffic. A 0 indicates that the address bits must be considered (exact match); a 1 in the mask is a "don't care".

### ***3.3. Standard ACL and Extended ACL***

#### ***3.3.1. Standard ACL***

Standard ACL use only the source IP address in an IP packet to filter the network. This basically permits or denies an entire suite of protocols. You can create a standard access list by using the number **1** to **99**.

#### ***3.3.2. Extended ACL***

On the other hand, extended ACL check for both source and destination IP address, protocol field in the Network layer header, and port number at the Transport layer header. You can use **100** to **199** for specifying your extended ACL.

### ***3.4. The Implied "Deny All Traffic" Criteria Statement***

At the end of every access list is an implied "deny all traffic" criteria statement. Therefore, if a packet does not match any of your criteria statements, the packet will be blocked.

### ***3.5. Configuring ACL***

#### ***3.5.1. Standard ACL configuration***

```
Router(config)# access-list <ACCESS-LIST-NUMBER> <permit|deny>  
<host|source sourceWildcardMask|any>
```

For example, in your network you want that no computer or devices from 192.X.0.0/24 network can send traffic to your network. To implement this rule, you need to write an ACL that will tell your router to discard all the traffic from 192.X.0.0/24. Now, let's see how to implement this into a router using standard ACL

```
Router(config)# access-list 10 deny 192.X.0.0 0.0.0.255
```

After the ACL is defined whether it is standard or an extended one, it must be applied to the interface (inbound or outbound).

```
Router(config)# interface <INTERFACE-NUMBER>
Router(config-if)# ip access-group <ACCESS-LIST-NUMBER> <in|out>
```

For example, if we want to add the access list on interface FastEthernet 0/0 on the inbound of the interface then we execute the following commands:

```
Router(config)# interface fa0/0
Router(config-if)# ip access-group 10 in
```

### 3.5.2. Extended Access List Configuration

```
Router(config)# access-list <ACCESS-LIST-NUMBER> <permit|deny>
<TRANSPORT-LAYER-PROTOCOL> <host|source sourcewildcardmask|any>
<host|destination destinationWildcardMask|any> eq <PORT-NUMBER>
```

Extended ACLs allow you to specify the source and destination IP address. Moreover, you can specify which protocols and service ports (i.e. www, telnet, and ftp) you want to deny in your router. For example you want to deny HTTP connection originating only from a host with IP 192.X.1.2 to your web server with IP 172.16.100.100, and to do that you have to write the following extended access control list on your router and then apply it to an interface that you expect to receive incoming HTTP request from outsiders.

```
Router(config)# access-list 120 deny tcp host 192.X.1.2 host
172.16.100.100 eq 80
```

```
Router(config)# access-list 120 permit ip any any
```

If you did not add the “eq 80” in the above access list, then your router would deny all the tcp packets irrespective of its destination port, which means if a person tries to FTP to your host he would be denied.

As standard access list, after the ACL is defined whether it is standard or an extended one, it must be applied to the interface (inbound or outbound). For example, if we want to add the access list on interface FastEthernet 0/1 on the outbound of the interface then we execute the following commands:

```
Router(config)# interface fa0/1  
Router(config-if)# ip access-group 10 out
```

You can Issue the `show access-list` command to view the ACL entries.

## 4. Procedure

In this lab we will connect two routers and several PCs on different networks. This will require configuring routing protocols between the routers. We will configure dynamic routing (OSPF) which will be used as a routing protocol and try putting some access lists to deny or allow connections between different end users and networks.

### 4.1. Building the topology

Build the topology shown in Figure 6-1.

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

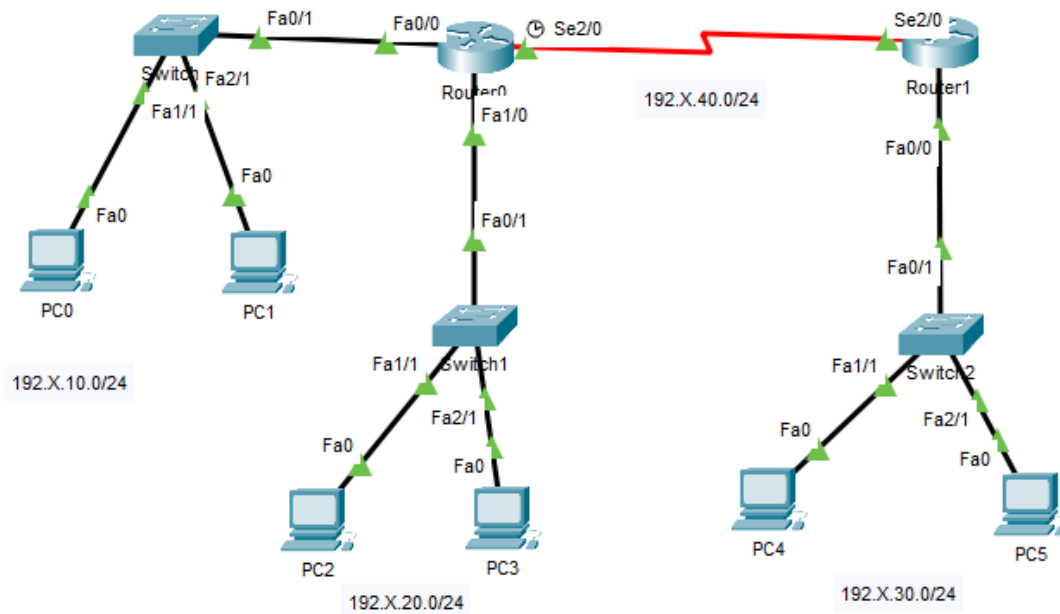


Figure 6-1 Topology

Table 6-1 Networks IPS

Area	Network	Device	Interface	IP	Subnet Mask	Wildcard Mask
Area 0	192.X.40.0/24	Router 0	Se2/0	192.X.40.1	255.255.255.0	0.0.0.255
		Router 1	Se2/0	192.X.40.2	255.255.255.0	0.0.0.255
	192.X.10.0/24	Router 0	Fa0/0	192.X.10.1	255.255.255.0	0.0.0.255
		PC0	Fa0	192.X.10.2	255.255.255.0	0.0.0.255
		PC1	Fa0	192.X.10.3	255.255.255.0	0.0.0.255
	192.X.20.0/24	Router 0	Fa1/0	192.X.20.1	255.255.255.0	0.0.0.255
		PC2	Fa0	192.X.20.2	255.255.255.0	0.0.0.255
		PC3	Fa0	192.X.20.3	255.255.255.0	0.0.0.255
	192.X.30. 0/24	Router 1	Fa0/0	192.X.30. 1	255.255.255.0	0.0.0.255
		PC4	Fa0	192.X.30. 2	255.255.255.0	0.0.0.255
		PC5	Fa0	192.X.30. 3	255.255.255.0	0.0.0.255

Use the following IPs shown in Table 6-1 for the configuration. In order to configure the IPs for the PCs and Routers see EXP. No. 2 sections 5.1.2 & 5.1.3.

## 4.2. Configuring OSPF Routing

We will configure OSPF routing protocol for both routers 0 and 1. In order to configure OSPF see EXP. No. 4 section 4.2.

At this point, all PCs must ping with any point included in the network. To make things easier, you will use the topology in each step so make sure to keep a copy of the topology with the basic configuration.

### 4.3. Configuring Standard Access List

In this section, you will use only standard access list, and make sure in each step to **copy the main topology**. Standard ACL takes IDs of 1 to 99.

#### A. Prevent PC0 to access network 192.X.20.0/24.

First, create an access list to deny PC0 we can use one of the following methods:

Method 1:

```
Router0(config)#access-list 10 deny host 192.X.10.2
Router0(config)#access-list 10 permit any
```

Method 2:

```
Router0(config)#access-list 10 deny 192.X.10.2 0.0.0.0
Router0(config)#access-list 10 permit any
```

The command `access-list 10 permit any` used because after assigning an access list, by default there is an implicit deny all traffic the end of every ACL see sec 3.4 in the introduction. Anything that is not explicitly permitted is denied.

Then you must give the ACL to an interface, in our case give it to fa1/0 for the outbound, using the following commands:

```
Router0(config)# interface fa1/0
Router0(config-if)#ip access-group 10 out
```

Now if you try to ping any pc in the network 192.X.20.0 from pc0, it going to fail.



- B. Allow just PC0 to access network 192.X.20.0/24 using the Slandered ACLs and deny any other traffic.
- C. Prevent network 192.X.10.0 from accessing network 192.X.20.0 only (use the wild-card, not 'any' option).
- D. Prevent PC0 from accessing Network 192.X.30.0 all other traffic is allowed. (Think carefully about in which router should you put the rule).

#### 4.4. Configuring Extended Access List

In this section, you will use only extended access list, and make sure in each step to **copy the main topology**. Extended ACL takes IDs of 100 to 199.

- A. Prevent PC0 from accessing PC2. (all other traffic is allowed).

First, create an access list to deny PC0, we can use one of the following methods:

Method 1:

```
Router0(config)#access-list 101 deny ip host 192.X.10.2 host
192.X.20.2
Router0(config)#access-list 101 permit ip any any
```

Method 2:

```
Router0(config)#access-list 101 deny ip 192.X.10.2 0.0.0.0
192.X.20.2 0.0.0.0
Router0(config)#access-list 101 permit ip any any
```

The command `access-list 101 permit ip any any` used because after assigning an access list, by default there is an implicit deny all traffic the end of every ACL see sec 3.4 in the introduction. Anything that is not explicitly permitted is denied.

Then you must give the ACL to an interface, in our case give it to fa0/0 for the inbound, using the following commands:

```
Router0(config)# interface fa0/0
Router0(config-if)#ip access-group 101 in
```

Now if you try to ping pc2 from pc0, it going to fail.

- B.** Allow PC0 to access PC2 all other traffic is denied.
- C.** Add a server to the topology to network 192.X.20.0/24 and activate http service on it, then deny PC0 to make HTTP request to this server. (Hint: use command: access-list 101 deny tcp host 192.X.10.2 host 192.X.20.4 eq 80 Where 80 is the port number for HTTP requests.)
- D.** Prevent PC0 from accessing PC4 all other traffic is allowed. (Think about in which router should you put the rule).
- E.** Enable telnet on Router1 then, deny all the host from making telnet with interface se2/0 of Router1 expect PC0, it can make telnet with any interface. [try to minimize the traffic on the serial line as much as possible]. All other traffic should be allowed.

## 5. Todo

This part will be given to you by the instructor.



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 7. Switching and VLANs 1 - Router on Stick**

### **1. Objectives**

- ❖ Learn how to configure a Cisco IOS Switch using the IOS command-line interface (CLI).
- ❖ Learn how to use switch simulator.
- ❖ Learn how to split Cisco router interface into sub interfaces.
- ❖ Learn how to split Cisco switches into multiple virtual ones and create VLANs.

### **2. Lab Requirements**

- ❖ Two Cisco router
- ❖ Six PCs.
- ❖ Three Cisco switches.
- ❖ several CAT5 straight-wired cable
- ❖ One Serial cable. (male and female).

### 3. Introduction

Switching is a core subject in computer networks. It discusses the different operations that a switch does in parallel to other kinds of concepts such as VLANs, multilayer switching, trunks, layering model of switches, bundle of cables... etc.

This experiment will be an introduction to switching in general and VLANs. The main concept of VLANs which will be discussed here is the router on stick. There are other ways to implement VLANs such as the SVI concept that will be discussed in the next experiment.

#### 3.1. *How does a switch work?*

A switch usually works as soon as it is plugged into a power source. Whenever a device is sending a frame at one of its ports, the switch extracts the MAC address of that device and links it to the port. At the end, a switch builds a table of MAC-Port mappings so it can deliver packets from one device to another.

#### 3.2. *IEEE 802.1Q VLAN*

A VLAN (Virtual Local Area Network) is an abbreviation of the term Virtual LAN. It is mainly a logical grouping of the switch ports into different subnets.

Short for virtual LAN, a network of computers that behave as if they are connected to the same wire even though they may be physically located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible. One of the biggest advantages of VLANs is that when a computer is physically moved to another location, it can stay on the same VLAN without any hardware reconfiguration.

IEEE 802.1Q is a protocol for carrying VLAN traffic on an Ethernet. A VLAN is a type of local area network that does not have its own dedicated physical infrastructure, but instead uses another LAN to carry its traffic. The traffic is encapsulated so that a number of logically separate VLANs can be carried by the same physical LAN.

You should consider using VLANs whenever there is a need for traffic to be segregated at the link layer. For example, on Internet Protocol networks it is considered good practice to use a separate VLAN for each IP subnet. Reasons for doing this include:

- Preventing a machine assigned to one subnet from joining a different one by changing its IP address.
- Avoiding the need for hosts to process broadcast traffic originating from other subnets.

### 3.2.1. Tagging

802.1Q VLAN frames are distinguished from ordinary Ethernet frames by the insertion of a 4-byte VLAN tag into the Ethernet header. It is placed between the source MAC and the EtherType fields see Table 7-1.

*Table 7-1 MAC address distribution*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18...
Destination address						Source address						VLAN tag		EtherType	Payload			
												0x8100	TCI					

The first two bytes of the tag contain the TPID (tag protocol identifier), which is defined to be equal to 0x8100. Since it is located where the EtherType would appear in an ordinary Ethernet frame, tagged frames appear to have an EtherType of 0x8100.

The remaining two bytes contain the TCI (tag control information), of which 12 bits correspond to the VID (VLAN identifier, described below) and 4 bits contain metadata used for quality of service management.

### 3.3. VLAN numbering

Each 802.1Q VLAN is identified by a 12-bit integer called a VID (VLAN Identifier) in the range 1 to 4094 inclusive. The values 0 and 4095 are reserved and should not be used.

The first VLAN, with a VID of 1, is the default VLAN to which ports are presumed to belong if they have not been otherwise configured. It is considered good practice to move traffic off the default VLAN where practicable (see below).

The remaining values have no special status and can be used freely, but be aware that many network devices place a limit on the number of VLANs that can be configured so it will not necessarily be feasible to make use of all 4094 possible VIDs.

### ***3.3.1. Creating a VLAN***

You can create a specific VLAN on a switch using the following command

```
Switch(config)# VLAN <VLAN-NUMBER>
```

Then you can check that this VLAN is configured using

```
Switch# show VLAN
```

### ***3.4. Trunk and access ports***

This is a special kind of cable that is used in case we have VLANs. The main purpose of trunk is to manage the VLAN traffic. It uses a concept called tagging to mark each packet so each switch knows where to forward the traffic.

There are many cases that we can use a trunk. It can be used between two switches or between a router and switch in case of RoS. It can also be used between a layer three switch and ordinary switch as we will discuss in the coming experiment.

There are two ways in which a machine can be connected to a switch carrying 802.1Q VLAN traffic:

- via an access port, where VLAN support is handled by the switch (so the machine sees ordinary, untagged Ethernet frames); or
- via a trunk port, where VLAN support is handled by the attached machine (which sees 802.1Q-tagged Ethernet frames).

It is also possible to operate a switch port in a hybrid mode, where it acts as an access port for one VLAN and a trunk port for others (so the attached Ethernet segment carries a mixture of tagged and untagged frames). This is not recommended due to the potential for VLAN hopping.

### 3.4.1. *Initializing Switch Port as Trunk*

Configuring a trunk cable on **switch** is simple. You must access the needed port and perform the following command

```
Switch(config -if)# switchport mode trunk
```

When one end of a link is configured as a trunk, the other end changes automatically to trunk mode.

### 3.4.2. *Initializing Switch Port as Access*

To Assigning an interface to an existing VLAN. We must access the needed port and perform the following command

```
Switch(config -if)# switchport access VLAN <VLAN-NUMBER>
```

This command assigns the interface to VLAN with the VLAN-NUMBER. You can also assign a group of interfaces to a VLAN. To do that, we can use the following command that creates a range of interfaces, and then you can assign them to any VLAN using the access command.

```
Switch(config)#interface range <TYPE> <SLOT>/<START-PORT> - <END-PORT>
```

For example:

```
Switch(config)#interface range fastethernet0/1 - 20
```

Then assign them to a specific VLAN using the access command.

### ***3.5. Sub interface on Routers***

This is a part of a main interface on a router. It takes part of the bandwidth and passes special kind of traffic. It has also its own IP address and encapsulation number (which is used to tag traffic). Main interface does not have to get an IP address in case of sub interfaces.

#### ***3.5.1. Initializing a sub interface***

```
Router(config)# interface <TYPE> <SLOT>/<PORT>.<SUB-INTERFACE-NUMBER>
```

This command defines a sub-interface on the main interface, the type, slot and port are as known before the ones shown on the router interface, and the sub-interface number is the number for the virtual interface created inside the main interface.

#### ***3.5.2. Initializing IP address for a sub interface***

```
Router(config-subif)#encapsulation dot1Q <VLAN-ID>  
Router(config-subif)#ip address <IP-ADDRESS> <SUBNET-MASK>
```

These commands are used to add an IP address to a sub-interface, the encapsulation command is used to configure the sub interface as part of IEEE 802.1Q standards and the ip address is to add ip for that sub-interface



## 4. Procedure

### 4.1. Building the topology

Build the topology shown in Figure 7-1.

- For the routers use Router-PT
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

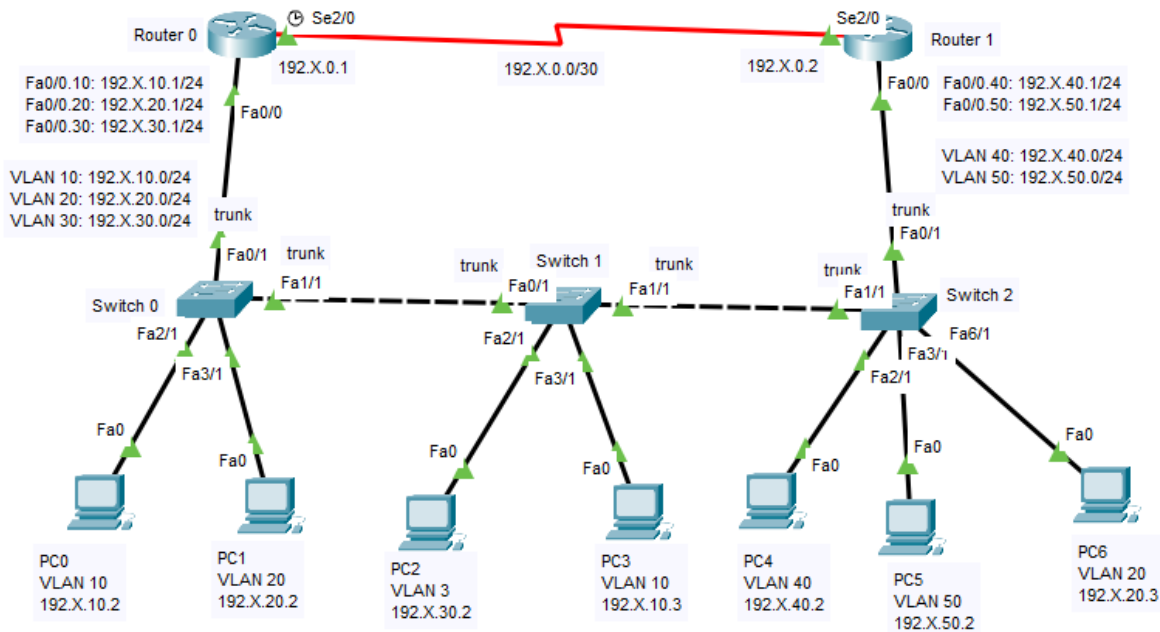


Figure 7-1 Topology

Use the following IPs shown in

Table 7-2 for the configuration. In order to configure the IPs for the PCs and the normal Routers interfaces (normal interfaces are shown red in

Table 7-2) see EXP. No. 2 sections 5.1.2 & 5.1.3.

For switch 2, you should add an extra interface physically using PT-SWITCH-NM-1CFE Module, as shown in the Figure below:



Figure 7-2 PT-SWITCH-NM-1CFE Module

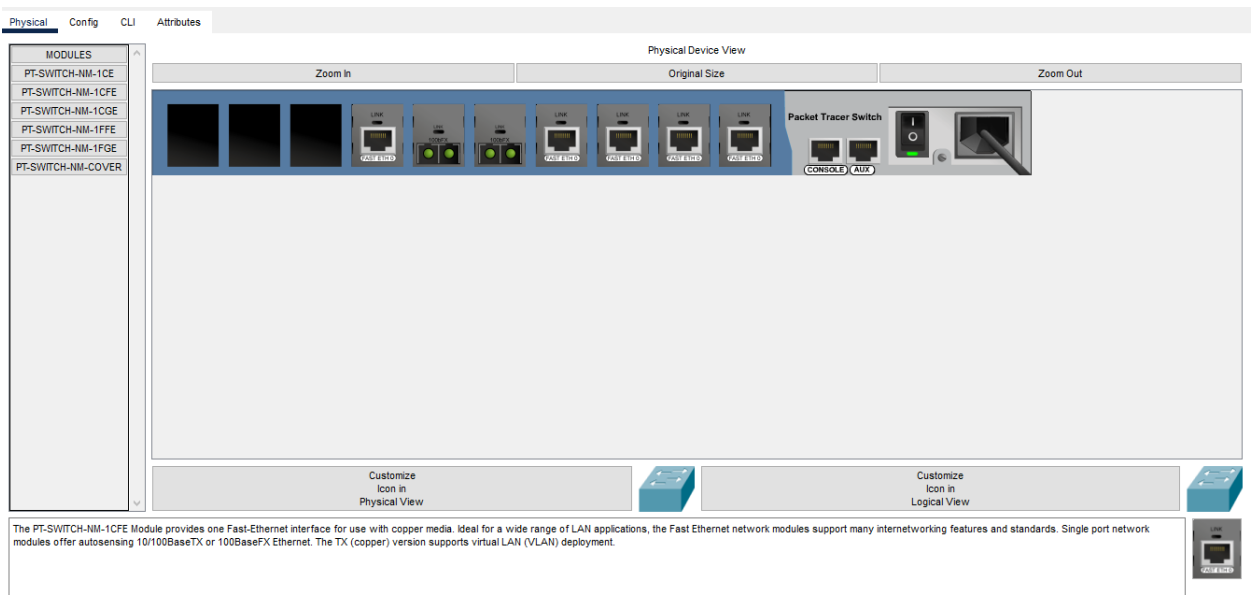


Figure 7-3 Adding interface to Switch 2

Table 7-2 Networks IPS

Area	Network	Device	Interface	IP	Subnet Mask	Wildcard Mask	VLAN Id
Area 0	192.X.0.0/30	Router 1	Se2/0	192.X.0.2	255.255.255.252	0.0.0.3	VLAN 1
		Router 0	Se2/0	192.X.0.1	255.255.255.252	0.0.0.3	VLAN 1
	192.X.10.0/24	Router 0	Fa0/0.10	192.X.10.1	255.255.255.0	0.0.0.255	VLAN 10
		PC0	Fa0	192.X.10.2	255.255.255.0	0.0.0.255	VLAN 10
		PC3	Fa0	192.X.10.3	255.255.255.0	0.0.0.255	VLAN 10
	192.X.20.0/24	Router 0	Fa0/0.20	192.X.20.1	255.255.255.0	0.0.0.255	VLAN 20
		PC1	Fa0	192.X.20.2	255.255.255.0	0.0.0.255	VLAN 20
		PC6	Fa0	192.X.20.3	255.255.255.0	0.0.0.255	VLAN 20
	192.X.30.0/24	Router 0	Fa0/0.30	192.X.30.1	255.255.255.0	0.0.0.255	VLAN 30
		PC2	Fa0	192.X.30.2	255.255.255.0	0.0.0.255	VLAN 30
	192.X.40.0/24	Router 1	Fa0/0.40	192.X.40.1	255.255.255.0	0.0.0.255	VLAN 40
		PC4	Se2/0	192.X.40.2	255.255.255.0	0.0.0.255	VLAN 40
	192.X.50.0/24	Router 1	Fa0/0.50	192.X.50.1	255.255.255.0	0.0.0.255	VLAN 50

		PC5	Fa0	192.X.50.2	255.255.255.0	0.0.0.255	VLAN 50
--	--	-----	-----	------------	---------------	-----------	---------

## 4.2. Configuring Routers

### 4.2.1. Configuring Routers Sub Interfaces

When we use the concept of Router on Stick in configuring VLANs, we must do a sub interface for each VLAN configured on the switch. A Sub interface acts as default gateway for a specific VLAN. The commands used for configuring a sub interface are:

```
Router(config)# interface <TYPE> <SLOT>/<PORT>.<SUB-INTERFACE-NUMBER>
Router(config-subif)#encapsulation dot1Q <VLAN-ID>
Router(config-subif)#ip address <IP-ADDRESS> <SUBNET-MASK>
```

Initializing a sub interface (Fa0/0.10) for Router 0 and VLAN 10 is done as follows:

```
Router(config)# interface Fa0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.X.10.1 255.255.255.0
```

▪ **Now you must complete configuring the IPs for VLAN 20 and VLAN 30 on router 0 and VLAN 40 and VLAN 50 on router 1 follow Use the following IPs shown in**

- 
- 
- 

▪ **Table 7-2 for the configuration. In order to configure the IPs for the PCs and the normal Routers interfaces (normal interfaces are shown red in**

- 
- 
-

- **Table 7-2). Use the following IPs shown in**
- 
- 
- 
- 
- **Table 7-2 for the configuration. In order to configure the IPs for the PCs and the normal Routers interfaces (normal interfaces are shown red in**
- 
- 
- 
- 
- **Table 7-2) see EXP. No. 2 sections 5.1.2 & 5.1.3.**
- **For switch 2, you should add an extra interface physically using PT-SWITCH-NM-1CFE Module, as shown in the Figure below:**



*Figure 7-2 PT-SWITCH-NM-1CFE Module*

*Figure 7-3 Adding interface to Switch 2*

- **Table 7-2 for knowing the IPs.**

#### ***4.2.2. Configuring OSPF Routing***

We will configure OSPF routing protocol for both routers 0 and 1 (make sure to add all networks connected to each router). In order to configure OSPF see EXP. No. 4 section 4.2.

Table 7-3 Switches Ports

Switch	Port	Kind	VLAN
Switch 0	Fa0/1	Trunk	---
	Fa1/1	Trunk	---
	Fa2/1	Access	VLAN 10
	Fa3/1	Access	VLAN 20
Switch 1	Fa0/1	Trunk	---
	Fa1/1	Trunk	---
	Fa2/1	Access	VLAN 30
	Fa3/1	Access	VLAN 10
Switch 2	Fa0/1	Trunk	---
	Fa1/1	Trunk	---
	Fa2/1	Access	VLAN 40
	Fa3/1	Access	VLAN 50
	Fa6/1	Access	VLAN 20

### 4.3. Configuring Switches

#### 4.3.1. Creating a VLAN

We can create a specific VLAN on a switch using the following command:

```
Switch(config)# VLAN <VLAN-NUMBER>
```

To initialize VLAN 10 of switch 0 we use:

```
Switch(config)# VLAN 10
Switch(config-vlan)# exit
```

- **Now VLAN 10 is configured inside switch 0. You must complete configuring the VLANs for switch 0, switch 1 and switch 2. (Each Switch must contain all the VLANs).**



### 4.3.2. Configuring Switch Access

Assigning an interface to an existing VLAN we must access the needed port and perform the access command:

```
Switch(config-if)# switchport access VLAN <VLAN-NUMBER>
```

To Assign port Fa2/1 on switch 0 to VLAN 10 we use the following commands.

```
Switch(config)# interface Fa2/1  
Switch(config-if)# switchport access VLAN 10
```

**Note** that if you did not create a VLAN and try to assign it to a port, the VLAN will be created automatically.

- **Now you need to configure the needed interfaces between each PC and switch to be access each with its VLAN on switch 0, switch 1 and switch 2.**

### 4.3.3. Configuring Switch Trunk

Assigning an interface to be a trunk is simple. You must access the needed port and perform the following command:

```
Switch(config -if)# switchport mode trunk
```

To Assign port Fa1/1 on switch 0 to be a trunk we use the following commands.

```
Switch(config)# interface Fa1/1  
Switch(config-if)# switchport mode trunk
```

When one end of a link is configured as a trunk, the other end changes automatically to trunk mode. This will also change Fa0/1 on switch 1 to become a trunk.

Now you need to configure every link between switch and switch or between router and a switch to be a trunk on switch 0, switch 1 and switch 2.

## 5. Todo (This part will be given to you by the instructor.)

Try to ping between PC1 and PC 6. Is it pinging correctly? If no, try to fix the issue.



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 8. Switching and VLANs 2 - Switch Virtual Interface**

### **1. Objectives**

- ❖ Learn how to configure a Cisco IOS Multi-layer Switch using the IOS command-line interface (CLI).
- ❖ Learn how to use switch simulator.
- ❖ Learn how to split Cisco Multi-layer Switch into multiple virtual ones and create VLANs.

### **2. Lab Requirements**

- ❖ Two Cisco router
- ❖ Six PCs.
- ❖ Three Cisco switches.
- ❖ One Cisco multi-layer switch 3560-24PS.
- ❖ several CAT5 straight-wired cable
- ❖ One Serial cable. (male and female).

## 3. Introduction

This experiment is the second part of VLANs. We discussed the first concept of VLANs which was Router on Stick. In this experiment we will go further by discussing another VLAN concept that is called Switch Virtual Interface by using a special kind of switches which is called multi-layer switch also referred as third layer switch.

### 3.1. *Third layer switch*

A layer three switch, as the name indicates, has some capabilities that are not found in an ordinary one. It can perform routing in parallel with switching. It combines the functionality of a switch and a router. It acts as a switch to connect devices that are on the same subnet or virtual LAN at lightning speeds and has IP routing intelligence built into it to double up as a router. It can support routing protocols, inspect incoming packets, and can even make routing decisions based on the source and destination addresses. This is how a layer 3 switch acts as both a switch and a router.

#### 3.1.1. *Features of a layer 3 switch*

- Comes with 24 Ethernet ports, but no WAN interface.
- Acts as a switch to connect devices within the same subnet.
- Switching algorithm is simple and is the same for most routed protocols.
- Performs on two OSI layers — layer 2 and layer 3.

#### 3.1.2. *Benefits of a layer 3 switch*

- Support routing between virtual LANs.
- Improve fault isolation.
- Simplify security management.
- Reduce broadcast traffic volumes.
- Ease the configuration process for VLANs, as a separate router isn't required between each VLAN.

- Separate routing tables, and as a result, segregate traffic better.
- Simplify troubleshooting as, fixing problems in L2 layer is tedious and time-consuming.
- Support flow accounting and high-speed scalability.
- Lower network latency as a packet does not have to make extra hops to go through a router.

### ***3.1.3. Disadvantages of layer 3 switch***

- Cost
- Limited application
- Lack of WAN functionality
- Multiple tenants and virtualization
- Lack of flexibility

## ***3.2. Configuring Third Layer Switch***

### ***3.2.1. Switch to Router link***

In order for a third layer switch port to work as a router port, we have to use the following command:

```
Switch(config-if)#no switchport  
Switch(config-if)# ip address <IP-ADDRESS> <SUBNET-MASK>
```

This command enables the interface to work as a router interface. It takes an IP address and Subnet mask.

### ***3.2.2. Enable routing***

For this kind of switch, we need to enable its ability to route packets as its default configuration would not do that. This can be done using the following command:

```
Switch(config)# ip routing
```

### 3.2.3. Create VLANs

We need to create our new VLANs on this switch as on a normal one, and then assign some ports for these VLANs, for creating VLANs and assigning them to ports we use the same commands used for a normal switch, see EXP. No. 7 Section 3.3.1 and Section 3.4.2

### 3.2.4. Switch Virtual Interfaces

we need to configure switch virtual interfaces on the switch to act as default gateways for the new VLANs. This can be done as follows:

```
Switch(config)#interface vlan <VLAN-NUMBER>  
Switch(config-if)# ip address <IP-ADDRESS> <SUBNET-MASK>
```

After this, a new VLAN interface will be created with the IP address assigned to it.

## 4. Procedure

In this Lab we will continue configuring using the topology of EXP. No. 7, we will add a multi-layer switch to the topology and add two new VLANs which are VLAN 60 and VLAN 70.

### 4.1. Building the topology

Build the topology shown in Figure 8-1.

- Use the topology built in EXP. No. 7
- For the Third layer switch user multi-layer switch **3560-24PS**
- For the PCs use PC-PT
- For the connections between the PCs and multi-layer switch use Automatically use connection type

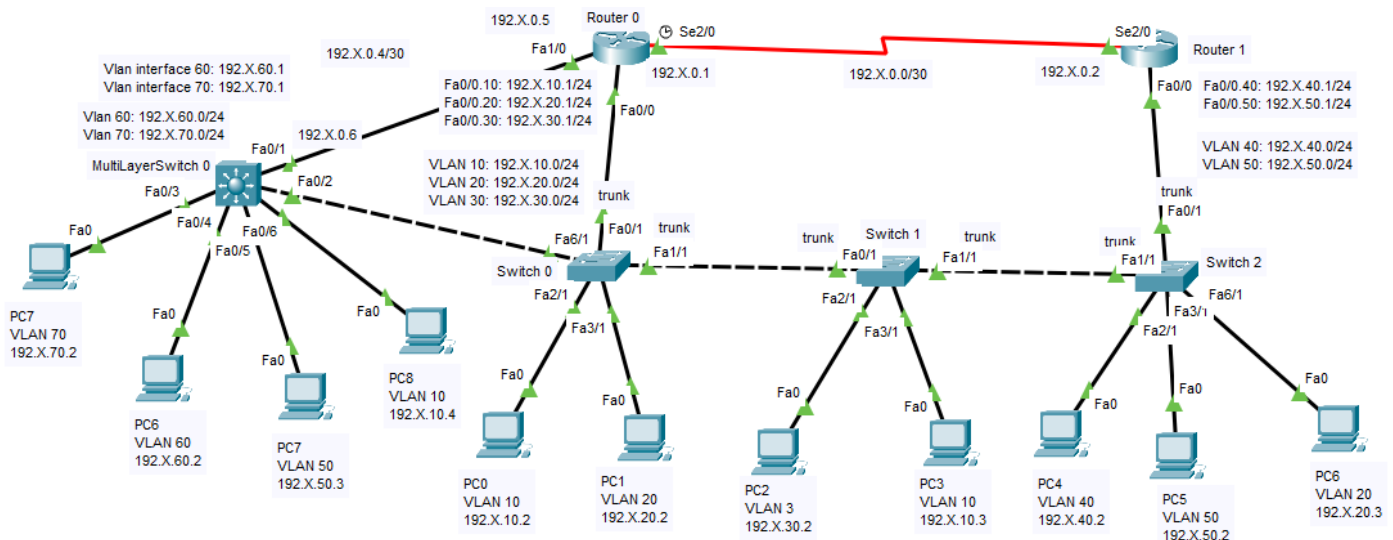


Figure 8-1 Topology

Use the following IPs shown in Table 8-1 for the configuration. In order to configure the IPs for the PCs and the normal Routers interfaces see EXP. No. 2 sections 5.1.2 & 5.1.3.

*Table 8-1 Newly Added Networks IPs*

Area	Network	Device	Interface	IP	Subnet Mask	Wildcard Mask	VLAN Id
Area 0	192.X.0.4/30	Router 0	Fa1/0	192.X.0.5	255.255.255.252	0.0.0.3	VLAN 1
		MLS 0	Fa0/1	192.X.0.6	255.255.255.252	0.0.0.3	VLAN 1
	192.X.10.0/24	PC10	Fa0	192.X.10.4	255.255.255.0	0.0.0.255	VLAN 10
	192.X.50.0/24	PC9	Fa0	192.X.50.3	255.255.255.0	0.0.0.255	VLAN 50
	192.X.60.0/24	MLS 0	VLAN 60	192.X.60.1	255.255.255.0	0.0.0.255	VLAN 60
		PC8	Fa0	192.X.60.2	255.255.255.0	0.0.0.255	VLAN 60
	192.X.70.0/24	MLS 0	VLAN 70	192.X.70.1	255.255.255.0	0.0.0.255	VLAN 70
		PC7	Fa0	192.X.70.2	255.255.255.0	0.0.0.255	VLAN 70

**\*\*Multi-layer switch 0 → MLS 0**

## 4.2. Configuration

### 4.2.1. Multi-Layer Switch to Router link

We need to add an IP address to the switch port connected to the router, so firstly we need to change the switch port to a router port and then add an IP address, to do that we will use the following command:

```
Switch(config-if)#no switchport
Switch(config-if)#ip address <IP-ADDRESS> <SUBNET-MASK>
```

To add IP address 192.X.0.5/30 to port Fa0/1 on the multi-layer switch we use the commands below:

```
Switch(config)#interface fa0/1
Switch(config-if)#no switchport
Switch(config-if)#ip address 192.X.0.6 255.255.255.252
```

- **Now we added an IP address to the third layer switch.**

#### ***4.2.2. Multi-Layer Switch Configuring VLAN Interfaces IPs (Switch Virtual Interfaces)***

We will use the following commands to configure switch virtual interfaces on the switch to act as default gateways for the new VLANs

```
Switch(config)#interface vlan <VLAN-NUMBER>
Switch(config-if)# ip address <IP-ADDRESS> <SUBNET-MASK>
```

To configure an IP address for VLAN 60 with an IP of 192.X.60.1 use the following commands:

```
Switch(config)#interface vlan 60
Switch(config-if)# ip address 192.X.60.1 255.255.255.0
```

- **Now configure switch virtual interface for VLAN 70**

#### ***4.2.3. Enable routing on Multi-Layer Switch and configuring OSPF***

We will configure OSPF routing protocol for both routers 0 the Multi-layer switch. To configure OSPF see EXP. No. 4 section 4.2. By default, the routing is disabled on the third layer switch, in order to enable it we use the following command:

```
Switch(config)# ip routing
```



#### ***4.2.4. Configuring VLANs on Multi-Layer Switch***

Configuring VLANs on a multi-layer switch is the same as configuring them on a normal switch, see EXP. No. 7 Section 4.3.1 and apply it here.

#### ***4.2.5. Configuring Access Ports on Multi-Layer Switch***

Configuring access ports on a multi-layer switch is the same as configuring them on a normal switch, see EXP. No. 7 Section 4.3.2 and apply it here.

#### ***4.2.6. Configuring Trunk on Multi-Layer Switch***

To configure a trunk on a third layer switch, we need to encapsulate that switch, to do this on interface Fa0/1 use the following commands:

```
Switch(config)#interface Fa0/1
Switch(config-if)#switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk
```

### **5. Todo (This part will be given to you by the instructor.)**

Try to ping between PC9 and PC 10. Is it pinging correctly? If no, try to fix the issue.

This part will be given to you by the instructor.



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 9. Internet Protocol Version 6 (IPv6) Configuration**

### **1. Objectives**

- ❖ Learn how to configure a Cisco IOS router using the IOS command-line interface (CLI).
- ❖ Learn how to use router simulator.
- ❖ Learn how to configure and verify IPv6 routing with Cisco routers.
- ❖ Static IPv6 routing
- ❖ Dynamic routing RIPng

### **2. Lab Requirements**

- ❖ Three Cisco routers 2811.
- ❖ Five PCs.
- ❖ Two Cisco switches 2950.
- ❖ Several CAT5 straight-wired cables.
- ❖ Two Serial cable. (male and female).

### 3. Introduction

You've no doubt heard the news stating that we will run out of IP addresses. The Number Resource Organization (NRO) reported that as of February 3, 2011, the free pool of IPv4 address space has been depleted.

IPv6 uses 128-bit addresses. That equates to  $3.40292367 \times 10^{38}$  addresses. IPv6 addresses do not use decimals like IPv4. They are composed of groups of four-digit hexadecimal numbers separated by colons. Luckily, leading zeros can be eliminated within each set of colons. For example:

```
AA76:0000:0000:0000:0012:A322:FE33:2267
```

May be represented as:

```
AA76:0:0:0:12:A322:FE33:2267
```

Additionally, any consecutive number of zeros can be replaced by a double colon once per address. Thus:

```
AA76:0000:0000:0000:0012:A322:FE33:2267
```

May also be written as:

```
AA76::12:A322:FE33:2267
```

Double colons can appear once in an IP address and can't exist more than once. For example:

```
AA76:0000:0000:0012:A322:0000:0000:2267
```

This can be either be written as:

```
AA76::12:A322:0:0:2267
```

Or as

```
AA76:0:0:12:A322::2267
```

But we can not write it as

```
AA76::12:A322::2267
```

Where two double colon appeared in the address.

### ***3.1. Address Types***

#### ***3.1.1. Unicast***

Packets addressed to a unicast address are delivered to a single interface.

#### ***3.1.2. Global unicast addresses***

These are like the public addresses in IPv4. Global addresses start at 2000::/3

#### ***3.1.3. Link-local addresses***

These are like the private addresses in IPv4 in that they're not meant to be routed and they start with FE80::/10. Think of them as a handy tool that gives you the ability to throw a temporary LAN together for meetings or to create a small LAN that's not going to be routed but still needs to share and access files and services locally.

#### ***3.1.4. Multicast***

Again, same as in IPv4, packets addressed to a multicast address are delivered to all interfaces tuned into the multicast address.

#### ***3.1.5. Anycast***

Like multicast addresses, an anycast address identifies multiple interfaces on multiple devices, but there is a big difference: The anycast packet is delivered to only one device—actually, to the closest one it finds defined in terms of routing distance.

### ***3.2. Reserved IPv6 addresses***

You are probably wondering if there are any special, reserved addresses in IPv6 because you know they are there in IPv4. Well there are—plenty of them! Let us go over some of them now:

- 0:0:0:0:0:0:1 Equals ::1. The equivalent of 127.0.0.1 in IPv4.
- 0:0:0:0:0:192.X.100.1 This is how an IPv4 address would be written in a mixed IPv6/IPv4 network environment.
- 2000::/3 The global unicast address range.
- FE80::/10 The link-local unicast range.

### 3.3. Configuring Cisco Routers with IPv6:

To configure an IPv6 address on an interface. You use the interface configuration command:

```
Router(config)#interface <TYPE> <SLOT>/<PORT>
Router(config-if)# ipv6 address <IPV6-PREFIX>/<PREFIX-LENGTH>
```

Now the chosen interface now has the IPv6 address assigned to it. Note that an interface can have more than one IPv6 address.

### 3.4. IPv6 Routing Protocols:

To enable IPv6 routing on a router, you must use the `ipv6 unicast-routing` global configuration command:

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

By default, IPv6 traffic forwarding is disabled, so using this command enables it.

#### 3.4.1. Static routing

To configure a static route, you can use this configuration command:

```
Router(config)#ipv6 route <IPV6-PREFIX>/<PREFIX-LENGTH> <IPV6-NEXT-HOP-ADDRESS>
```

This is a simple entry to the routing table so in order for the router to route the <IPV6-PREFIX>/<PREFIX-LENGTH> the router should pass the packet to the IPV6-NEXT-HOP-ADDRESS.

### 3.4.2. RIPng

the primary features of RIPng are the same as they were with RIPv2. It is still a distance-vector protocol, has a max hop count of 15.

But of course, there are differences in the new version, or it wouldn't be a new version, would it? We know that routers keep the next-hop addresses of their neighbor routers for every destination network in their routing table.

The difference is that with RIPng, the router keeps track of this next-hop address using the link-local address, not a global address.

Probably one of the biggest changes with RIPng is the fact that you configure or enable the advertisement of a network from interface configuration mode instead of with a network command in router configuration mode. So, in RIPng's case, if you enable it directly on an interface without going into router configuration mode and starting a RIPng process, a new RIPng process will simply be started for you. It will look something like this:

```
Router(config)#interface <TYPE> <SLOT>/<PORT>
Router(config)#ipv6 rip <RIP-ID> enable
```

The RIP-ID you see in this command is a tag (that can also be named rather than numbered) that identifies the process of RIPng that's running, and as I said, this will start a process of RIPng so you don't have to go into router configuration mode.

### 3.5. Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is a device-discovery protocol that runs on all Cisco network equipment. Each device sends identifying messages to a multicast address, and each device monitors the messages sent by other devices. Information in CDP packets are used in network management software and to share information about other directly connected Cisco equipment, such as the operating system version and IP address.

## 4. Procedure

In this lab we will connect three routers and several PCs on different networks using IPv6 IPs. This will require configuring routing protocols between the routers. We will configure static routing and dynamic routing (RIPng) as routing protocols.

### 4.1. Building the topology

Build the topology shown in Figure 9-1

- For the routers use Router-2811
- For the switches use Switch-PT
- For the PCs use PC-PT
- For the connections between the PCs, switches and routers use Automatically use connection type

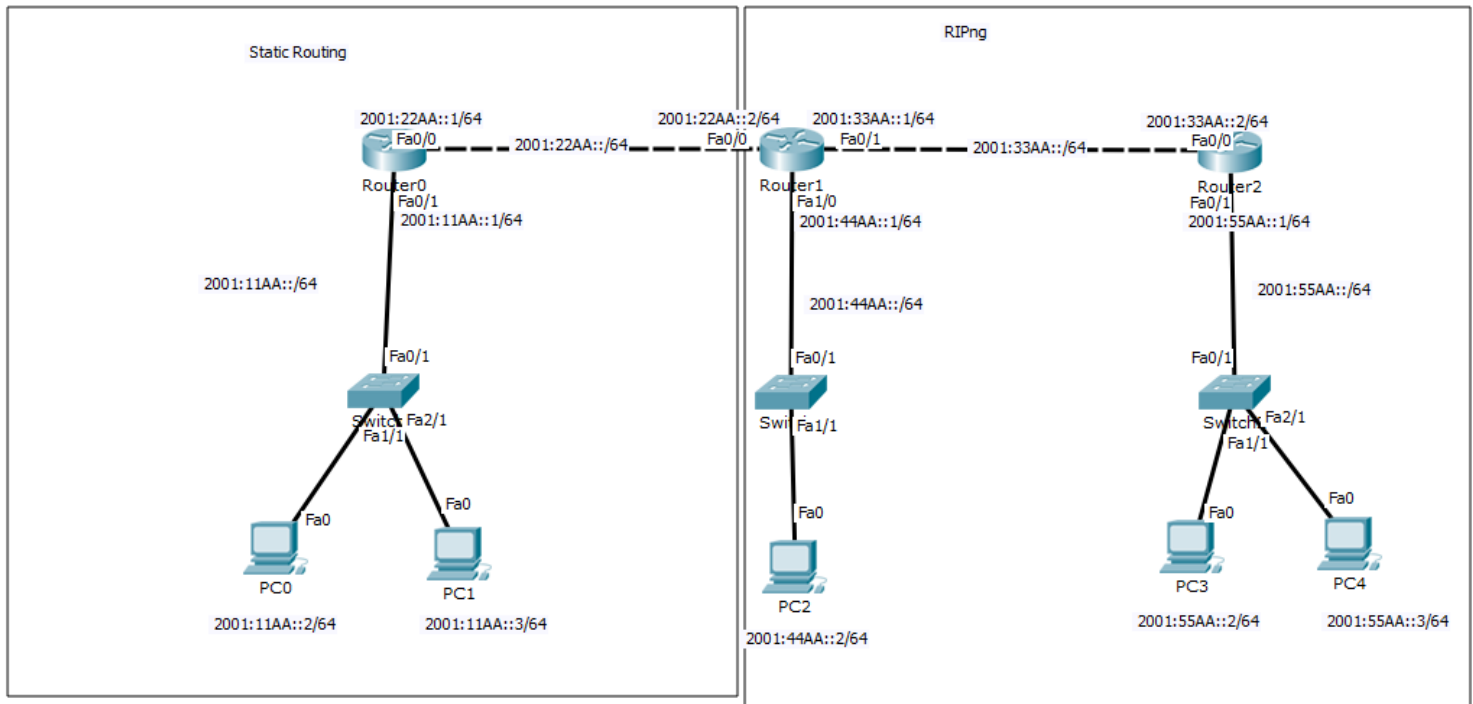


Figure 9-1 Topology

Use the following IPs shown in Table 9-1 for the configuration.

*Table 9-1 Networks IPs*

Network	Device	Interface	IP	Subnet Mask
2001:11AA:: /64	Router 0	Fa0/1	2001:11AA::1	/64
	PC0	Fa0	2001:11AA::2	/64
	PC1	Fa0	2001:11AA::3	/64
2001:22AA:: /64	Router 0	Fa0/0	2001:22AA::1	/64
	Router 1	Fa0/0	2001:22AA::2	/64
2001:33AA:: /64	Router 1	Fa0/1	2001:33AA::1	/64
	Router 2	Fa0/0	2001:33AA::1	/64
2001:44AA:: /64	Router 1	Fa1/0	2001:44AA::1	/64
	PC3	Fa0	2001:44AA::2	/64
2001:55AA:: /64	Router 0	Fa0/1	2001:55AA::1	/64
	PC3	Fa0	2001:55AA::2	/64
	PC4	Fa0	2001:55AA::3	/64



### 4.1.1. Configuring IPv6 for the PCs

- Click on the PC0 and go to desktop tab
- Choose IP configuration to add an IPv6 for the PC as shown in Figure 9-2.

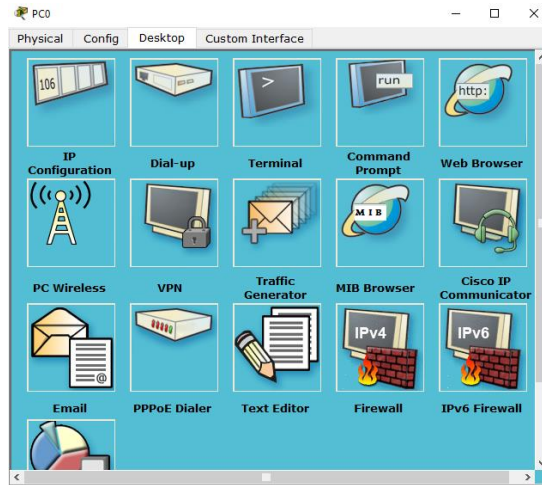


Figure 9-2 PC Desktop

- Add the following IPv6 address (2001:11AA::2/64) as shown in Figure 2-3.
- Add the IPv6 gateway address (2001:11AA::1) as shown in Figure 2-3.

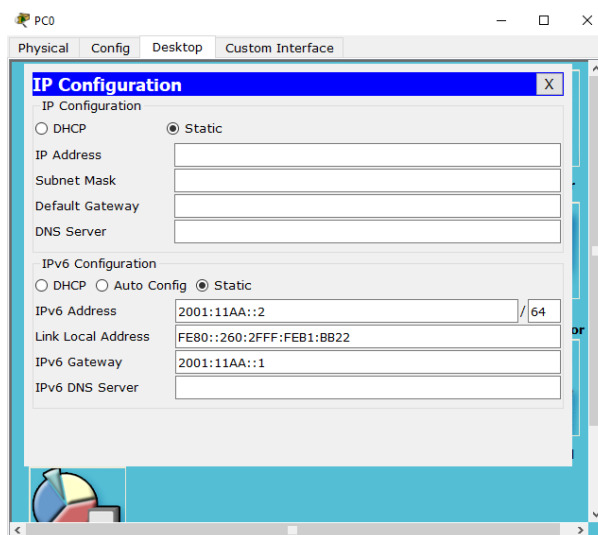


Figure 9-3 PC0 IP address

Repeat the previous steps for PC1, PC2, PC3 and PC4 with the IPv6 shown in Table 9-1.

### 4.1.2. Configuring IPv6 for the routers

To configure an IPv6 for any router, access the interface you want to add the IPv6 using the following command:

```
Router(config)#interface <TYPE> <SLOT>/<PORT>
```

Then add an IPv6 to that interface using the following command:

```
Router(config-if)# ipv6 address <IPV6-PREFIX>/<PREFIX-LENGTH>
```

For Example, to add an IPv6 (2001:11aa::1/64) to the interface Fa0/1 on router 0 then we will use the following commands

```
Router(config)#interface Fa0/1  
Router(config-if)# ipv6 address 2001:11AA::1/64
```

Do not forget to turn on the interface using the following command:

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

- **Now repeat the previous step for all interfaces on routers 0,1 and 2 with the correct IPv6 shown in Table 9-1.**

### 4.1.3. Configuring routing protocols

To enable IPv6 routing on a router, you must use the ipv6 unicast-routing global configuration command:

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

We will enable the routing protocols for all routers (router 0, router1 and router 2).

### 4.1.4. Configuring Static routing

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

```
Router(config)#ipv6 route <IPV6-PREFIX>/<PREFIX-LENGTH> <IPV6-NEXT-HOP-ADDRESS>
```

This list describes each command in the string:

- **Ipv6 route:** The command used to create the static route.
- **IPV6-PREFIX:** The network you're placing in the routing table.
- **PREFIX-LENGTH:** The subnet mask being used on the network.
- **IPV6-NEXT-HOP-ADDRESS:** The address of the next-hop router that will receive the packet and forward it to the remote network. This is a router interface that's on a directly connected network. You must be able to ping the router interface before you add the route. If you type in the wrong next hop address, or the interface to that router is down, the static route will show up in the router's configuration, but not in the routing table.

For example, to configure Router 0 to network 2001:44AA::/64 the following command is used:

**IPV6-PREFIX:** the network Router 0 is not connected to 2001:44AA::/64.

**PREFIX-LENGTH:** the subnet mask being used on the destination network /64

**IPV6-NEXT-HOP-ADDRESS:** is the address of Fa0/0 on Router 1 (2001:22AA::2)

```
Router(config)# ipv6 route 2001:44AA::/64 2001:22AA::2
```

- **Repeat this step 4.1.4 for router 1 and router 2 with the only the needed (ask if you do not know) and correct addresses.**

#### 4.1.5. Configuring RIPng routing protocol

To Enable RIPng routing protocol, we only need to enable the interfaces to work as RIPng to do that we need to access the interfaces needed to be configured as RIPng using this command:

```
Router(config)#interface <TYPE> <SLOT>/<PORT>
```

Then enable RIPng using the following command:

```
Router(config)#ipv6 rip <RIP-ID> enable
```

To enable RIP on router 2 on interface Fa0/0 use the following commands:

```
Router(config)#interface Fa0/0
Router(config-if)#ipv6 rip 1 enable
```

- **Repeat step 4.1.5 for router 2 Fa0/1 and for router 1 interfaces Fa0/1 and Fa1/0 only.**

## 4.2. Verifying Your Configuration

Every PC should ping with each node on the network. Make sure that PC4 pings with PC0.

## 5. Todo (This part will be given to you by the instructor.)

### 5.1. Monitoring and Maintaining CDP

try the following commands in privileged mode on R1. Discuss the outputs with your teacher.

- Router# sh cdp neighbors
- Router # sh cdp neighbors detail

### 5.2. Disabling CDP on an Interface

CDP is enabled by default on all supported interfaces to send and receive CDP information. To disable CDP on an interface, follow these steps, beginning in privileged mode:

```
Router(config-if)#no cdp enable
```

### 5.3. Disabling CDP

To disable the CDP device discovery capability, follow these steps, beginning in privileged mode:

```
Router(config)#no cdp run
```



**Birzeit University**  
**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
**Computer Networks Laboratory ENCS413**

## **EXP. No. 10. Packet Sniffing and Domain Name System (DNS)**

### **1. Objectives**

- ❖ Recognizing and decoding certain packets of interest using Wireshark Ethernet packet sniffing tool.
- ❖ Learn about various network protocols such as IP, TCP, and ICMP.

### **2. Lab Requirements**

- ❖ PC with Network Interface Card (NIC) connected to a network
- ❖ Wireshark tool installed on the PC [ <http://www.wireshark.org/download.html> ]
- ❖ PingPlotter tool installed on the PC [ <http://www.pingplotter.com> ]

### 3. Introduction

One's understanding of network protocols can often be greatly deepened by "seeing protocols in action" and by "playing around with protocols" – observing the sequence of messages exchanged between two protocol entities, delving down into the details of protocol operation, and causing protocols to perform certain actions and then observing these actions and their consequences. This can be done in simulated scenarios or in a "real" network environment such as the Internet.

#### 3.1. *Packet Sniffer*

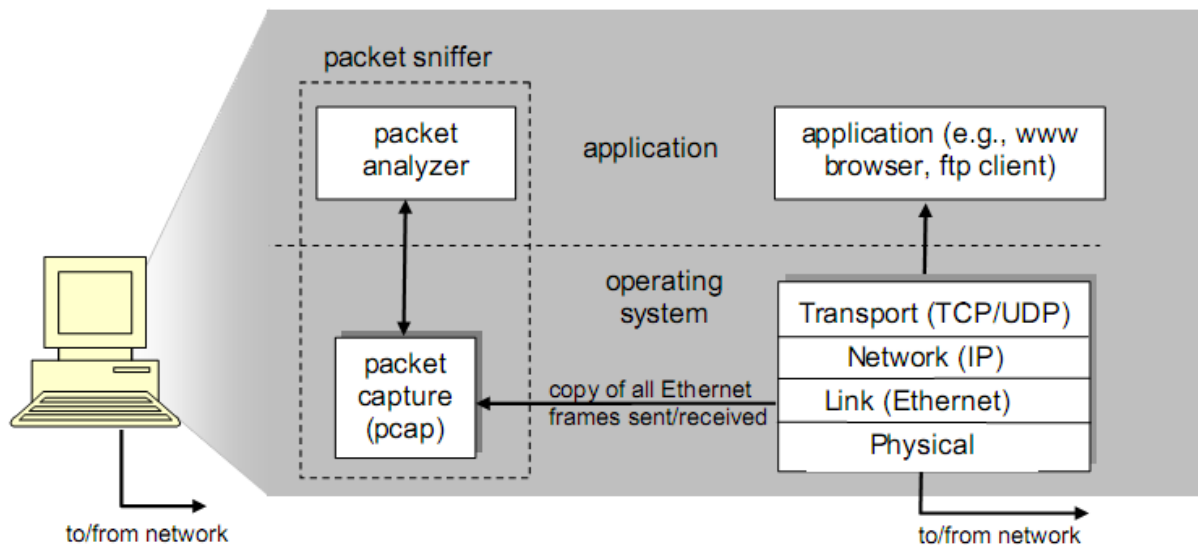
Packet sniffer is a program that captures all of the packets of data that pass through a given network interface, and recognizes and decodes certain packets of interest without modifying it. A packet sniffer is sometimes referred to as a network monitor, or network analyzer. It is normally used by network or system administrator to monitor and troubleshoot network traffic. However, it is sometimes also used by malicious intruders for illicit purpose such as stealing a user's password or credit-card number. By comparison, a firewall sees all of a computer's packet traffic as well, but it has the ability to block and drop any packets that its programming dictates. Packet sniffers merely watch, display, and log this traffic.

One disturbingly powerful aspect of packet sniffers is their ability to place the hosting machine's network adapter into "promiscuous mode." Network adapters running in promiscuous mode receive not only the data directed to the machine hosting the sniffing software, but also ALL of the traffic on the physically connected local network. Unfortunately, this capability allows packet sniffers to be used as potent spying tools. A packet sniffer can only capture packets within a given subnet.

The use of powerful packet sniffing software by people who lack a thorough understanding of TCP/IP and Internet protocols will — without question — create significant confusion and raise a large number of questions.

Figure 10-1 shows the structure of a packet sniffer. At the right of Figure 10-1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in

Figure 10-1 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Recall that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 10-1, the assumed physical media is an Ethernet, and so all upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.



*Figure 10-1 Packet Sniffer Structure*

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 10-1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol.

### ***3.2. Packet Sniffing Tool***

There exist various commercial and free packet sniffer tools. We will be using the Wireshark packet sniffer [<http://www.wireshark.org/>] for this experiment, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It is an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support, rich functionality that includes the capability to analyze hundreds of protocols, and a well-designed user interface. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs, and ATM connections.

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the libpcap or WinPCap packet capture library. The libpcap software will be installed for you if it is not installed within your operating system when you install Wireshark. See <http://www.wireshark.org/download.html> for a list of supported operating systems and download sites.

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 10-2 will be displayed. Initially, no data will be displayed in the various windows.

In this experiment you will be running various network applications in different scenarios using a computer on your desk, at home, or in a lab. You will observe the network protocols in your computer “in action,” interacting and exchanging messages with protocol entities executing elsewhere on the Internet.



Command Menus

Display filter specification

List of captured packets

Details of selected packet header

Packet content in hexadecimal and ASCII

The screenshot shows the Wireshark graphical user interface. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 1) is a TLSv1.2 packet of length 110 bytes, containing Application Data. The details pane below shows the packet structure: Frame 1 (110 bytes on wire), Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet content pane shows hexadecimal and ASCII representations of the data.

Figure 10-2 Wireshark Graphical User Interface

▪ Can We make a packet analyzing on the data link layer? If yes, what are the purposes of packet sniffing under MAC layer?

### 3.3. Domain Name Server (DNS)

The Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we will take a closer look at the client side of DNS. Recall that the client’s role in the DNS is relatively simple – a client sends a query to its local DNS server, and receives a response back. much can go on “under the covers,” invisible to the DNS clients, as the hierarchical DNS servers communicate with each other to either recursively or

iteratively resolve the client's DNS query. From the DNS client's standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

Before beginning this lab, you will probably want to review DNS by reading Section 2.5 of the text. In particular, you may want to review the material on local DNS servers, DNS caching, DNS records and messages, and the TYPE field in the DNS record.

## 4. Procedure

### 4.1. Part 1 Packet Sniffing

In this part of the lab we will look on how can we capture HTTP request and IP sniffing using Wireshark software follow up the following steps.

#### 4.1.1. Running the Wireshark program

After installing the Wireshark program start by running the application on your computer. The program initially will look as shown in Figure 10-3, you will select the port that have traffic on, in our case it can be Ethernet or Wi-Fi.

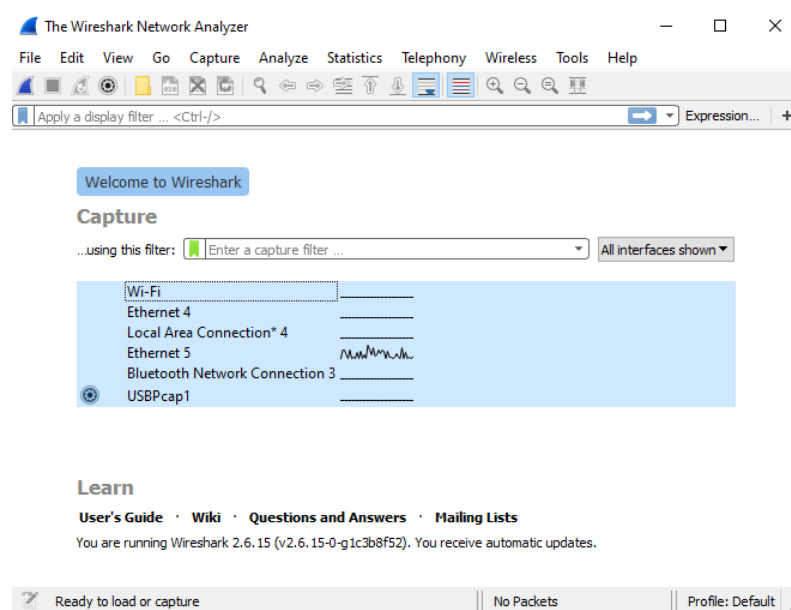




Figure 10-3 Wireshark Home Page Program

#### 4.1.2. Start capturing for requesting a web site

In the top left side of the command menus you will see the Wireshark blue Icon [  ] this will start capturing any packet on the network. Besides it, there is a red button [  ] this will stop capturing the packets in order to analyze.

➤ Starting Capturing

We will start capturing the packets by clicking on the blue button.

➤ Requesting HTTP request

We will try reaching this web site [<http://http-login.badssl.com/>] using our web browser, after filtering specific packets shown in Figure 10-4. We can notice the following:

- No ARP requests and responses because the MAC address of the router is already known.
- Packet No. 1 DNS requests, the source IP 192.X.1.231 in the PC IP address, the destination IP 192.X.1.1 is the router IP address the Protocol is DNS and the info shows that it is requesting the IP address for http-login.bandssl.com domain name.
- Packet No. 2 is a DNS response, the source IP 192.X.1.1 is the router IP, the destination IP 192.X.1.231 is the PC IP address, the Protocol is DNS and the info shows that http-login.bandssl.com IP address is 104.154.89.105.
- Packets 3, 4 and 5 are a 3-way handshaking source IP 192.X.1.231 is the PC, the destination IP address is 104.154.89.105 which is the http-login.bandssl.com server IP address. From the info we can see that packet No 3 is a [SYN] packet (synchronize), packet No 4 is a response from the server [SYN, ACK] packet (synchronize and acknowledgment) and Packet No 5 is a [ACK] packet (acknowledgment) from the user.
- Packets 6-22 are a HTTP GET requests and responses for requesting the web page. You can see that after each packet there is a TCP packet which is an [ACK] this specifies that the HTTP packet has reached the destination successfully. Notice that the HTTP requests are separated into multiple ones:
  - ◆ packet No 6 is for requesting the HTML text files → the response is packet No 8.
  - ◆ Packet No 10 is for requesting the style.css file → the response is packet No 12.
  - ◆ Packet No 14 is for requesting an icon called favicon-red.ico → the response is packet No 21.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.231	192.168.1.1	DNS	81	Standard query 0x7cb6 A http-login.badssl.com
2	0.218035	192.168.1.1	192.168.1.231	DNS	215	Standard query response 0x7cb6 A http-login.badssl.com A 104.154.89.105 NS ns-cloud-d3.google
3	0.220105	192.168.1.231	104.154.89.105	TCP	66	52998 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.399144	104.154.89.105	192.168.1.231	TCP	66	80 → 52998 [SYN, ACK] Seq=0 Ack=1 Win=28400 Len=0 MSS=1360 SACK_PERM=1 WS=128
5	0.399526	192.168.1.231	104.154.89.105	TCP	54	52998 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
6	0.405092	192.168.1.231	104.154.89.105	HTTP	585	GET /?fbclid=IwAR1M1K1Qp_qQXyeJ40fpg077Y1PeyJs3kCV3L4hfP7bsF62PV2EyFDCKFrS HTTP/1.1
7	0.586859	104.154.89.105	192.168.1.231	TCP	60	80 → 52999 [ACK] Seq=1 Ack=532 Win=231 Len=0
8	0.587556	104.154.89.105	192.168.1.231	HTTP	883	HTTP/1.1 200 OK (text/html)
9	0.628356	192.168.1.231	104.154.89.105	TCP	54	52999 → 80 [ACK] Seq=532 Ack=830 Win=512 Len=0
10	0.686339	192.168.1.231	104.154.89.105	HTTP	464	GET /style.css HTTP/1.1
11	0.868491	104.154.89.105	192.168.1.231	TCP	1414	80 → 52999 [ACK] Seq=830 Ack=942 Win=239 Len=1360 [TCP segment of a reassembled PDU]
12	0.869296	104.154.89.105	192.168.1.231	HTTP	472	HTTP/1.1 200 OK (text/css)
13	0.869871	192.168.1.231	104.154.89.105	TCP	54	52999 → 80 [ACK] Seq=942 Ack=2608 Win=515 Len=0
14	0.995772	192.168.1.231	104.154.89.105	HTTP	497	GET /icons/favicon-red.ico HTTP/1.1
15	1.178081	104.154.89.105	192.168.1.231	TCP	1414	80 → 52999 [ACK] Seq=2608 Ack=1385 Win=247 Len=1360 [TCP segment of a reassembled PDU]
16	1.179235	104.154.89.105	192.168.1.231	TCP	1414	80 → 52999 [ACK] Seq=3968 Ack=1385 Win=247 Len=1360 [TCP segment of a reassembled PDU]
17	1.179474	192.168.1.231	104.154.89.105	TCP	54	52999 → 80 [ACK] Seq=1385 Ack=5328 Win=515 Len=0
18	1.180388	104.154.89.105	192.168.1.231	TCP	1414	80 → 52999 [ACK] Seq=5328 Ack=1385 Win=247 Len=1360 [TCP segment of a reassembled PDU]
19	1.181092	104.154.89.105	192.168.1.231	TCP	1414	80 → 52999 [ACK] Seq=6688 Ack=1385 Win=247 Len=1360 [TCP segment of a reassembled PDU]
20	1.181346	192.168.1.231	104.154.89.105	TCP	54	52999 → 80 [ACK] Seq=1385 Ack=8048 Win=515 Len=0
21	1.182619	104.154.89.105	192.168.1.231	HTTP	321	HTTP/1.1 200 OK (image/x-icon)
22	1.222336	192.168.1.231	104.154.89.105	TCP	54	52999 → 80 [ACK] Seq=1385 Ack=8315 Win=514 Len=0

Figure 10-4 Packets for http-login web site

- Now try to send a POST request with the EMAIL and PASSWORD and try to retrieve them from the HTTP request using Wireshark sniffing tool.

### 4.1.3. IP (Internet Protocol) Sniffing

In this part of the laboratory, we will investigate the IP protocol, focusing on the IP datagram. We will do so by analyzing a trace of IP datagrams sent and received by an execution of the tracerout program. We will investigate the various fields in the IP datagram, and study IP fragmentation in detail.

#### ➤ Running traceroute command

We will want to run traceroute and have it send datagrams of various lengths. The tracert program (used for our ICMP Wireshark lab) provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is pingplotter, available both in free version and shareware versions at <http://www.pingplotter.com>. The size of the ICMP echo request message can be explicitly set in pingplotter by selecting the menu item Edit>Options>Engine Options and then filling in the Packet Size field. The default packet size is 56 bytes. Once pingplotter has sent a series of packet with the

increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting Trace Interval amount of time. The value of Trace Interval and the number of intervals can be explicitly set in pingplotter.

- Startup Wireshark and begin packet capture (Capture->Option) and then press OK on the Wireshark Packet Capture Options screen (we will not need to select any options here).
- Startup pingplotter and enter the name of a target destination in the “Address to Trace Window”. Select the menu item Edit>Options>Engine and enter a value of 56 in the Packet Size field and then press OK. Then press the Trace button.
- Next, send a set of datagrams with a longer length, by selecting Edit>Options>Engine and enter a value of 2000 in the Packet Size field and then press OK. Then press the Resume button.
- Finally, send a set of datagrams with a longer length, by selecting Edit>Options>Engine and enter a value of 3500 in the Packet Size field and then press OK. Then press the Resume button.
- Stop Wireshark tracing.

➤ **A look at the captured trace**

In your trace, you should be able to see the series of ICMP Echo Request sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. Whenever possible, when answering a question, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. To print a packet, use File->Print, choose Selected packet only, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

1. Select the first ICMP Echo Request message sent by your computer and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?
2. Within the IP packet header, what is the value in the upper layer protocol field?

3. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.
4. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Next, sort the traced packets according to IP source address by clicking on the Source column header; a small downward pointing arrow should appear next to the word Source. If the arrow points up, click on the Source column header again. Select the first ICMP Echo Request message sent by your computer and expand the Internet Protocol portion in the “details of selected packet header” window. In the “listing of captured packets” window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use the down arrow on your keyboard to move through the ICMP messages sent by your computer.

5. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?
6. Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?
7. Describe the pattern you see in the values in the Identification field of the IP datagram.

Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

8. What is the value in the Identification field and the TTL field?
9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

To study IP Fragmentation, sort the packet listing according to time again by clicking on the Time column.

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?
12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?
13. What fields change in the IP header between the first and second fragment?

Now find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 3500.

14. How many fragments were created from the original datagram?
15. What fields change in the IP header among the fragments?

## **4.2. Part 2 DNS sniffing:**

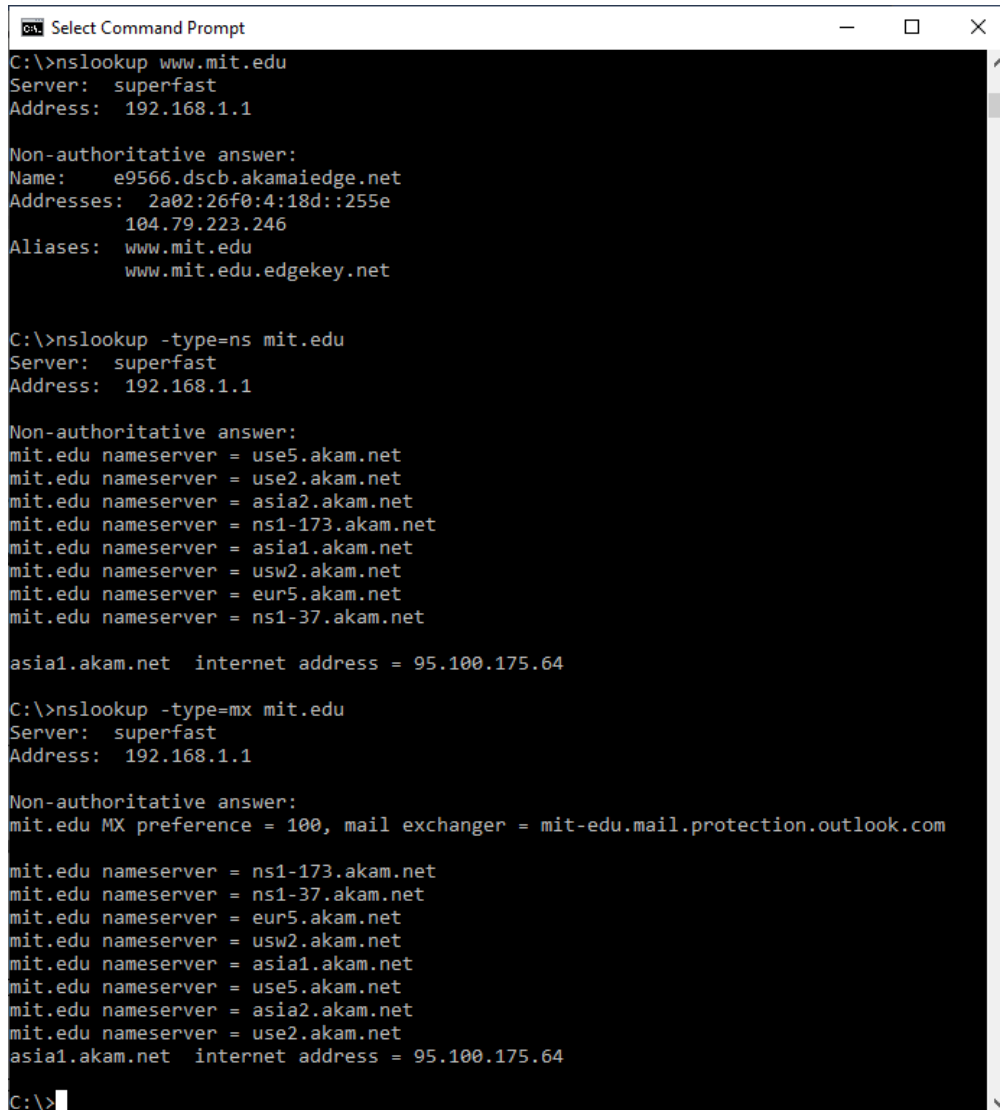
In this part of the lab, we will take a closer look at the client side of DNS. Recall that the client's role in the DNS is relatively simple – a client sends a query to its local DNS server, and receives a response back.

### **4.2.1. nslookup**

we'll make extensive use of the nslookup tool, which is available in most Linux/Unix and Microsoft platforms today. To run nslookup in Linux/Unix, you just type the nslookup command on the command line. To run it in Windows, open the Command Prompt and run nslookup on the command line.

In its most basic operation, nslookup tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, nslookup sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.





```

C:\>nslookup www.mit.edu
Server:      superfast
Address:    192.168.1.1

Non-authoritative answer:
Name:       e9566.dscb.akamaiedge.net
Addresses:  2a02:26f0:4:18d::255e
            104.79.223.246
Aliases:   www.mit.edu
            www.mit.edu.edgekey.net

C:\>nslookup -type=ns mit.edu
Server:      superfast
Address:    192.168.1.1

Non-authoritative answer:
mit.edu nameserver = use5.akam.net
mit.edu nameserver = use2.akam.net
mit.edu nameserver = asia2.akam.net
mit.edu nameserver = ns1-173.akam.net
mit.edu nameserver = asia1.akam.net
mit.edu nameserver = usw2.akam.net
mit.edu nameserver = eur5.akam.net
mit.edu nameserver = ns1-37.akam.net

asia1.akam.net internet address = 95.100.175.64

C:\>nslookup -type=mx mit.edu
Server:      superfast
Address:    192.168.1.1

Non-authoritative answer:
mit.edu MX preference = 100, mail exchanger = mit-edu.mail.protection.outlook.com

mit.edu nameserver = ns1-173.akam.net
mit.edu nameserver = ns1-37.akam.net
mit.edu nameserver = eur5.akam.net
mit.edu nameserver = usw2.akam.net
mit.edu nameserver = asia1.akam.net
mit.edu nameserver = use5.akam.net
mit.edu nameserver = asia2.akam.net
mit.edu nameserver = use2.akam.net
asia1.akam.net internet address = 95.100.175.64

C:\>

```

*Figure 10-5 Command Line nslookup commands*

Figure 10-5 shows the results of three independent nslookup commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of Polytechnic University in Brooklyn, where the default local DNS server is dns-prime.poly.edu. When running nslookup, if no DNS server is specified, then nslookup sends the query to the default DNS server, which in this case is dns-prime.poly.edu. Consider the first command:

```
nslookup www.mit.edu
```

In words, this command is saying “Please send me the IP address for the host www.mit.edu.” As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of www.mit.edu. Although the response came from the local DNS server at Polytechnic University, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.5 of the textbook.

Now consider the second command:

```
nslookup -type=NS mit.edu
```

In this example, we have provided the option “-type=NS” and the domain “mit.edu”. This causes nslookup to send a query for a type-NS record to the default local DNS server. In words, the query is saying, “Please send me the host names of the authoritative DNS for mit.edu.” (When the -type option is not used, nslookup uses the default, which is to query

### ▪ What is the option MX in the third command?

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of nslookup commands. The syntax is:

```
nslookup -option1 -option2 host-to-find dns-server
```

In general, nslookup can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

Now that we have provided an overview of nslookup, it is time for you to test drive it yourself. Do the following (and write down the results):

1. Run nslookup to obtain the IP address of a Web server in Asia.
2. Run nslookup to determine the authoritative DNS servers for a university in Europe.
3. Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail.

### 4.2.2. Ipconfig

ipconfig (for Windows) and ifconfig (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues. Here we will only describe ipconfig, although the Linux/Unix ifconfig is very similar. ipconfig can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you want to see all this information about your host, simply enter:

```
ipconfig /all
```

into the Command Prompt, as shown in Figure 10-6.

```

Command Prompt
C:\>ipconfig /all

Windows IP Configuration

Host Name . . . . . : DESKTOP-1NQV10C
Primary Dns Suffix . . . . . :
Node Type . . . . . : Mixed
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : mynet

Wireless LAN adapter Local Area Connection* 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #8
Physical Address. . . . . : B2-52-16-41-B1-CF
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Wireless LAN adapter Local Area Connection* 4:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #9
Physical Address. . . . . : C2-52-16-41-B1-CF
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Ethernet adapter Ethernet 4:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . :
Description . . . . . : TAP-Windows Adapter V9
Physical Address. . . . . : 00-FF-BA-6D-1E-F2
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Ethernet adapter Ethernet 5:

Connection-specific DNS Suffix . . . . . : mynet
Description . . . . . : ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter
Physical Address. . . . . : 00-0A-CD-31-77-CB
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::c031:a69:28a0:7cae%18(Preferred)
IPv4 Address. . . . . : 192.168.1.231(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Tuesday, June 2, 2020 8:56:59 PM
Lease Expires . . . . . : Friday, June 5, 2020 8:56:56 PM
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 1342180045
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-28-EF-64-80-52-16-41-B1-CF
DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled
Connection-specific DNS Suffix . . . . . : mynet
  
```

Figure 10-6 ipconfig /all command

`ipconfig` is also very useful for managing the DNS information stored in your host. A host could cache DNS records it recently obtained. To see these cached records, after the prompt `C:\>` provide the following command:

```
ipconfig /displaydns
```

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter

```
ipconfig /flushdns
```

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.