CSE-331 Computer Networks

CSE-331L Computer Networks

Course Instructor: Mr. Amjad Majeed

University of Engineering & Technology Lahore.
(KSK Campus)

CSE-331 Computer Networks

Lab No. 01 CLO1

Objective:

1. Study of following Network Devices in Detail
    - Repeater
    - Hub
    - Switch
    - Bridge
    - Router
    - Gate Way
2. Study of different types of Network cables and Practically implement the cross-wired cable and straight through cable using clamping tool.

NETWORK DEVICES:

i- **Repeater:** Functioning at Physical Layer. A repeater is an electronic device that receives a signal and retransmits it at a higher level and/or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances. Repeater have two ports, so cannot be used to connect for more than two devices

ii- **Hub:** An Ethernet hub, active hub, network hub, repeater hub, hub or concentrator is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. The device is a form of multiport repeater. Repeater hubs also participate in collision detection, forwarding a jam signal to all ports if it detects a collision.

iii- **Switch:** A network switch or switching hub is a computer networking device that connects network segments. The term commonly refers to a network bridge that processes and routes data at the data link layer (layer 2) of the OSI model. Switches that additionally process data at the network layer (layer 3 and above) are often referred to as Layer 3 switches or multilayer switches.

iv- **Bridge:** A network bridge connects multiple network segments at the data link layer (Layer 2) of the OSI model. In Ethernet networks, the term bridge formally means a device that behaves according to the IEEE 802.1D standard. A bridge and switch are very much alike; a switch being a bridge with numerous ports. Switch or Layer 2 switch is often used interchangeably with bridge Bridges can analyze incoming data packets to determine if the bridge is able to send the given packet to another segment of the network.

v- **Router:** A router is an electronic device that interconnects two or more computer networks and selectively interchanges packets of data between them. Each data packet contains address information that a router can use to determine if the source and destination are on the same network, or if the data packet must be transferred from one network to another. Where multiple routers are used in a large collection of interconnected networks, the routers exchange information about target system addresses, so that each router can build up a table showing the preferred paths between any two systems on the interconnected networks.

vi- **Gate Way:** In a communications network, a network node equipped for interfacing with another network that uses different protocols.

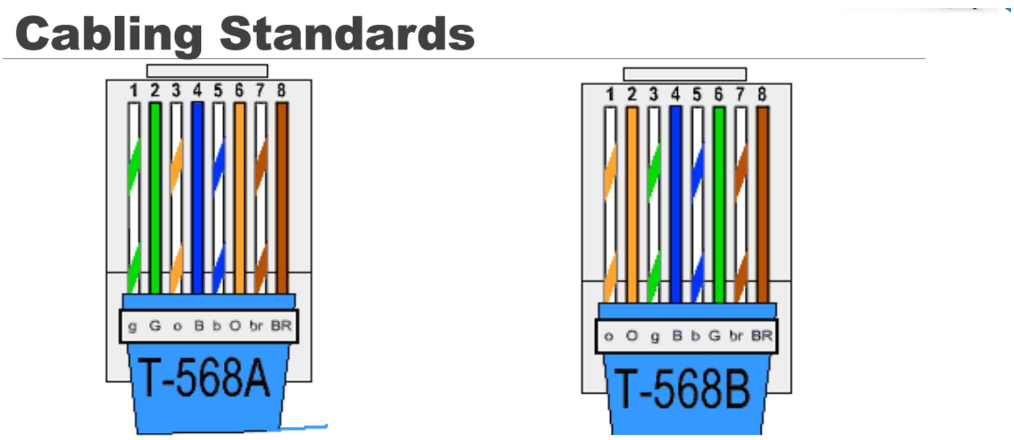
- A gateway may contain devices such as protocol translators, impedance matching devices, rate converters, fault isolators, or signal translators as necessary to provide system interoperability. It also requires the establishment of mutually acceptable administrative procedures between both networks.
- A protocol translation/mapping gateway interconnects networks with different network protocol technologies by performing the required protocol conversions

CODING STANDARDS

What is 568?

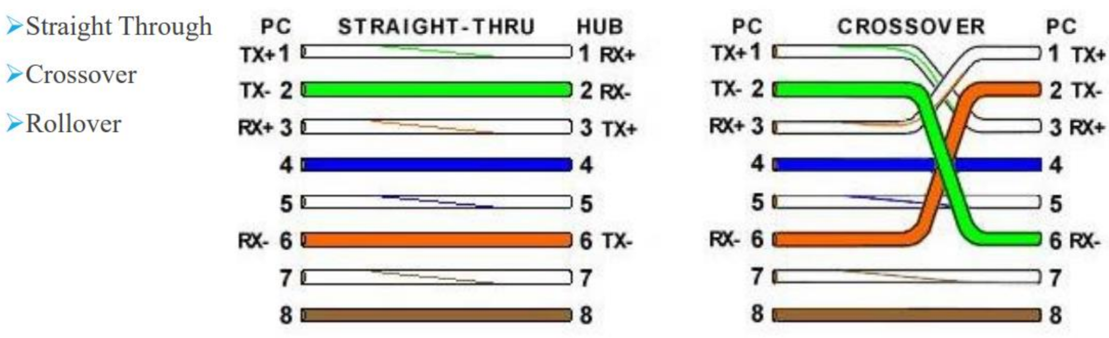
ANSI/TIA-568 is a set of telecommunication standards from the Telecommunication Industry Association (TIA). The standard address commercial building cabling for telecommunication products and services.

TIA defines two different separate pin outs 568A and 568B for any eight conductor twisted pair cabling. 568A is recommended for most cabling systems and 568B as an alternative to accommodate certain cable systems.

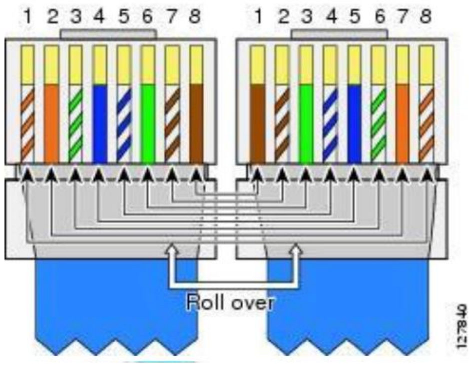


Cabling Types:

There are three types of cables:



- Rollover Cable



Straight Through	Cross-over	Rollover
1 ↔ 1	1 1	1 ↔ 8
2 ↔ 2	2 2	2 ↔ 7
3 ↔ 3	3 3	3 ↔ 6
4 ↔ 4	4 ↔ 4	4 ↔ 5
5 ↔ 5	5 5	5 ↔ 4
6 ↔ 6	6 6	6 ↔ 3
7 ↔ 7	7 ↔ 7	7 ↔ 2
8 ↔ 8	8 ↔ 8	8 ↔ 1

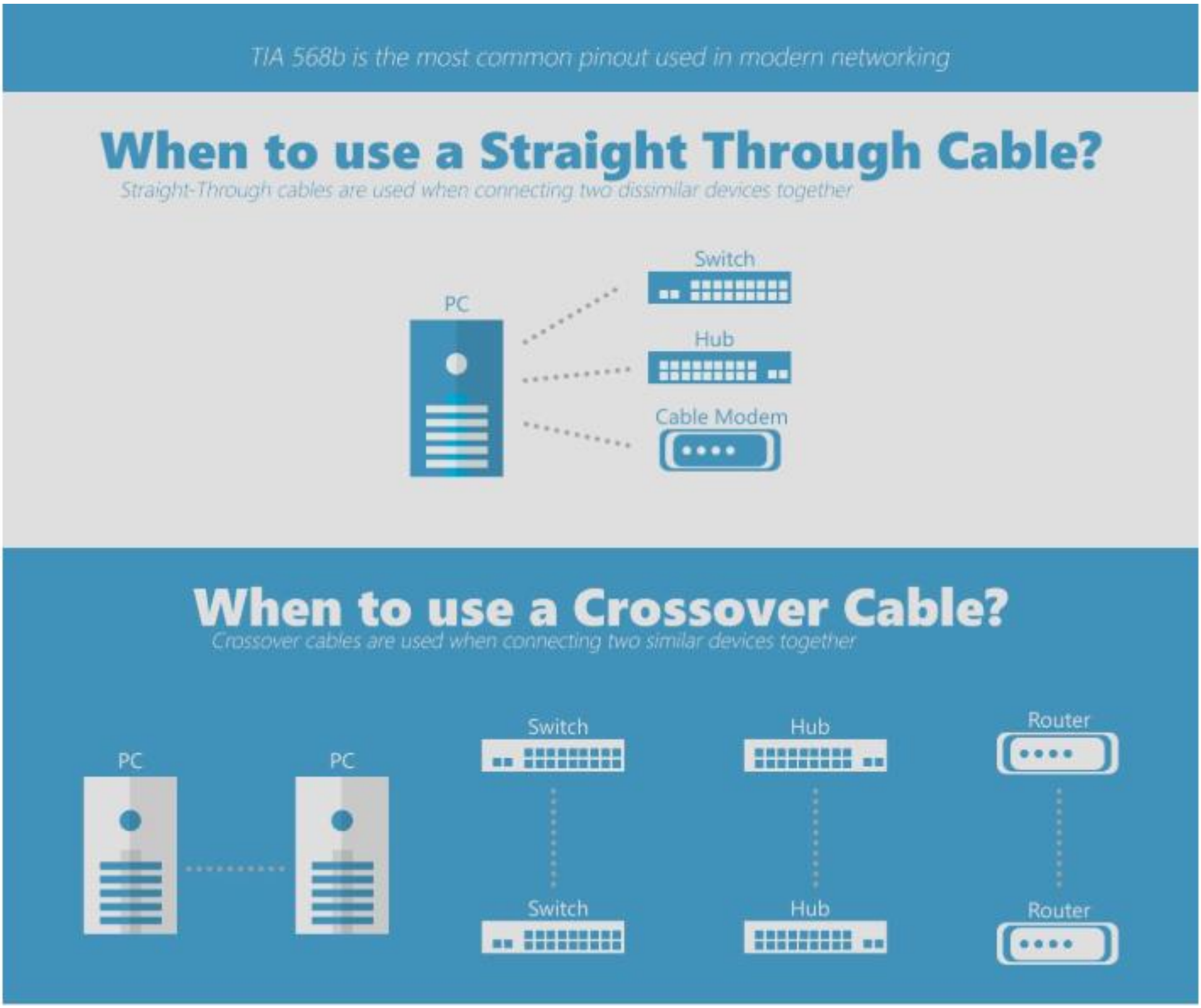
Crimping Tool

A crimping tool is a device used to conjoin two pieces of metal by deforming one or both of them in a way that causes them to hold each other. The result of the tool's work is called a crimp. A good example of crimping is the process of affixing a connector to the end of a cable.

Cable Tester:

CSE-331 Computer Networks

Cable Tester



Course Instructor: Mr. Amjad Majeed **Graduate Assistant: Engr Farwa Shah**

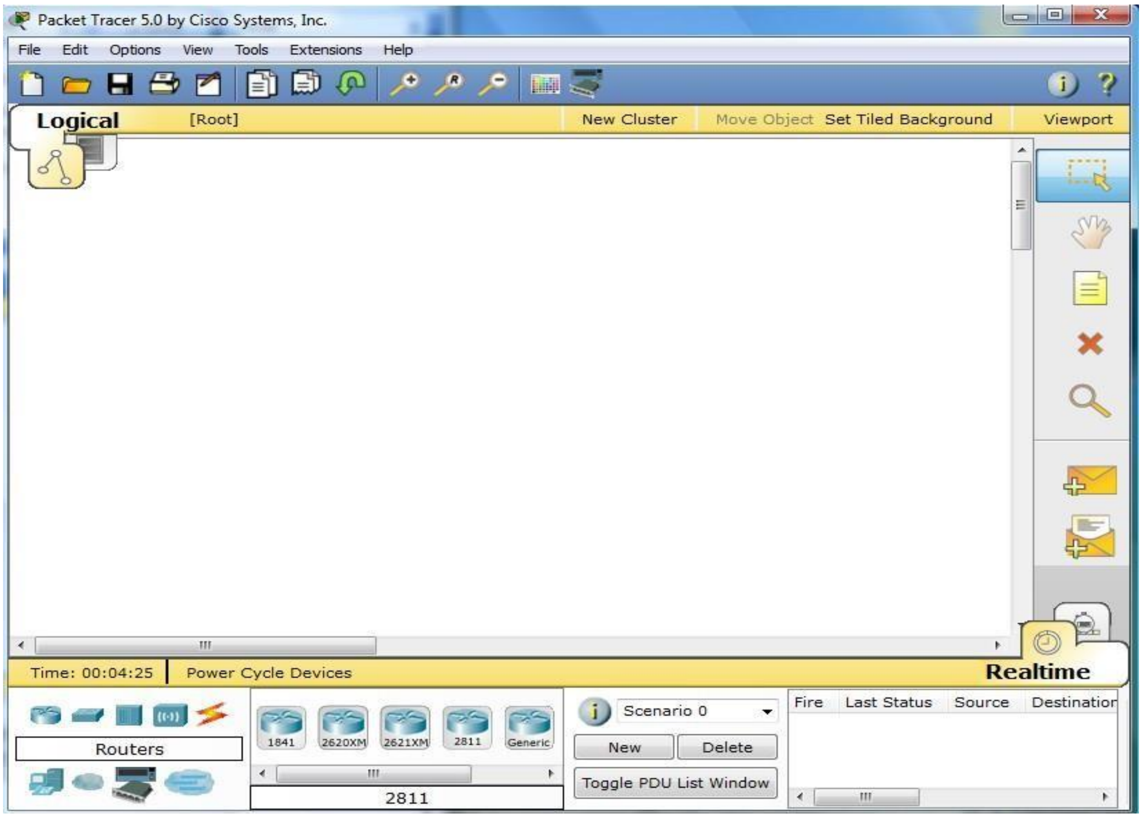
Lab No. 02 CLO2

Aim: Creating a Network topology using CISCO packet tracer software.

Packet Tracer – Creating a New Topology

What is Packet Tracer? Packet Tracer is a protocol simulator developed by Dennis Frezzo and his team at Cisco Systems. Packet Tracer (PT) is a powerful and dynamic tool that displays the various protocols used in networking, in either Real Time or Simulation mode. This includes layer 2 protocols such as Ethernet and PPP, layer 3 protocols such as IP, ICMP, and ARP, and layer 4 protocols such as TCP and UDP. Routing protocols can also be traced.

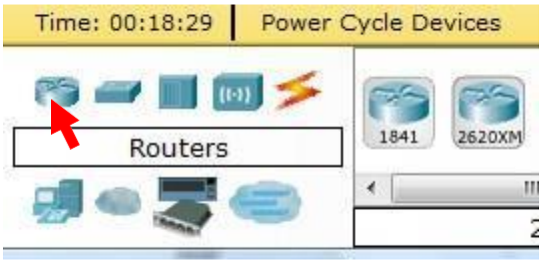
Step 1: Start Packet Tracer

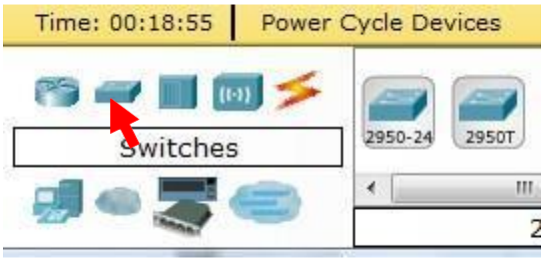


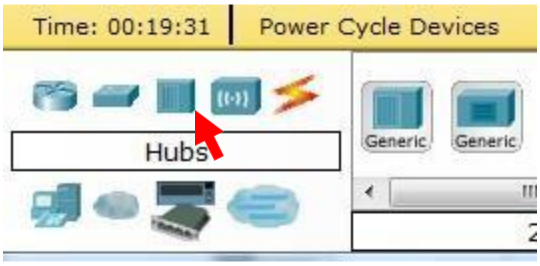
Step 2: Choosing Devices and Connections

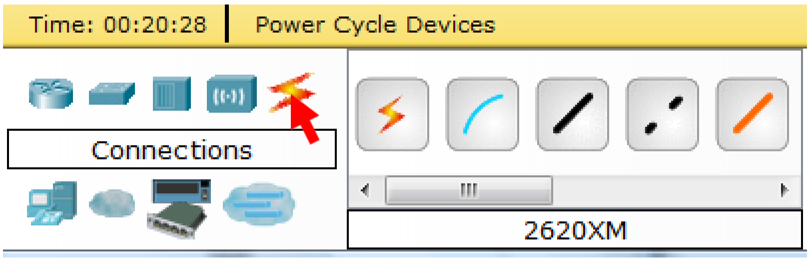
We will begin building our network topology by selecting devices and the media in which to connect them. Several types of devices and network connections can be used. For this lab we will keep it simple by using **End Devices**, **Switches**, **Hubs**, and **Connections**.

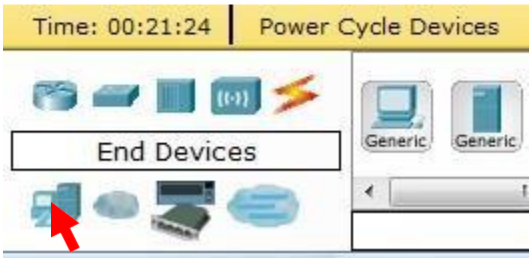
Single click on each group of devices and connections to display the various choices. The devices you see may differ slightly.

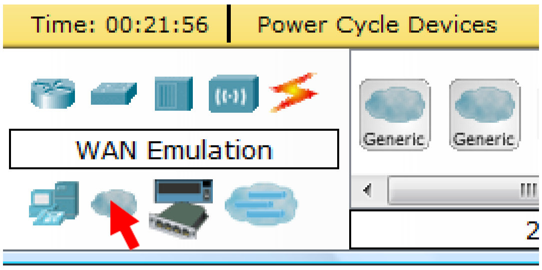






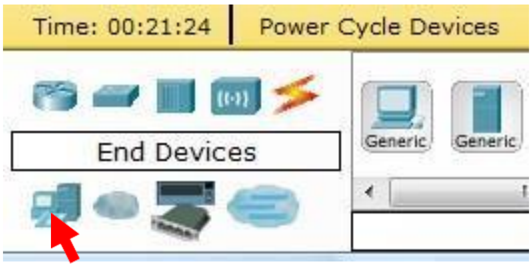




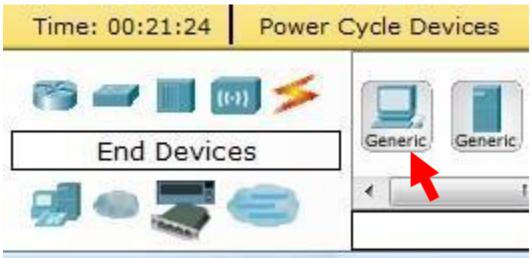


Step 3: Building the Topology – Adding Hosts

Single click on the **End Devices**.



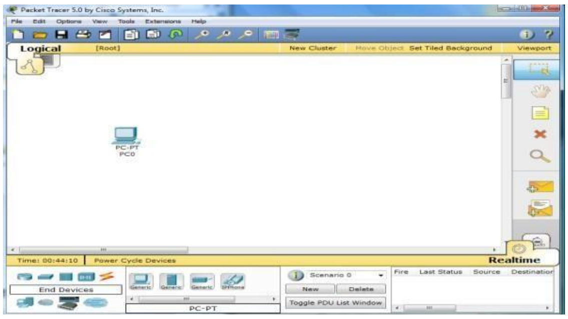
Single click on the **Generic** host.



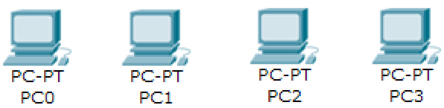
Move the cursor into topology area. You will notice it turns into a plus "+" sign.

+

Single click in the topology area and it copies the device.



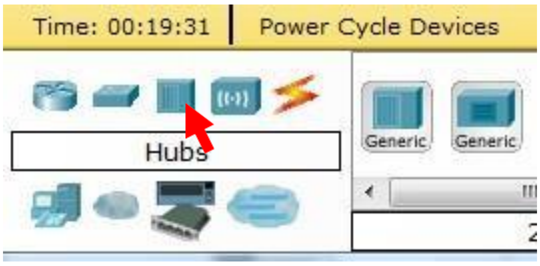
Add three more hosts.

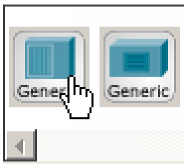


Step 4: Building the Topology – Connecting the Hosts to Hubs and Switches

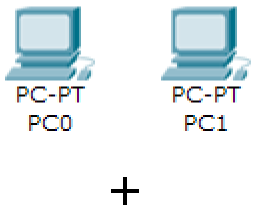
Adding a Hub

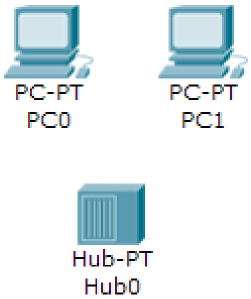
Select a hub, by clicking once on **Hubs** and once on a **Generic** hub.



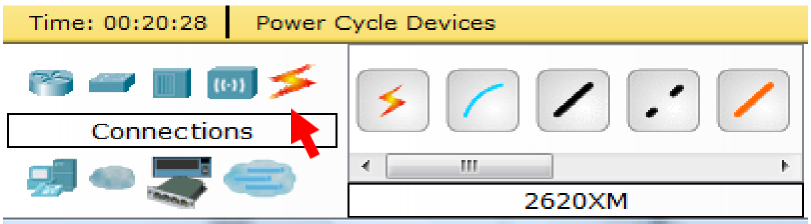


Add the hub by moving the plus sign "+" below PC0 and PC1 and click once.

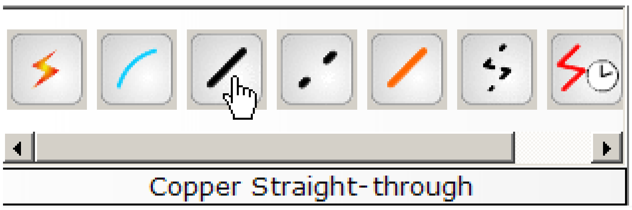




Connect PC0 to Hub0 by first choosing **Connections.**



Click once on the **Copper Straight-through** cable.

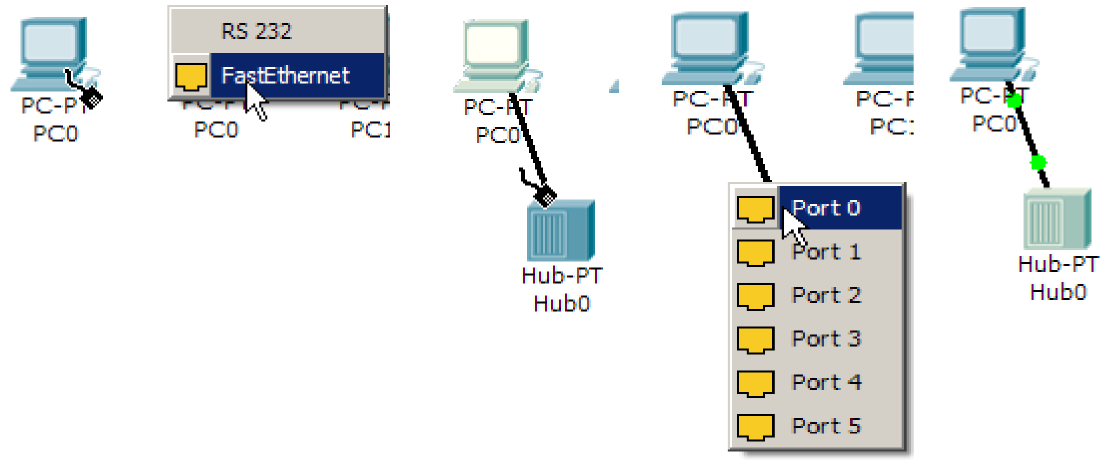


Perform the following steps to connect **PC0** to **Hub0**:

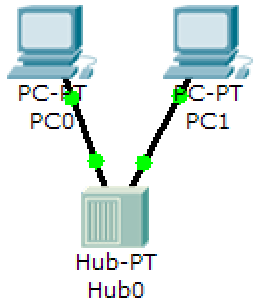
1. Click once on **PC0**
2. Choose **FastEthernet**
3. Drag the cursor to **Hub0**
4. Click once on **Hub0** and choose **Port 0**

5. Notice the green link lights on both the **PC0** Ethernet NIC and the **Hub0** Port 0 showing that the link is active.

1 **2** **3** **4** **5**



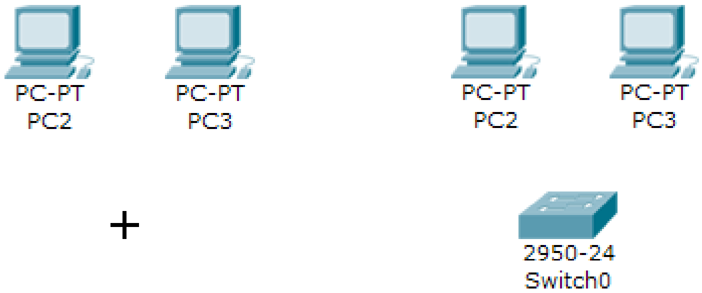
Repeat the steps above for **PC1** connecting it to **Port 1** on **Hub0**. (The actual hub port you choose does not matter.)



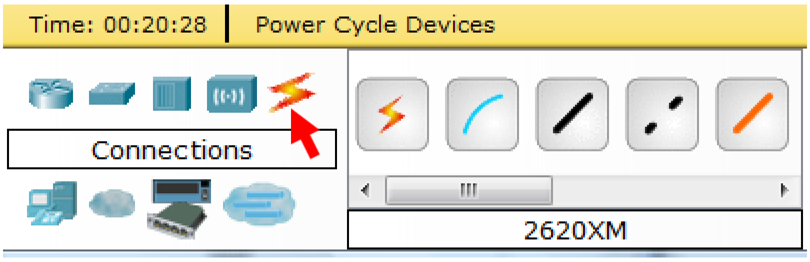
Adding a Switch

Select a switch, by clicking once on **Switches** and once on a **2950-24** switch.

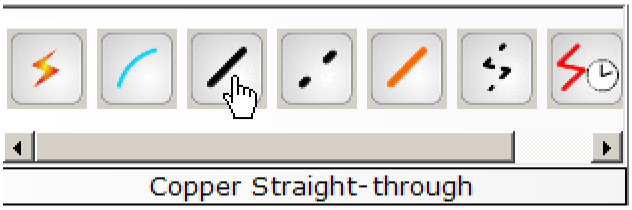
Add the switch by moving the plus sign “+” below PC2 and PC3 and click once.



Connect PC2 to Hub0 by first choosing **Connections.**



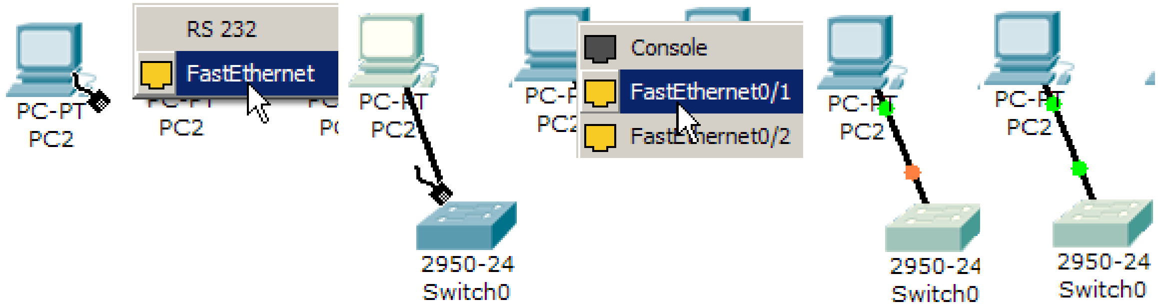
Click once on the **Copper Straight-through** cable.



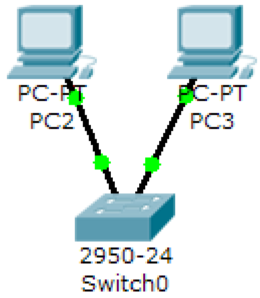
Perform the following steps to connect **PC2** to **Switch0**:

1. Click once on **PC2**
2. Choose **Fast Ethernet**
3. Drag the cursor to **Switch0**
4. Click once on **Switch0** and choose **FastEthernet0/1**
5. Notice the green link lights on **PC2** Ethernet NIC and amber light **Switch0 FastEthernet0/1 port**. The switch port is temporarily not forwarding frames, while it goes through the stages for the Spanning Tree Protocol (STP) process.
6. After a about 30 seconds the amber light will change to green indicating that the port has entered the forwarding stage. Frames can now forward out the switch port.

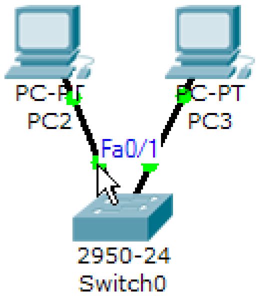
1 2 3 4 5 6



Repeat the steps above for **PC3** connecting it to **Port 3** on **Switch0** on port **FastEtherent0/2**. (The actual switch port you choose does not matter.)



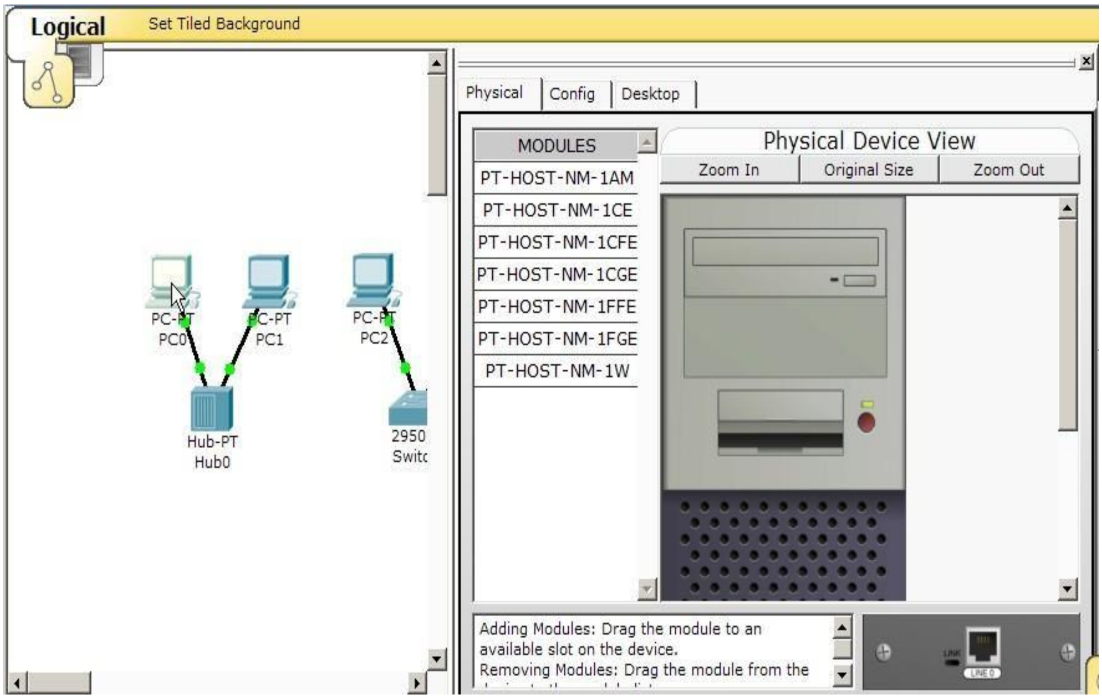
Move the cursor over the link light to view the port number. **Fa** means FastEthernet, 100 Mbps Ethernet.



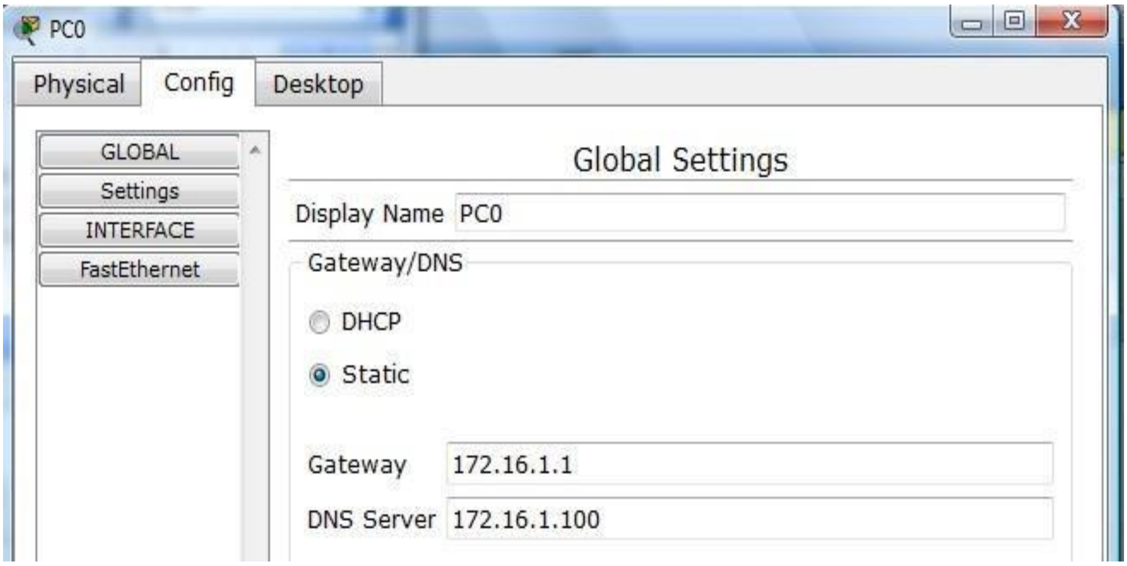
Step 5: Configuring IP Addresses and Subnet Masks on the Hosts

Before we can communicate between the hosts we need to configure IP Addresses and Subnet Masks on the devices.

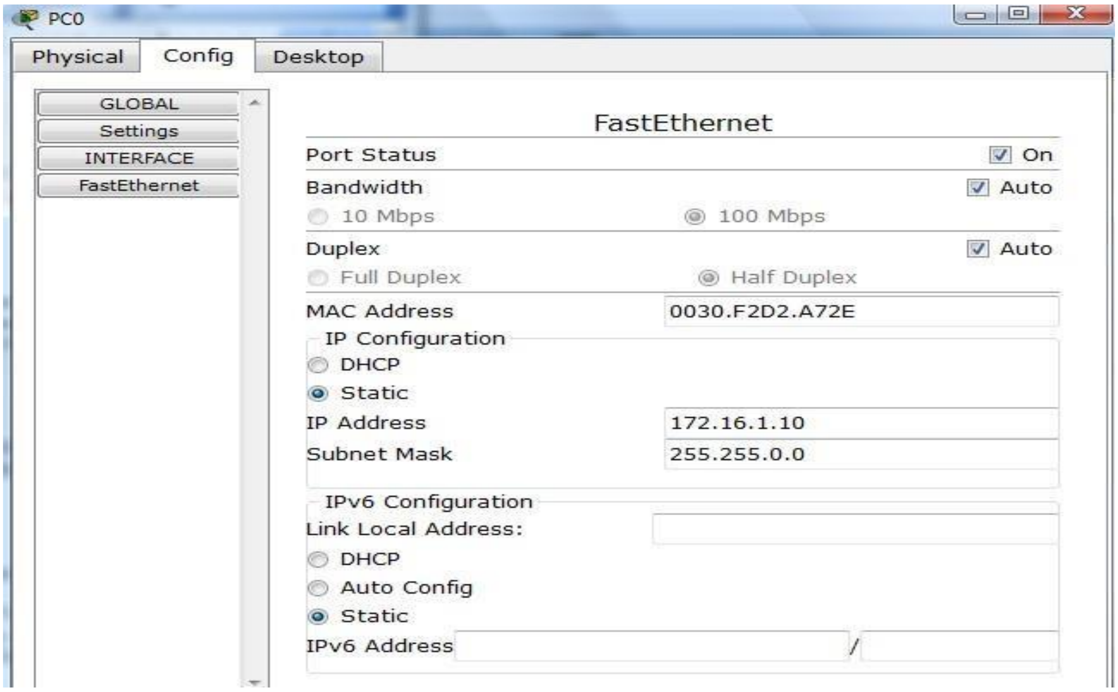
Click once on PC0.



Choose the **Config** tab and click on **Settings**. It is here that you can change the name of PC0. It is also here where you would enter a **Gateway** IP Address, also known as the default gateway and the **DNS Server** IP Address. We will discuss this later, but this would be the IP address of the local router. If you want, you can enter the Gateway IP Address 172.16.1.1 and DNS Server IP Address 172.16.1.100, although it will not be used in this lab.



Click on **Interface** and then **Fast Ethernet**. Although we have not yet discussed IP Addresses, add the IP Address to 172.16.1.10. Click once in the Subnet Mask field to enter the default Subnet Mask. You can leave this at 255.255.0.0. We will discuss this later.



Also, notice this is where you can change the Bandwidth (speed) and Duplex of the Ethernet NIC (Network Interface Card). The default is Auto (auto negotiation), which means the NIC will negotiate with the hub or switch. The bandwidth and/or duplex can be manually set by removing the check from the **Auto** box and choosing the specific option.

Bandwidth - Auto

If the host is connected to a hub or switch port which can do 100 Mbps, then the Ethernet NIC on the host will choose 100 Mbps (Fast Ethernet). Otherwise, if the hub or switch port can only do 10 Mbps, then the Ethernet NIC on the host will choose 10 Mbps (Ethernet).

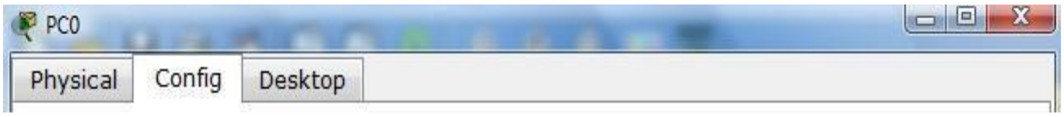
Duplex - Auto

Hub: If the host is connected to a hub, then the Ethernet NIC on the host will choose Half Duplex.

Switch: If the host is connected to a switch, and the switch port is configured as Full Duplex (or Auto negotiation), then the Ethernet NIC on the host will choose Full Duplex. If the switch port is configured as Half Duplex, then the Ethernet NIC on the host will choose Half Duplex. (Full Duplex is a much more efficient option.)

The information is automatically saved when entered.

To close this dialog box, click the "**X**" in the upper right.

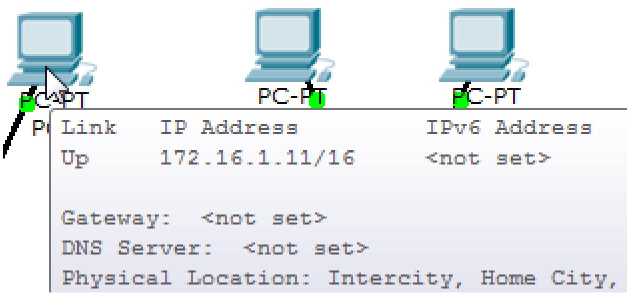


Repeat these steps for the other hosts. Use the information below for IP Addresses and Subnet Masks.

Host	**IP Address**	**Subnet Mask**
PC0	172.16.1.10	255.255.0.0
PC1	172.16.1.11	255.255.0.0
PC2	172.16.1.12	255.255.0.0
PC3	172.16.1.13	255.255.0.0

Verify the information

To verify the information that you entered, move the Select tool (arrow) over each host.

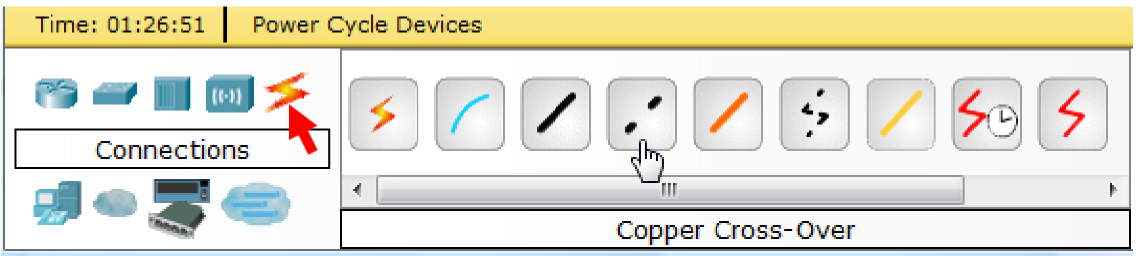


Deleting a Device or Link

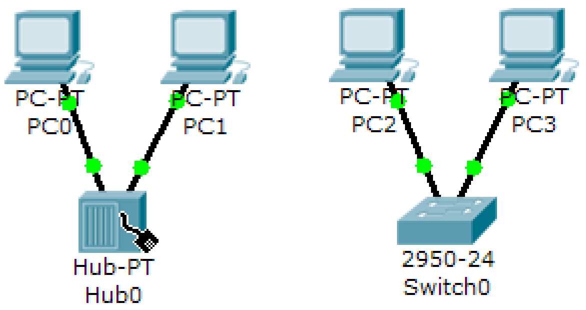
To delete a device or link, choose the **Delete** tool and click on the item you wish to delete.

Step 6: Connecting Hub0 to Switch0

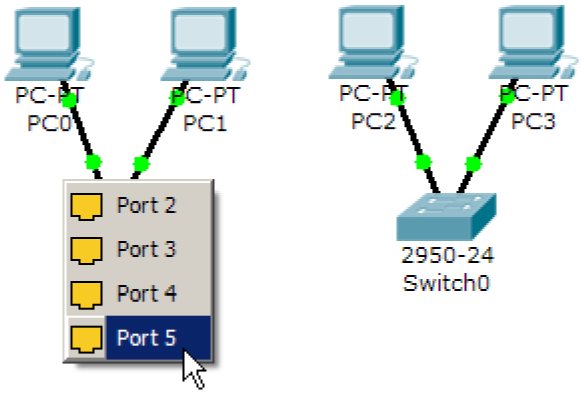
To connect like-devices, like a Hub and a Switch, we will use a Cross-over cable. Click once the **Cross-over** Cable from the **Connections** options.



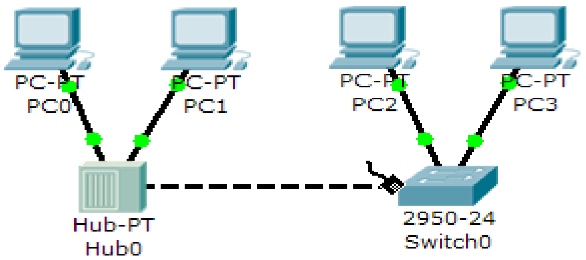
Move the Connections cursor over **Hub0** and click once.



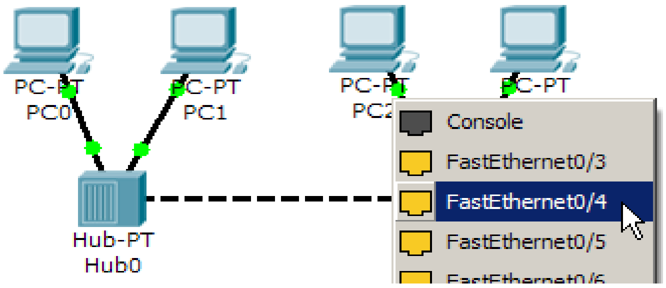
Select **Port 5** (actual port does not matter).



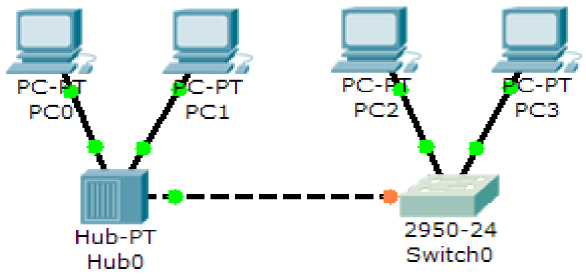
Move the Connections cursor to **Switch0**.

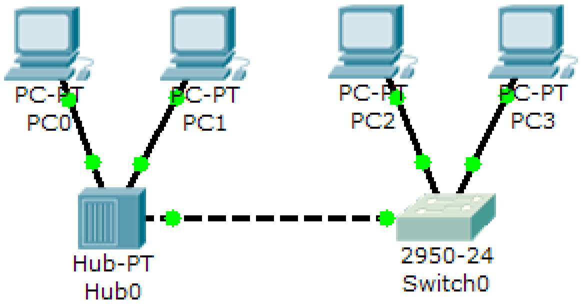


Click once on **Switch0** and choose **FastEthernet0/4** (actual port does not matter).



The link light for switch port **FastEthernet0/4** will begin as amber and eventually change to green as the Spanning Tree Protocol transitions the port to forwarding.

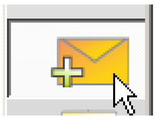




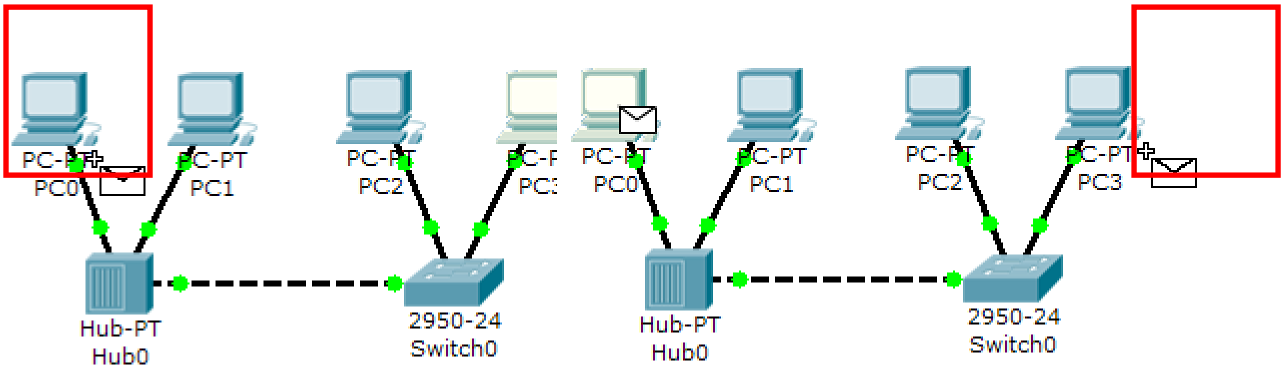
Step 7: Verifying Connectivity in Real-time Mode

Be sure you are in **Real-time** mode.

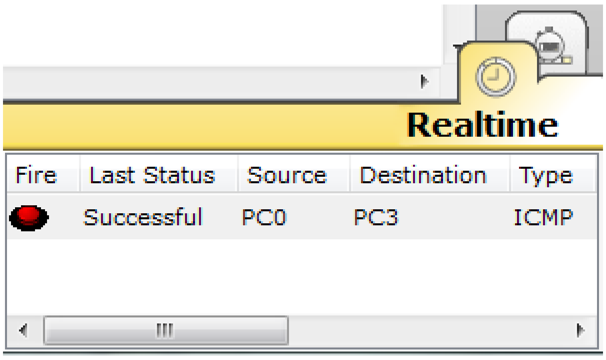
Select the **Add Simple PDU** tool used to ping devices.



Click once on PC0, then once on PC3.



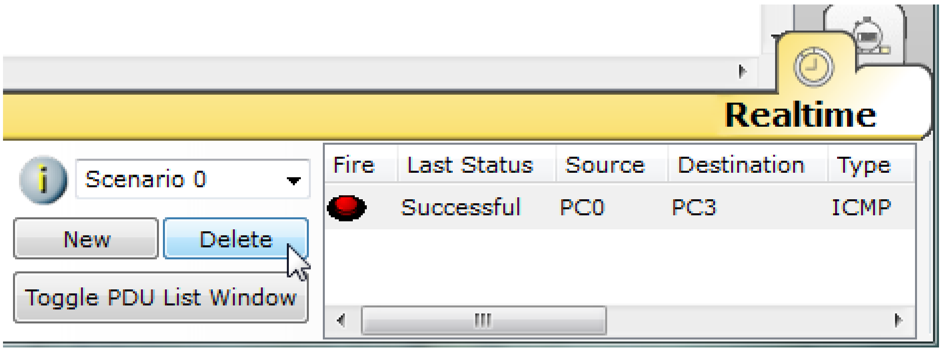
The PDU **Last Status** should show as **Successful**.



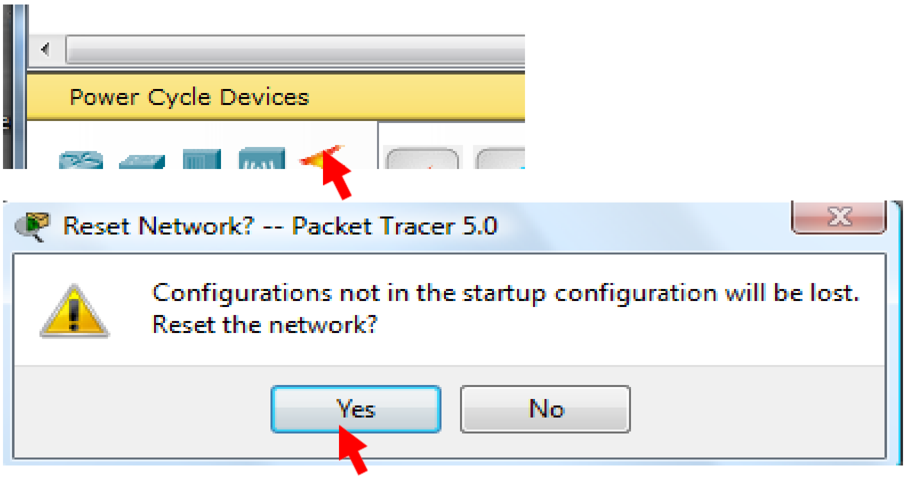
Retting the Network

At this point we will want to reset the network, whenever you want to reset the network and begin the simulation again, perform the following tasks:

Click **Delete** in the PDU area.

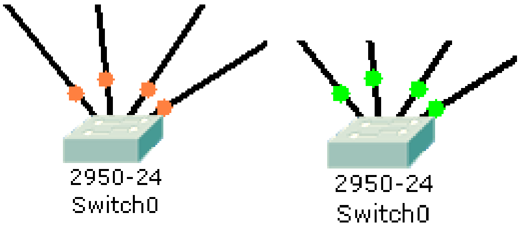


Now, Power Cycle Devices and confirm the action.



Waiting for Spanning Tree Protocol (STP)

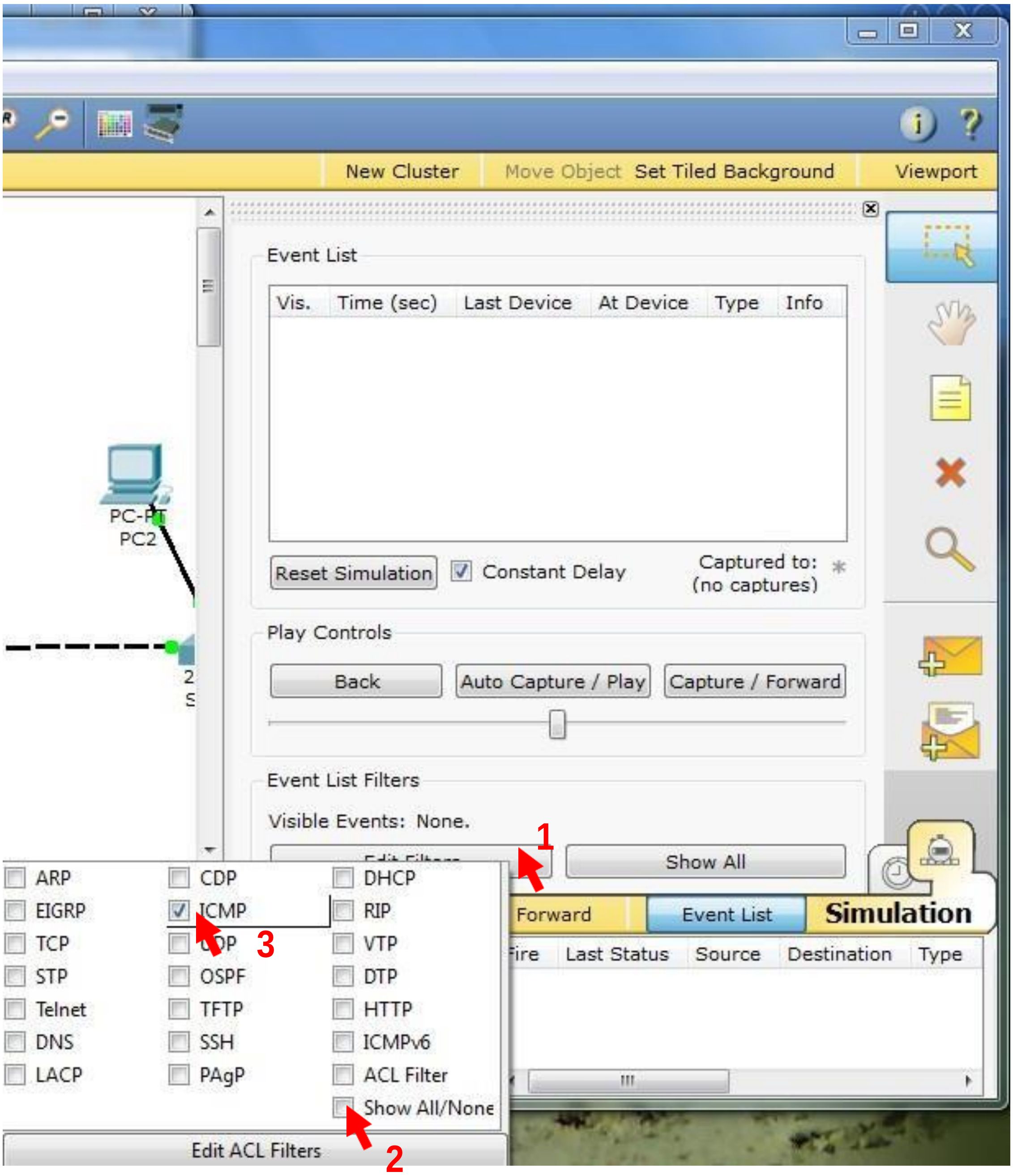
Note: Because Packet Tracer also simulates the Spanning Tree Protocol (later), at times the switch may show amber lights on its interfaces. You will need to wait for the lights to turn green on the switches before they will forward any Ethernet frames.



Step 8: Verifying Connectivity in Simulation Mode

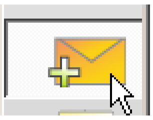
Be sure you are in **Simulation** mode.

Deselect all filters (All/None) and select only **ICMP**.

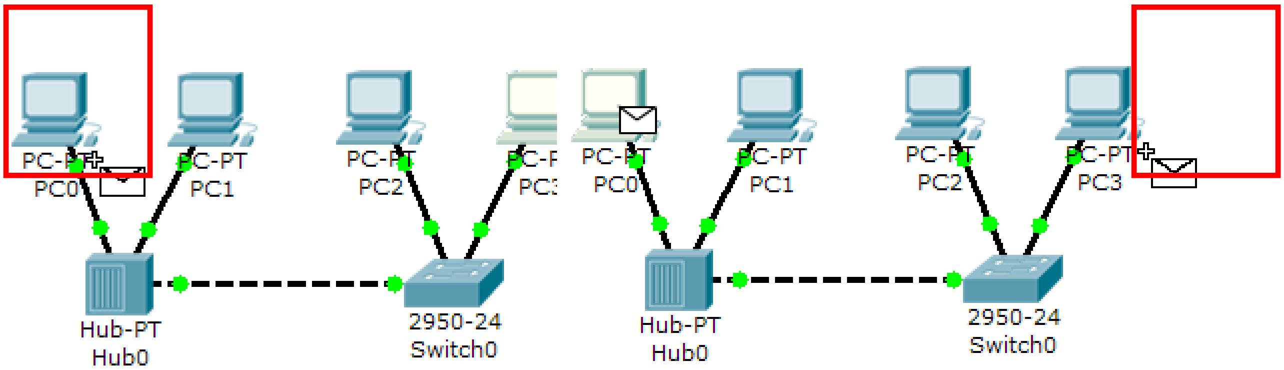


New Cluster
Move Object
Set Tiled Background
Viewport
Event List
Vis.
Time (sec)
Last Device
At Device
Type
Info
PC-PT
PC2
Reset Simulation
Constant Delay
Captured to:
(no captures)
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters
Visible Events: None.
Show All
1
ARP
CDP
DHCP
EIGRP
ICMP
RIP
TCP
VTP
STP
OSPF
DTP
Telnet
TFTP
HTTP
DNS
SSH
ICMPv6
LACP
PAgP
ACL Filter
Show All/None
Edit ACL Filters
2
3
Forward
Event List
Simulation
Last Status
Source
Destination
Type

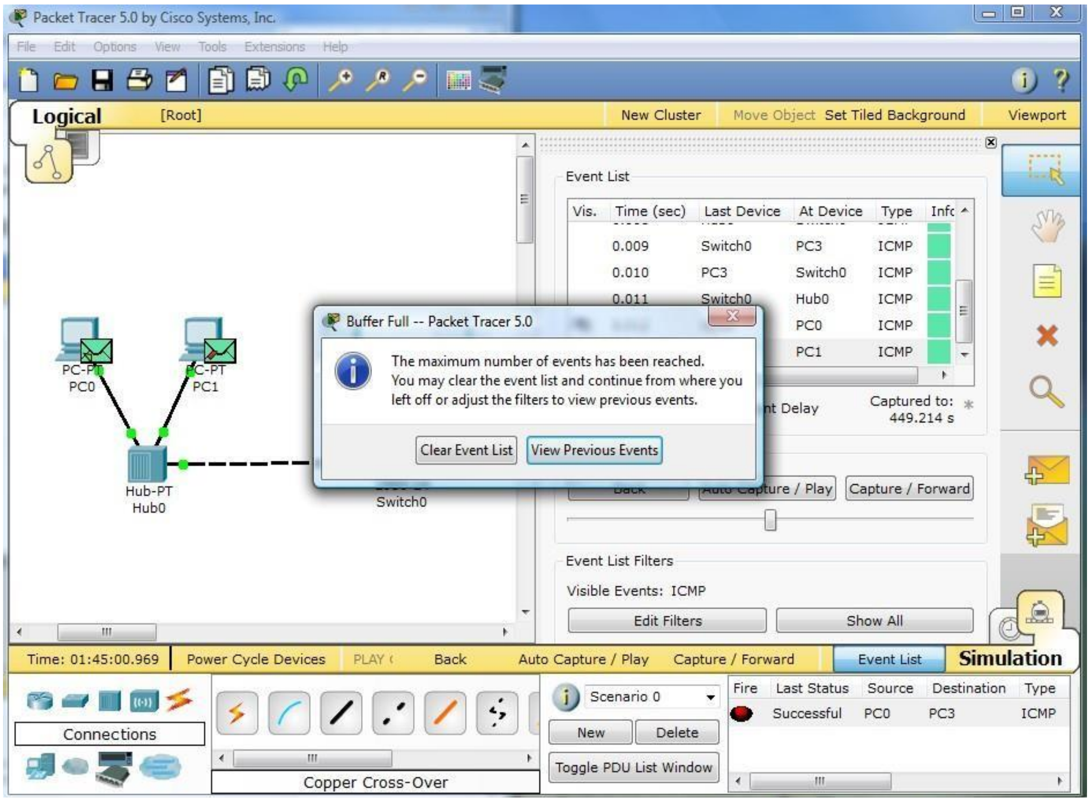
Select the **Add Simple PDU** tool used to ping devices..



Click once on PC0, then once on PC3.

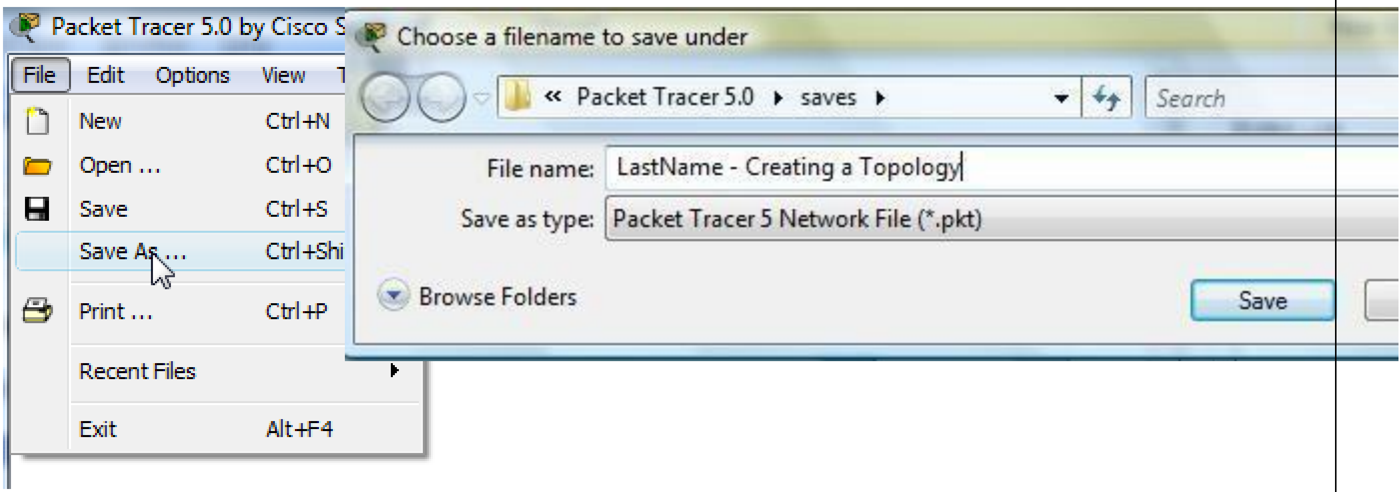


Continue clicking **Capture/Forward** button until the ICMP ping is completed. You should see the ICMP messages move between the hosts, hub and switch. The PDU **Last Status** should show as **Successful**. Click on **Clear Event List** if you do not want to look at the events or click **Preview Previous Events** if you do. For this exercise it does not matter.

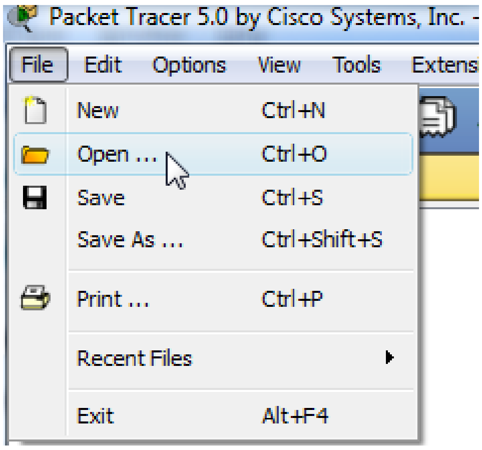


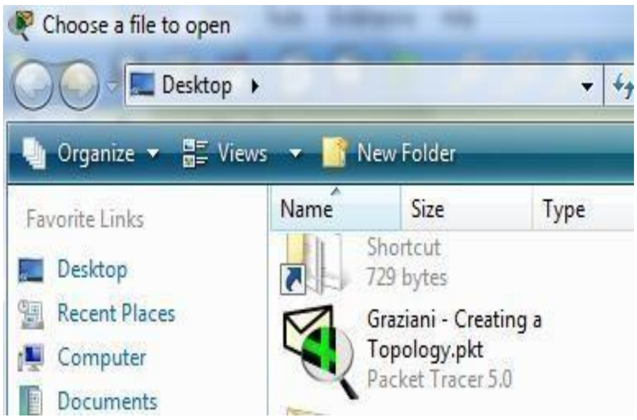
Step 9: Saving the Topology

Perform the following steps to save the topology (uses .pkt file extension).

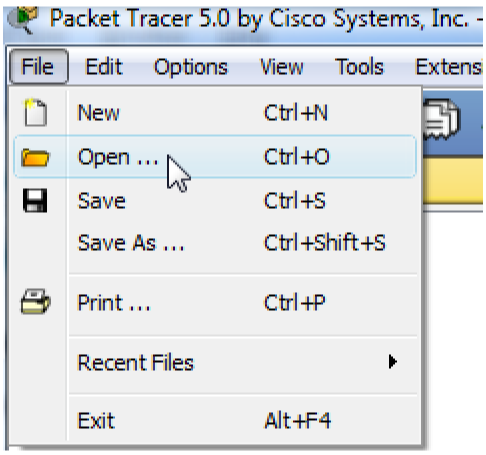


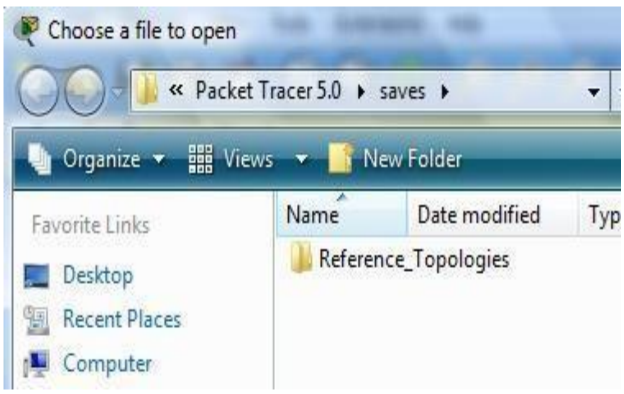
Opening Existing Topologies





Opening Existing PT Topologies





*****

CSE-331 Computer Networks
Lab No. 03 CLO1

Objective:

1. Study of Switch Algorithm and Switch MAC Address Tables

Looking at the Switch Algorithm and Switch MAC Address Tables

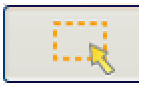
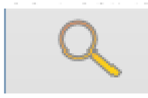
Step 1

Open the UsingSwitch.pkt file. Do not save the changes to the current network. Notice the similarity to the previous topology. The layer 1 hub has been replaced with a layer 2 switch (if not present, create the topology).

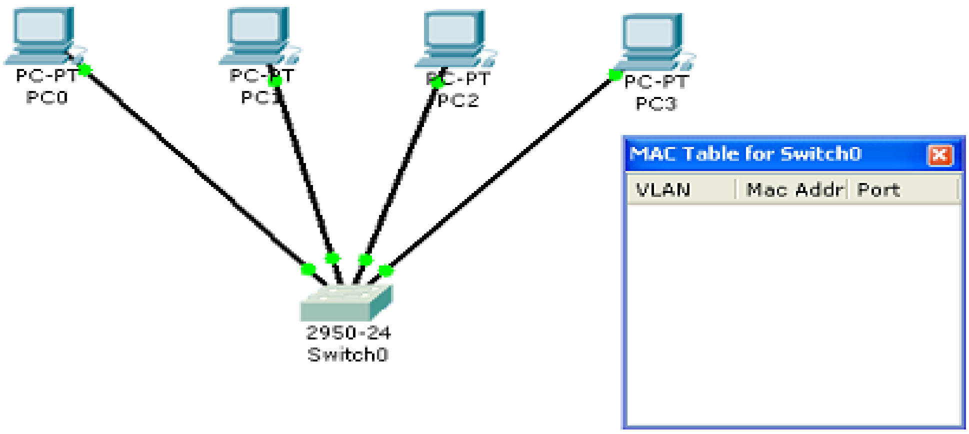
Click on the Simulation icon to switch to simulation mode.

Step 2 Viewing the Switch MAC Address Table

Use the Select tool to view IP address and MAC address information for the various hosts.

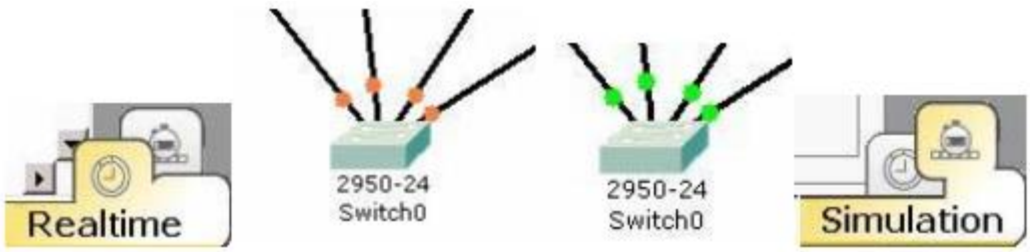
 Use the Inspect tool to view the MAC Address Table of the switch. 

The MAC Address Table is empty as it has not learned any Source Ethernet MAC Addresses. Notice that there is also a VLAN column in this table. This will be discussed in future courses.



Waiting for STP

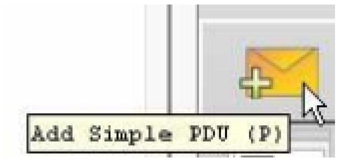
Note: Because of how Packet Tracer deals with the Spanning Tree Protocol, at times the switch may show amber lights on its interfaces. To correct this, click the Real-time mode icon, wait for the lights to turn green, and then click the Simulation mode icon, returning to where you left off.



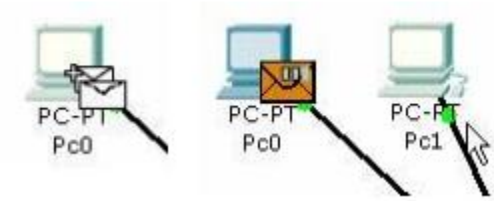
Step 3: Issuing a Ping and Viewing the MAC Address Table

ARP is used to learn the MAC address to use to encapsulate the IP packet in an Ethernet frame. The ARP packet will precede the ICMP packet.

Using the Add Simple PDU perform a ping from PC0 to PC1.
Choose the Add Simple PDU tool from the toolbox:

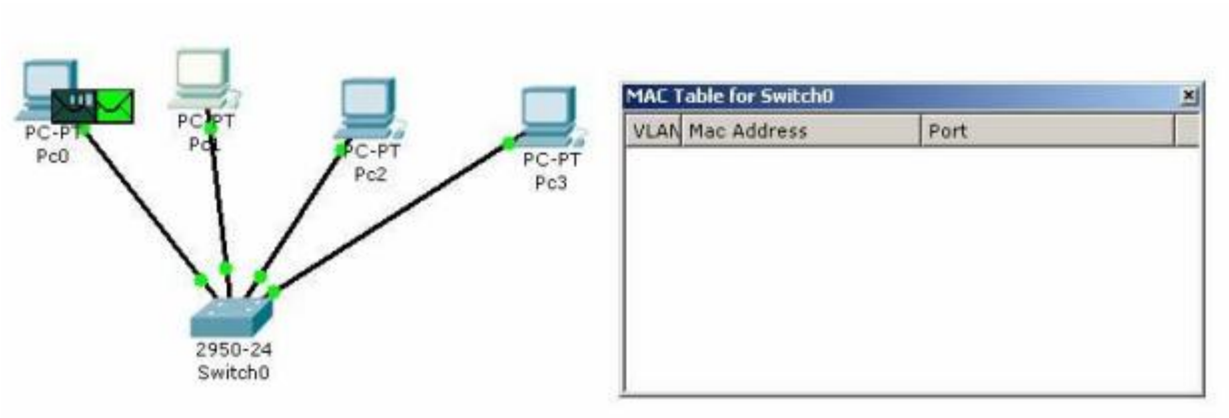


Click once on PC0, the device issuing the ping (ICMP Echo Request) and then click once on PC1 (the destination of the ICMP Echo Request).

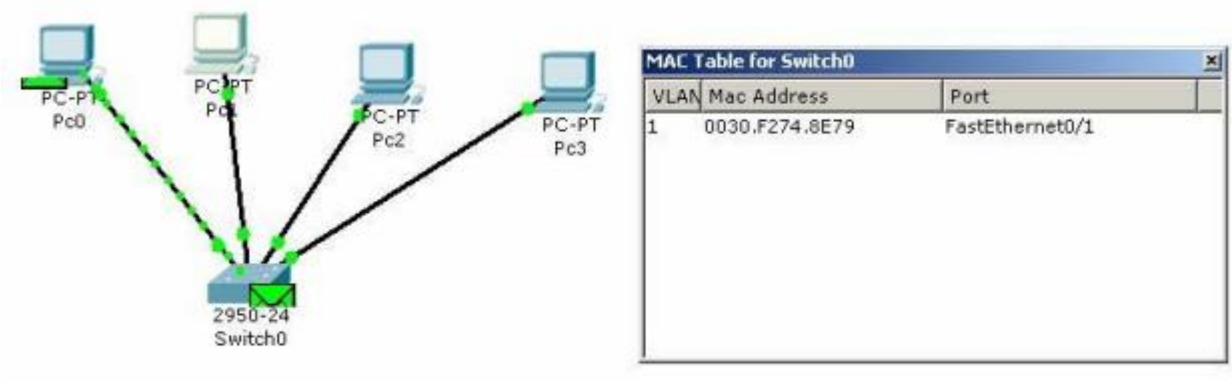


Run the simulation using the Play button.

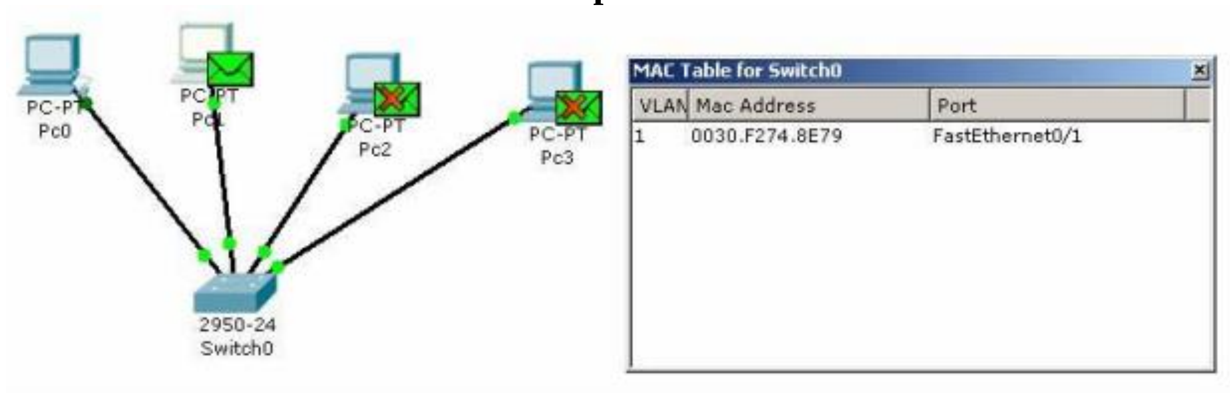
PC0 forwards the frame containing the ARP request to Switch0:



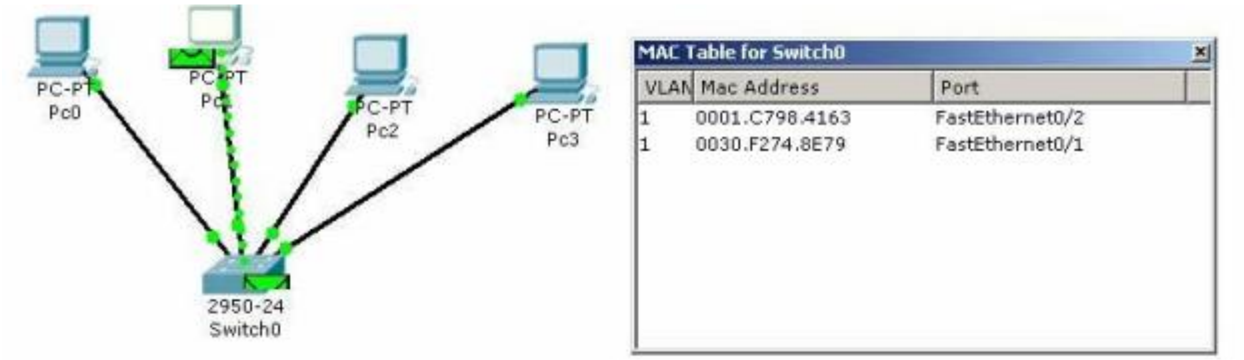
Notice how the Switch learns the Source MAC Address of the frame:



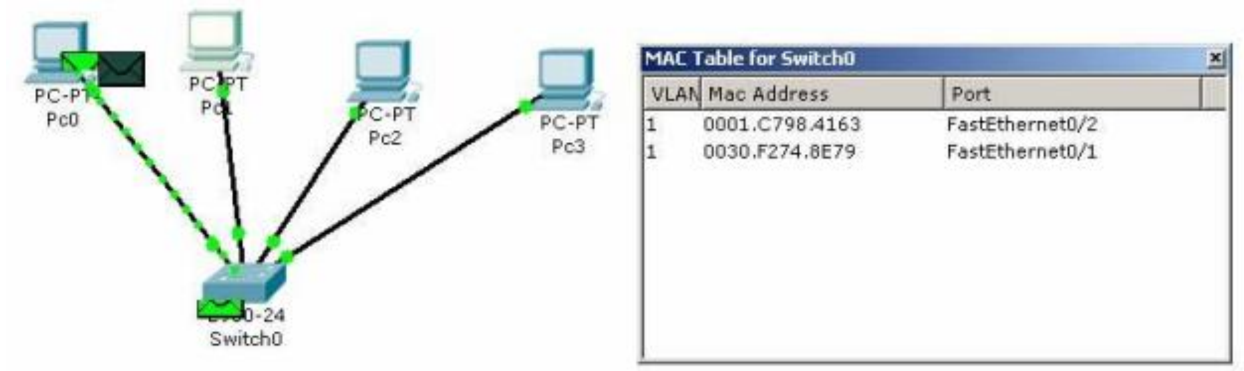
The packet is flooded out all ports because the Switch's MAC Address Table does not contain the Destination Address of the Ethernet frame. PC2 and PC3 disregard the frame:



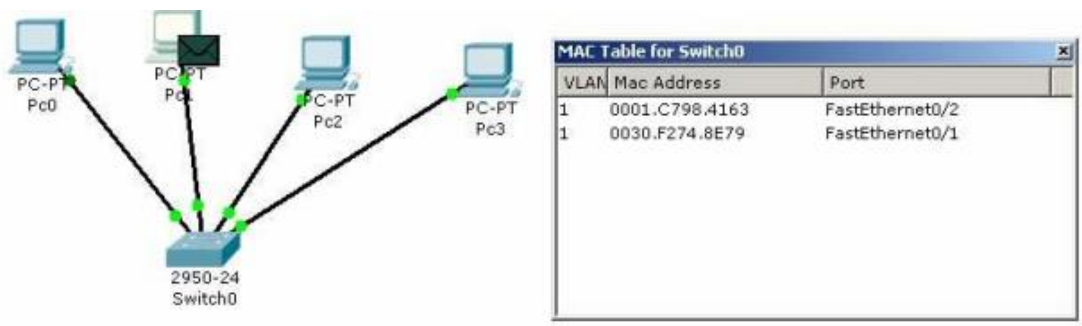
PC1 return the ARP reply. Switch0 learns the Source MAC Address of PC1:



Because the Source MAC Address of PC0 was learned previously, when examining the Destination MAC Address of the frame, Switch0 filters the frame by only sending it out FastEthernet port 0/1



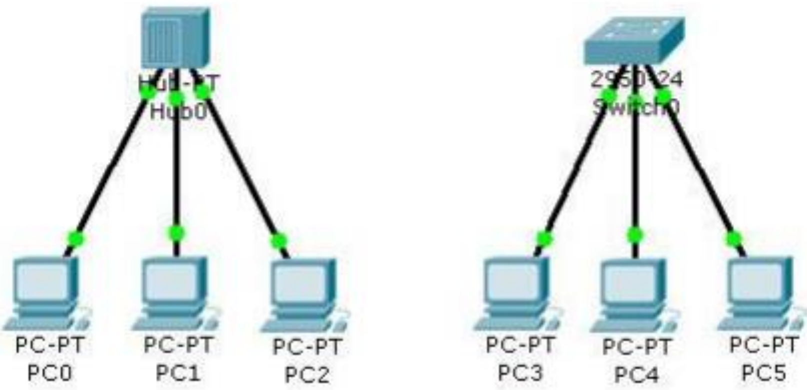
The rest of the pings, frames with IP packets containing ICMP Echo Requests from PC0 destined for PC1 and frames with IP packets containing ICMP Echo Replies from PC1 destined for PC0, are filtered by the switch an only sent out the appropriate interface (port).



CSE-331 Computer Networks

ACTIVITY 3

Build the following network on the same window in packet tracer.



Assign the IP's as:

PC	IP Address	Subnet Mask
PC0	192.168.1.1	255.255.255.0
PC1	192.168.1.2	255.255.255.0
PC2	192.168.1.3	255.255.255.0
PC3	192.168.2.1	255.255.255.0
PC4	192.168.2.2	255.255.255.0
PC5	192.168.2.3	255.255.255.0

If all connections and addressing are correct, the link lights should be green. If not, troubleshoot the cabling type, connections, and addressing.

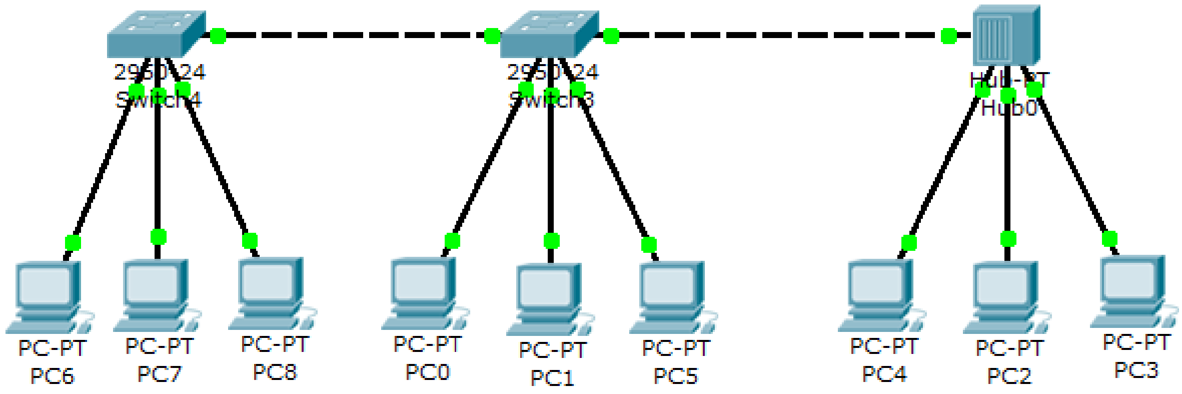
Now do the following:

1) Go to Simulation mode. In the Event List filters, enable only ICMP and ARP. Using the "Simple PDU", issue a ping from PC0 to PC1. Play the simulation. Pay close attention to how the hub processes the ICMP and ARP packets. After that, once again, use "Add Simple PDU" to issue a ping from PC1 to PC0. Play the simulation again. How has the behavior of the hub changed from the first and second ping attempts, if at all?

2) Still in Simulation mode, in the Event List Filters, enable only ICMP and ARP. Using "Add Simple PDU", issue a ping from PC3 to PC4. Play the simulation. Pay close attention to how the switch processes the ICMP and ARP packets. After that, once again, use "Add Simple PDU" to issue a ping from PC4 to PC3. Play the simulation again. How has the behavior of the switch changed from the first and second ping attempts, if at all? In what ways did the switch process the packets similarly or differently from the hub between the first and second ping attempts?
By this time, you should be clear about how switch and hub differ from each other.

Tasks:

Build the following network in Packet Tracer:



Task 1:

Assign first 6 PC's with class C addresses and last three with Class B addresses. Now send packets using PDU tool from PC0 to PC3. Does the packet transmit?

	From	To
Class A	0.0.0.0 (Netid, Hostid)	127.255.255.255 (Netid, Hostid)
Class B	128.0.0.0 (Netid, Hostid)	191.255.255.255 (Netid, Hostid)
Class C	192.0.0.0 (Netid, Hostid)	223.255.255.255 (Netid, Hostid)
Class D	224.0.0.0 (Multicast Address)	239.255.255.255 (Multicast Address)
Class E	240.0.0.0 (Reserved)	255.255.255.255 (Reserved)

Task 2:

Now assign all the PC's with class C addresses and again send packets using PDU tool from PC0 to PC3. Does this ping this time?

Task 3:

Using PDU tool, send the packets from PC6 to PC4 and observe the transmission of switch and the hub. How they differ?

Task 4:

Again, using PDU tool, send packets to complete the MAC address tables of both the switches. Also observe how the switches update their MAC tables.

Aim: Study of basic network command and Network configuration commands

Apparatus (Software): Command Prompt and Packet Tracer.

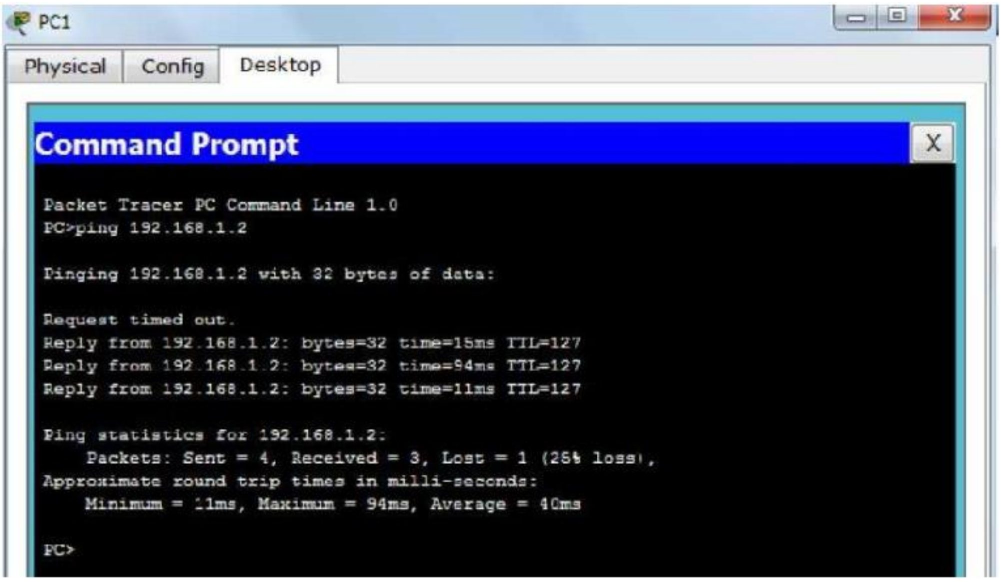
Procedure: To do this experiment- follows these steps:

In this experiment- students have to understand basic networking commands e.g ping, tracert etc.

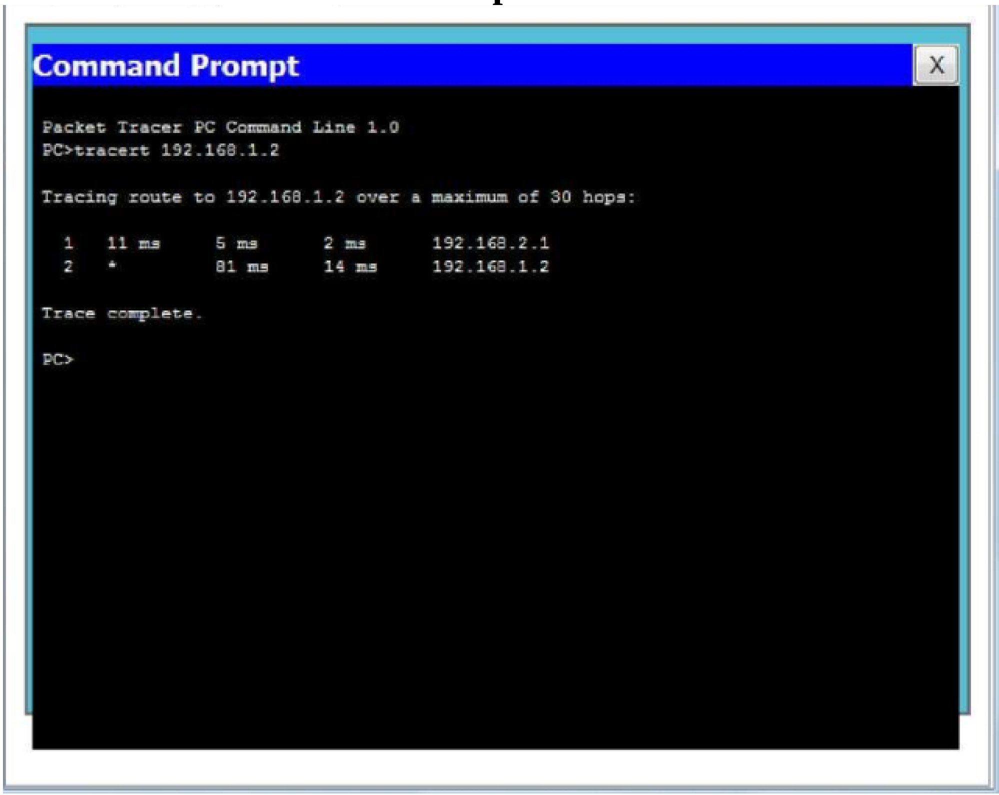
All commands related to Network configuration which includes how to switch to privilege mode and normal mode and how to configure router interface and how to save this configuration to flash memory or permanent memory.
These commands include:

- Configuring the Router commands
- General Commands to configure network
- Privileged Mode commands of a router
- Router Processes & Statistics
- IP Commands
- Other IP Commands e.g. show ip route etc.

ping:
ping(8) sends an ICMP ECHO_REQUEST packet to the specified host. If the host responds, you get an ICMP packet back. Sound strange? Well, you can “ping” an IP address to see if a machine is alive. If there is no response, you know something is wrong.



Traceroute:
Tracert is a command which can show you the path a packet of information taken from your computer to one you specify. It will list all the routers it passes through until it reaches its destination or fails to and is discarded. In addition to this, it will tell you how long each 'hop' from router to router takes.



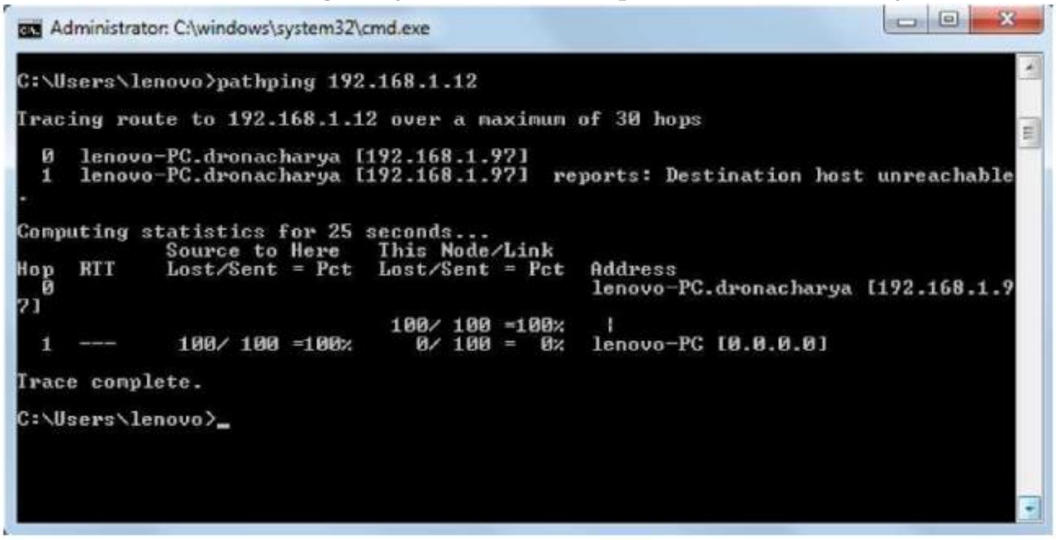
nslookup:

Displays information from Domain Name System (DNS) name servers.

NOTE: If you write the command as above it shows as default your pc's server name firstly.

pathping:

A better version of tracert that gives you statics about packet lost and latency.



Getting Help

In any command mode, you can get a list of available commands by entering a question mark(?).

Router>**?**

To obtain a list of commands that begin with a particular character sequence, type in those characters followed immediately by the question mark (?).

Router#**co?**

configure connect copy

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark.

Router#**configure ?**

memory Configure from NV memory network Configure from a TFTP network host terminal Configure from the terminal

You can also abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh**.

Configuration Files

Any time you make changes to the router configuration, you must save the changes to memory because if you don't, they will be lost if there is a system reload or power outage. There are two types of configuration files: the running (current operating) configuration and the startup configuration.

Use the following privileged mode commands to work with configuration files.

CSE-331 Computer Networks

Lab No. 05 CLO3

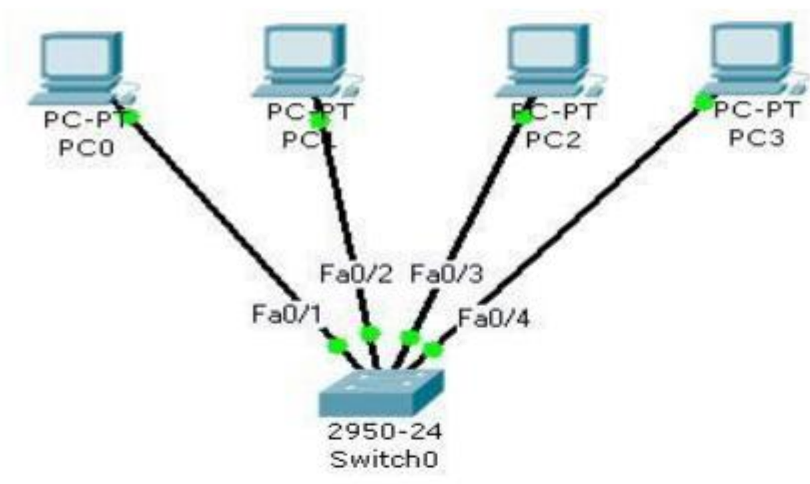
Task	1	2	3	4	5
Set time and date of a router having your ID as hostname	Student select a router	Student set the hostname of router as their respective registration ID	Student set the time and date of the system as current	Student enables user-mode password with their name	Student enables privileged-mode password with their name
Router configuration	Student designs the network as said	Student configure the router	Student assign Class B IP address to the PCs	Student ping the default gateway	Student save the current configuration

Design a network having one router, a switch and two PC's as shown in the figure. Configure the router and also assign the **Class B** IP addresses to the PC's and ping the default gateway. Also save the current configuration of the router.

Aim: Router Configuration using CLI

ACTIVITY 1

SWITCHED NETWORK



Step 1 Open the PT activity file then add a Switch and four generic PC's. Arrange them as shown in the above figure.

Step 2 Connect the devices using the appropriate connection type for each link**.**

Step 3 Using the Config tab, configure the IP address and subnet mask for each PC as shown in the following table:

- NVRAM : start up configuration .
- CPU .
- PCMCIA (flash memory) .

2- Software (IOS) .

Router configuration modes :

1- User mode (router>).

2- Privilage mode (router #)

CSE-331 Computer Networks

PC	IP Address	Subnet Mask
PC0	172.16.128.1	255.255.192.0
PC1	172.16.128.2	255.255.192.0
PC2	172.16.128.3	255.255.192.0
PC3	172.16.128.4	255.255.192.0

Global

...ed in user EXEC mode. The user EXEC commands are a subset of the privileged EXEC commands.

Privileged EXEC Mode:

Privileged commands include the following:
- Configure – Changes the software configuration.
- Debug – Display process and hardware event messages.
- Setup – Enter configuration information at the prompts.

Enter the command disable to exit from the privileged EXEC mode and return to user EXEC mode.

Configuration Mode:

Configuration mode has a set of submodes that you use for modifying interface settings, routing protocol settings, line settings, and so forth. Use caution with configuration mode because all changes you enter take effect immediately.

Step 4 Verify that the connections are correct. At this point, all link lights should be green.

Step 5 Change the bandwidth on PC0 to 10 Mbps and set the duplex settings to full instead of Auto.

Notice that the connection immediately goes down. To bring it back up again, change the bandwidth and the duplex settings of the switch's Ethernet port to match the current settings of PC0.

Questions:

- Explain why the connection fails if the bandwidth or duplex settings for the devices do not match.
- If a device is configured with bandwidth and duplex settings on automatic, does the device at the other end of the connection have to be set to automatic as well?

ACTIVITY 2

Router:

Functions:

1- IP addressing.

2- Routing.

Components:

1- Hardware:
- Interfaces.
- DRAM: running configuration.
- NVRAM: start up configuration.
- CPU.
- PCMCIA (flash memory).

2- Software (IOS).

Router configuration modes:

1- User mode (router>).
2- Privilage mode (router #).
3- Global configuration mode (router (config)#).

User enable / disable Privilage conf t / exit Global

User EXEC Mode:

When you are connected to the router, you are started in user EXEC mode. The user EXEC commands are a subset of the privileged EXEC commands.

Privileged EXEC Mode:

Privileged commands include the following:
• Configure – Changes the software configuration.
• Debug – Display process and hardware event messages.
• Setup – Enter configuration information at the prompts.

Enter the command disable to exit from the privileged EXEC mode and return to user EXEC mode.

Configuration Mode:

Configuration mode has a set of submodes that you use for modifying interface settings, routing protocol settings, line settings, and so forth. Use caution with configuration mode because all changes you enter take effect immediately.

To enter configuration mode, enter the command configure terminal and exit by pressing Ctrl-Z.

Getting Help:

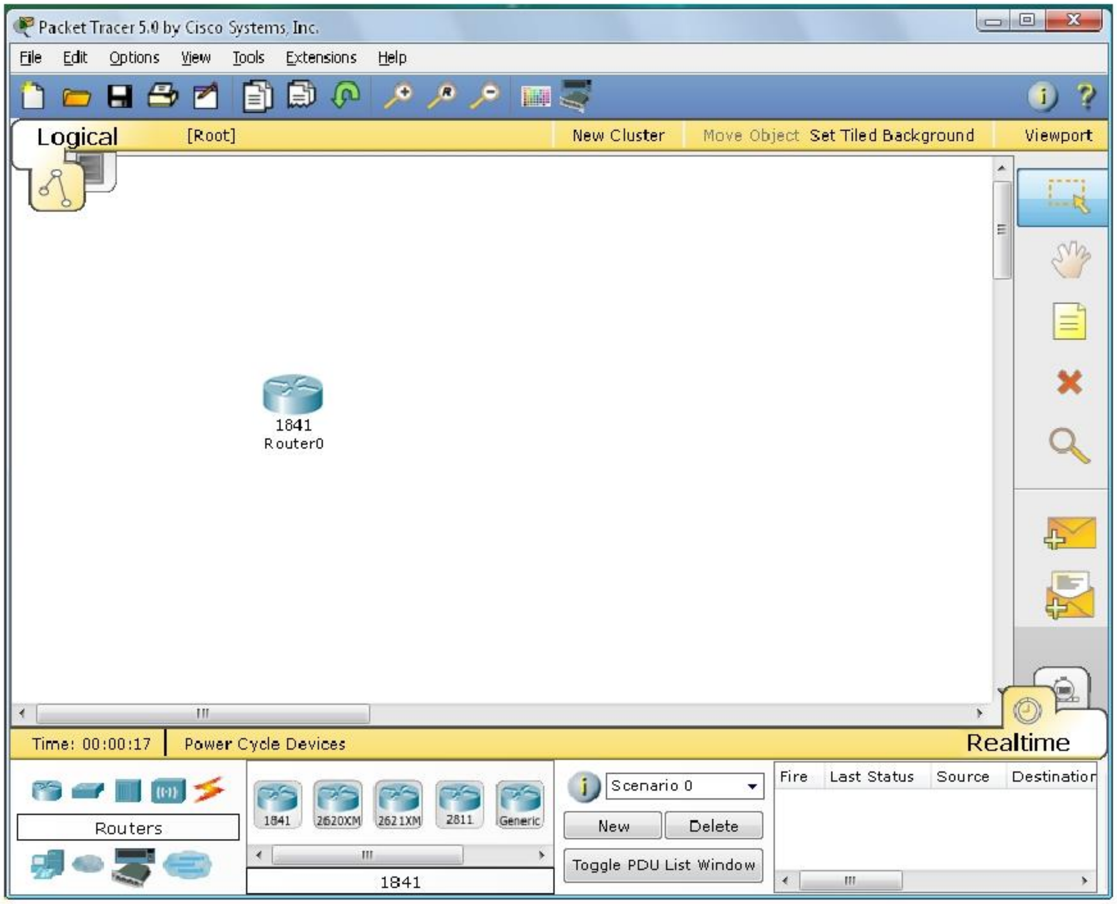
In any command mode, you can get a list of available commands by entering a question mark (?).

Router>?

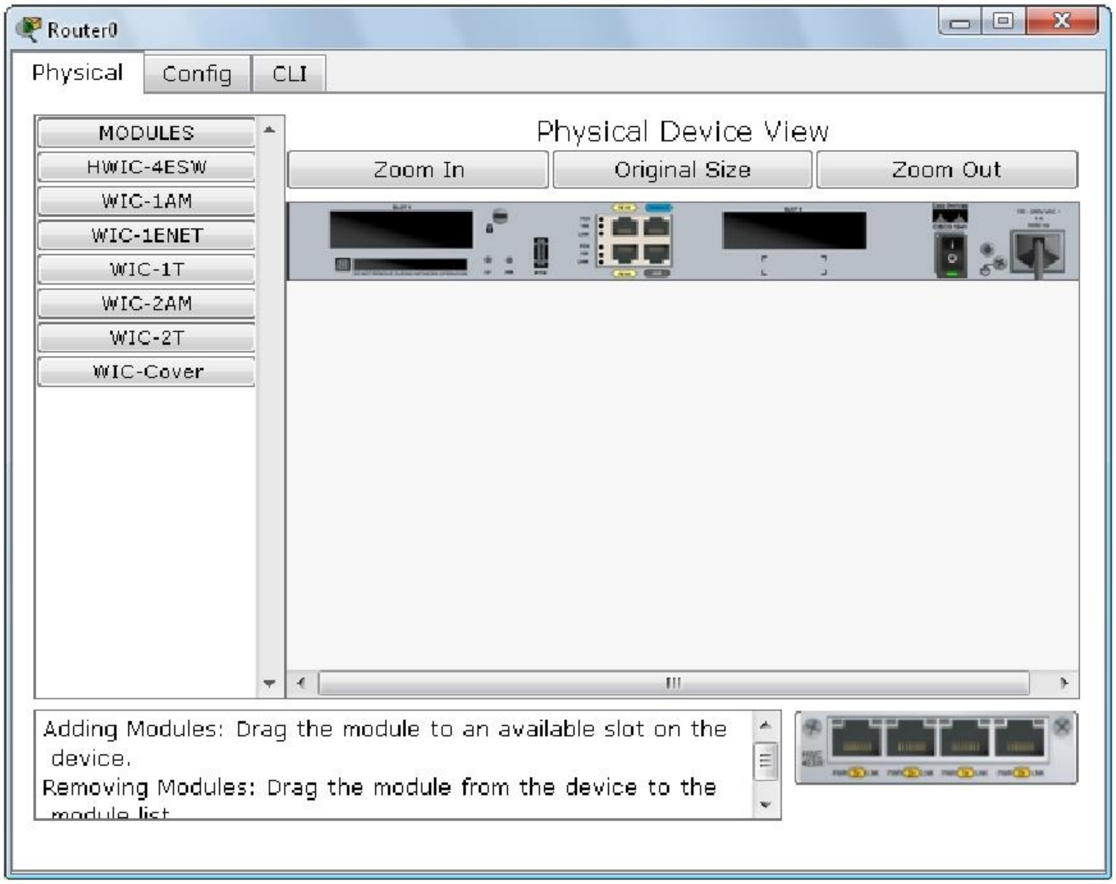
To obtain a list of command that begin with a particular character sequence, type in those characters followed immediately by the question mark (?).

We use packet tracer program for router configuration in the previous 3 modes.

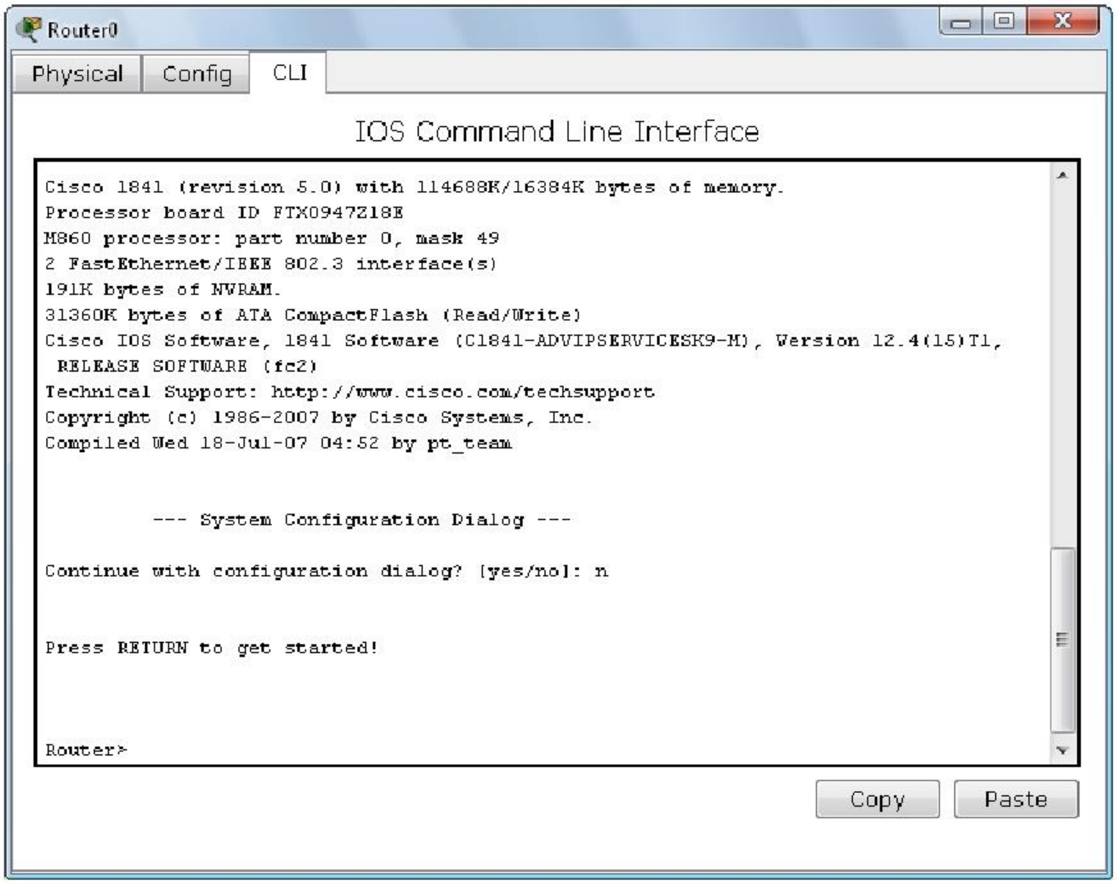
We choose a router:



Double click on the router chosen:

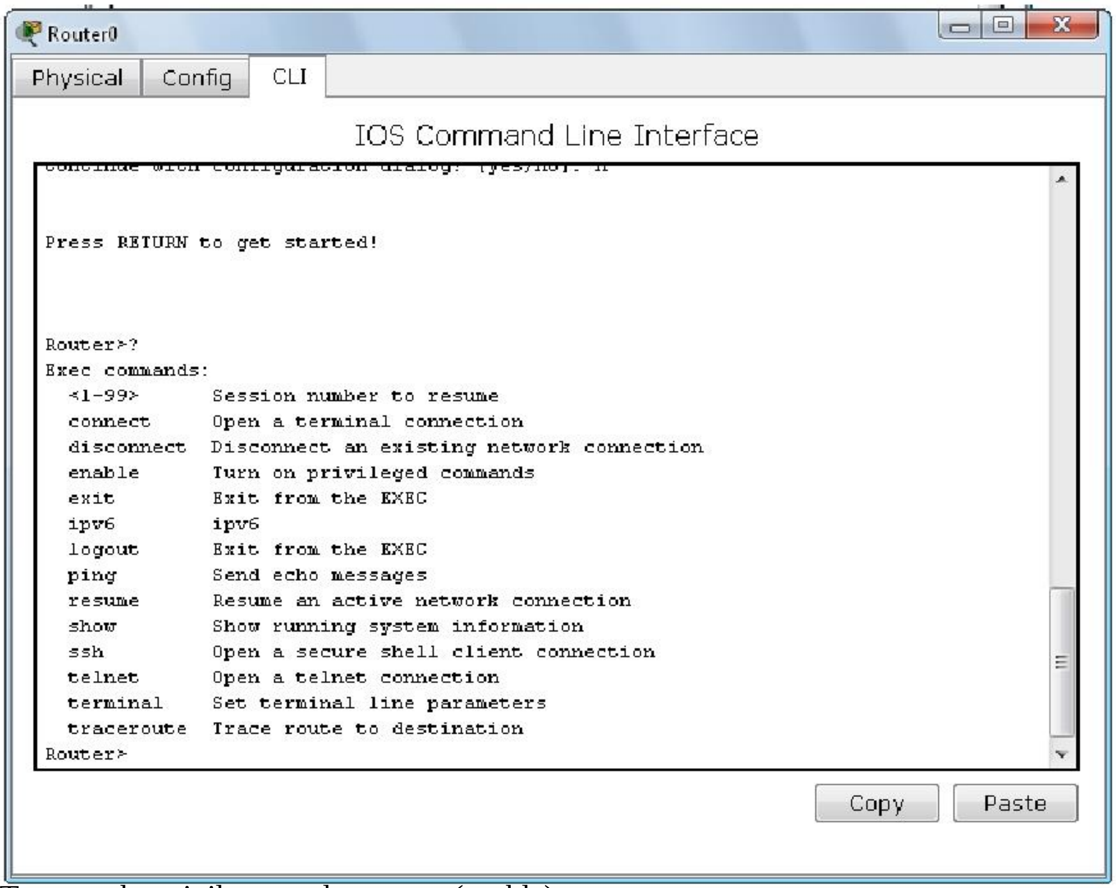


We enter CLI for router configuration:

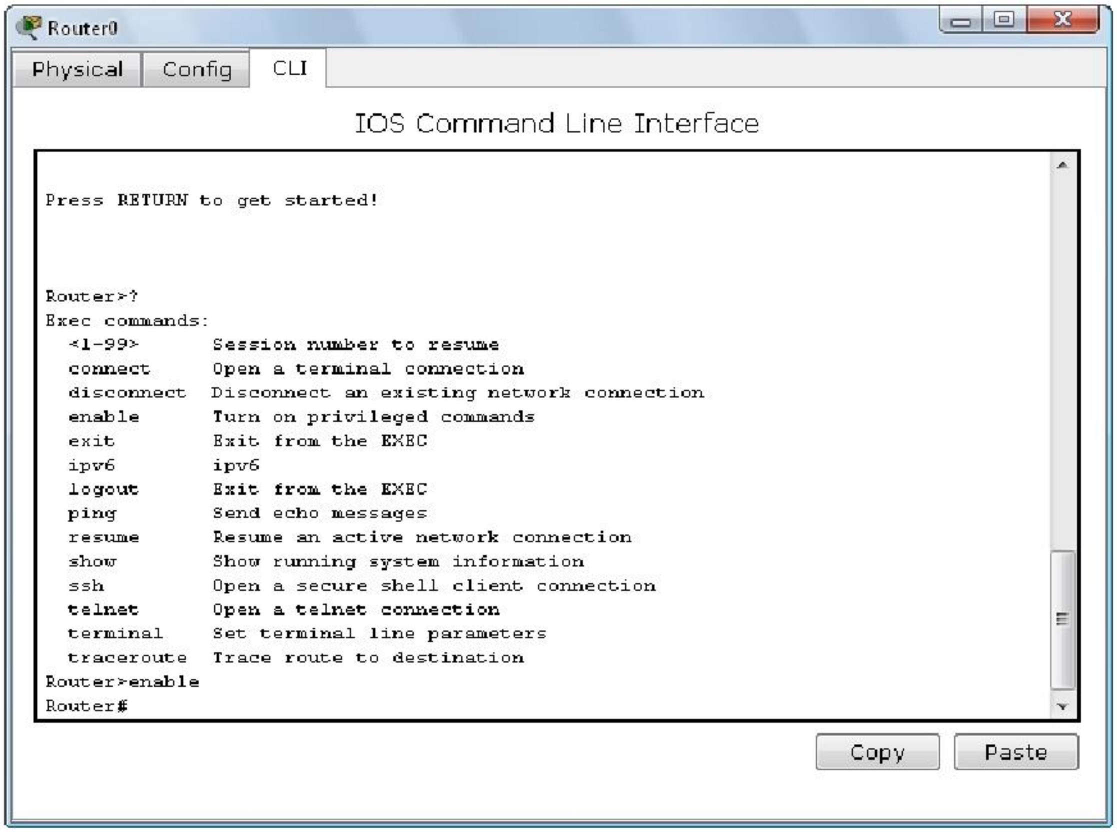


Now we are in the user mode:

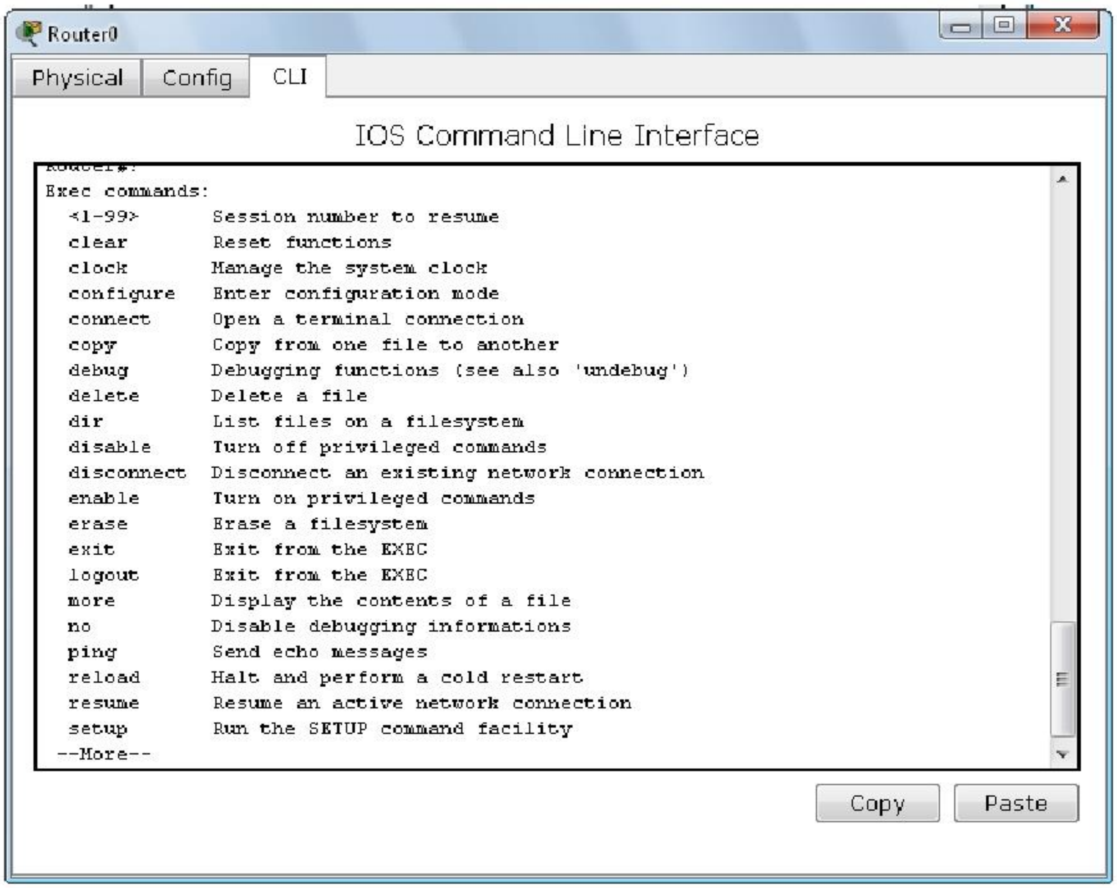
To know the commands in user mode we use (?) :



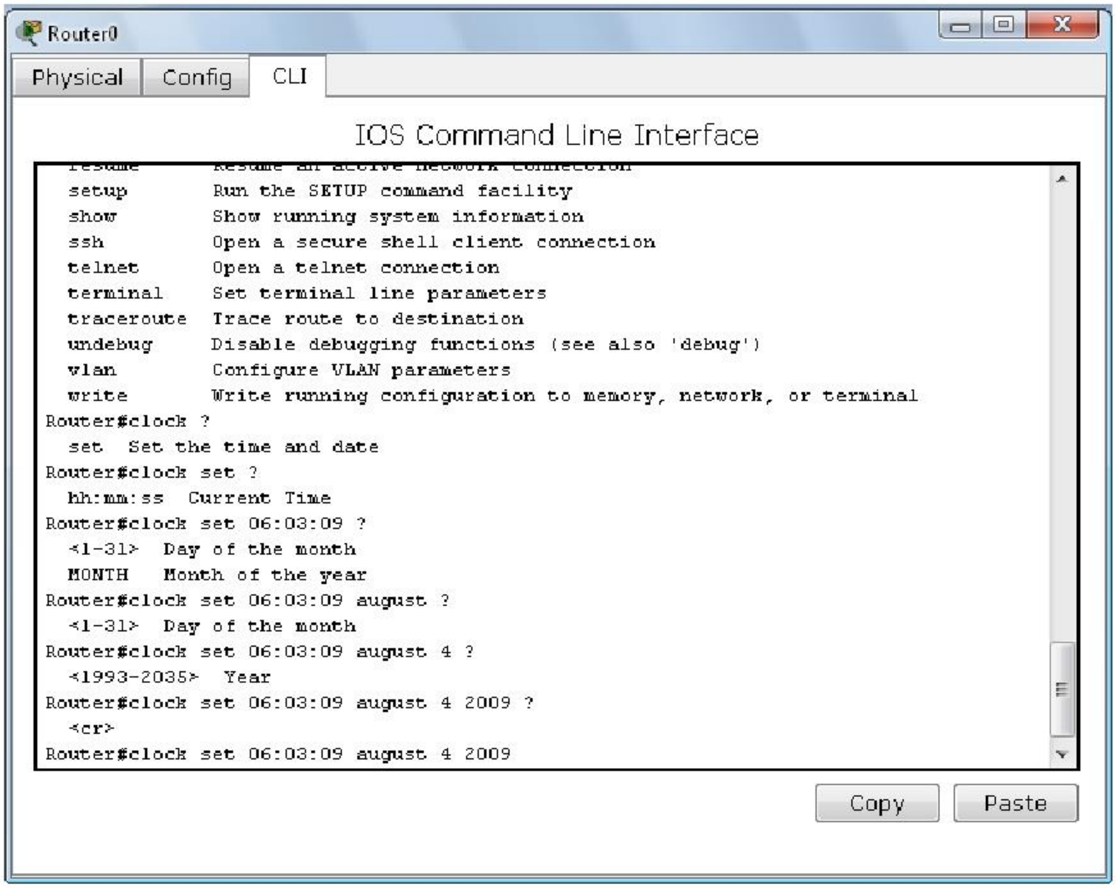
To enter the privilege mode, we use (enable):



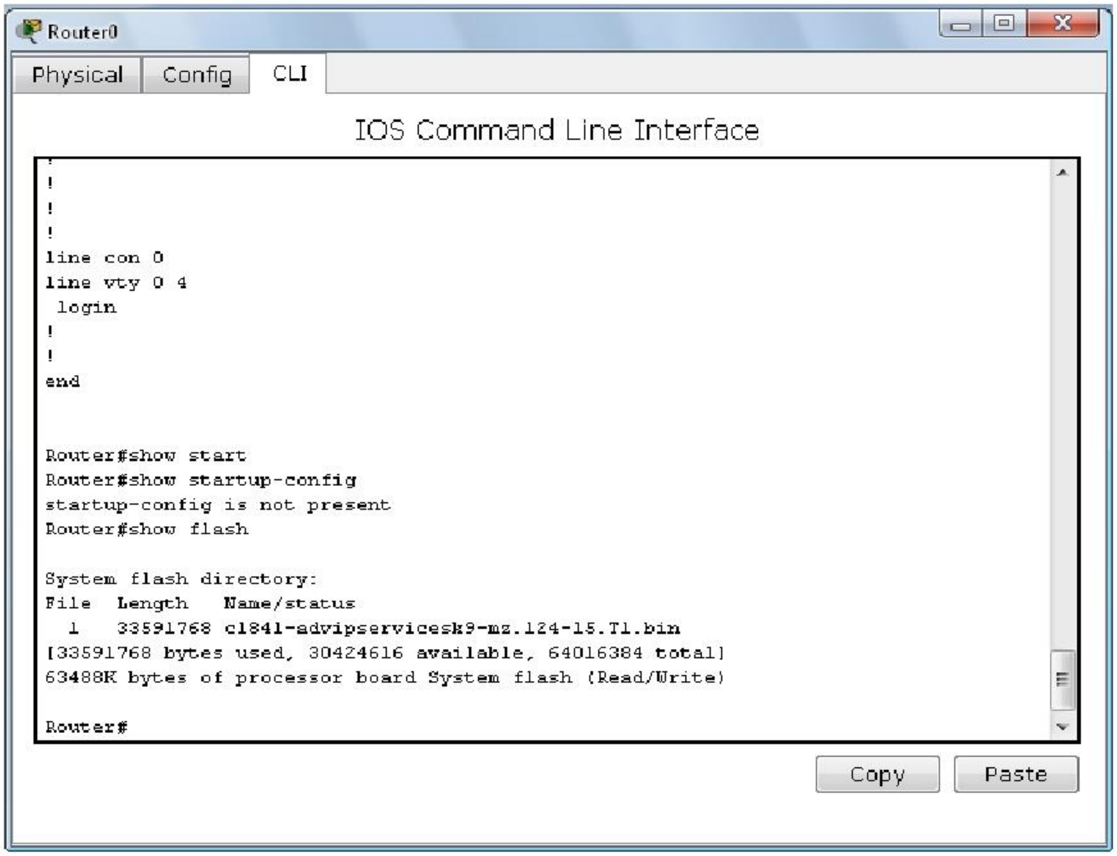
To see the commands in privileged mode we use (?):



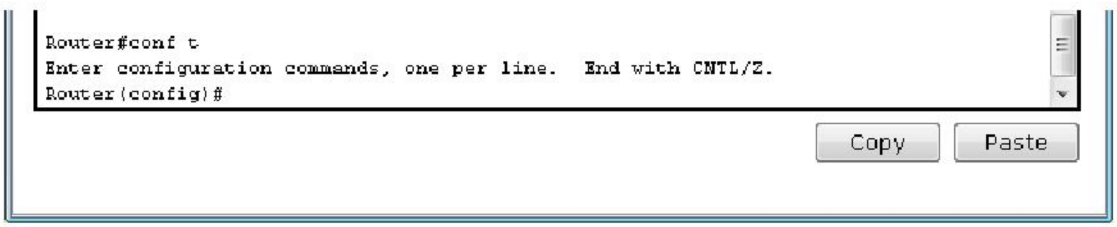
To manage the system clock, we use (clock):



To see the time, we use (show clock):

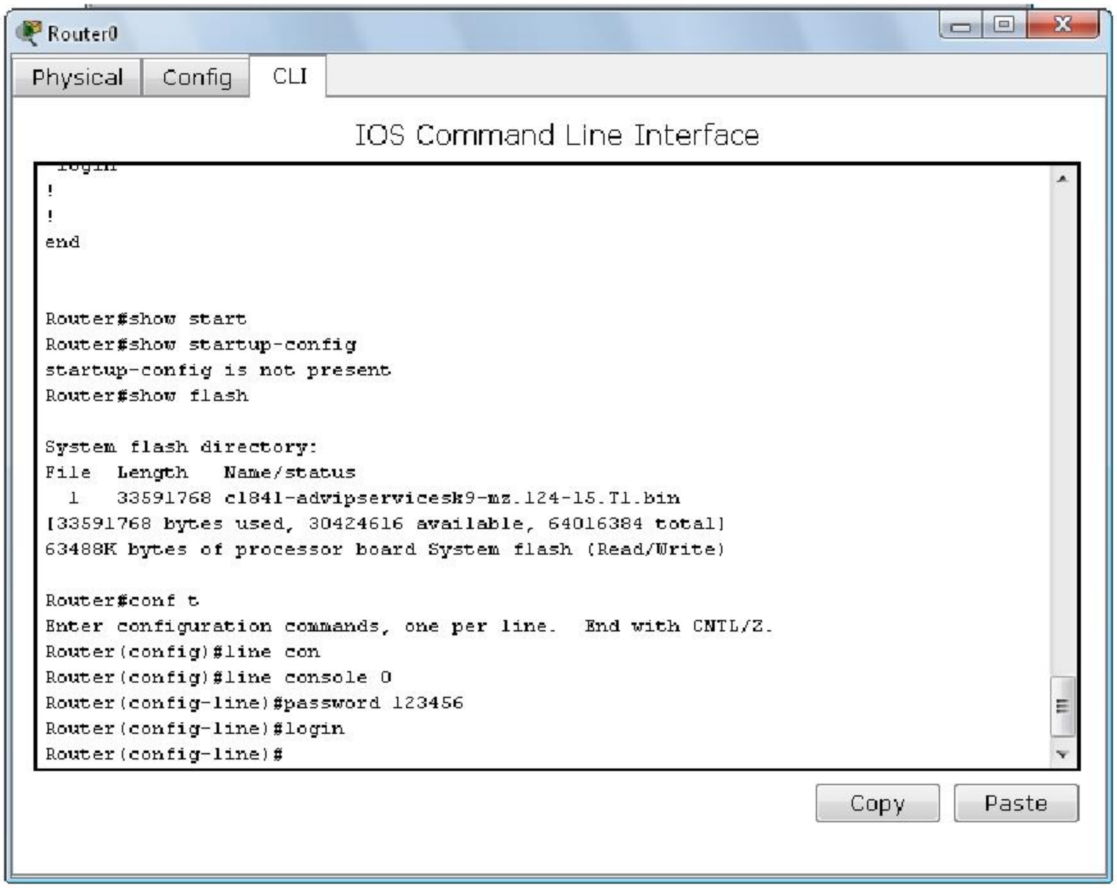


To go to global configuration, we use (conf t):

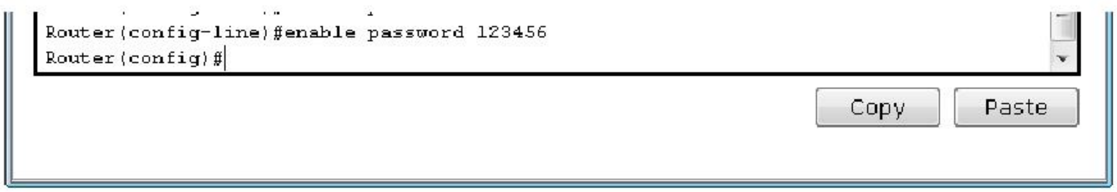


Passwords:

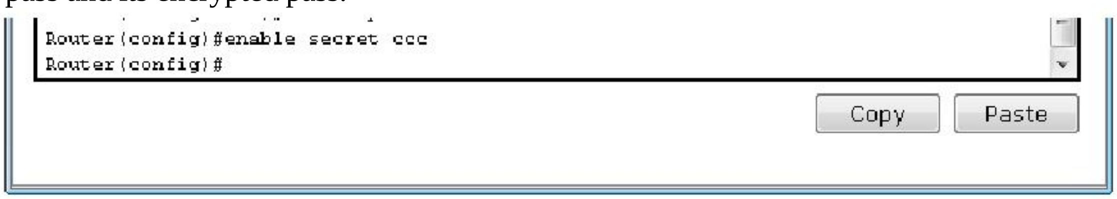
1- Line console password to protect the user mode:



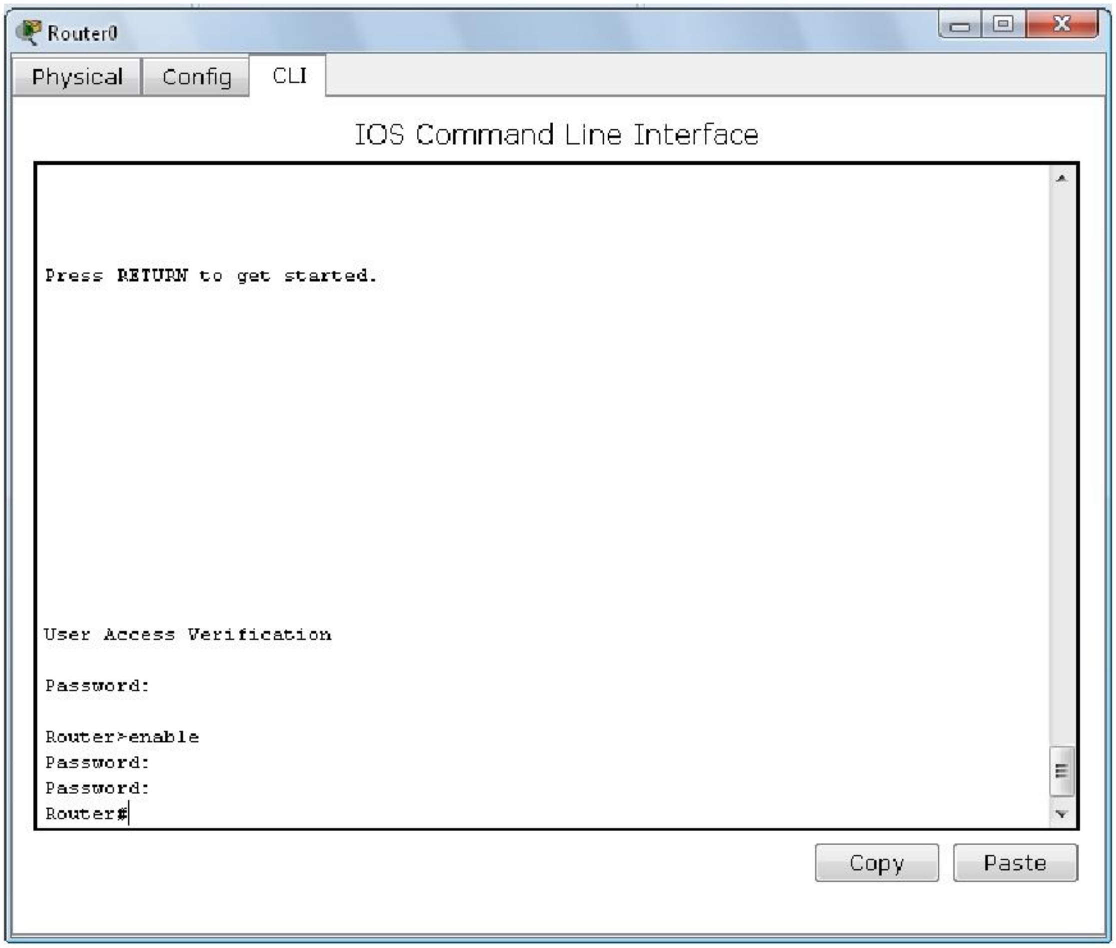
2- Enable password to protect the privilege mode:



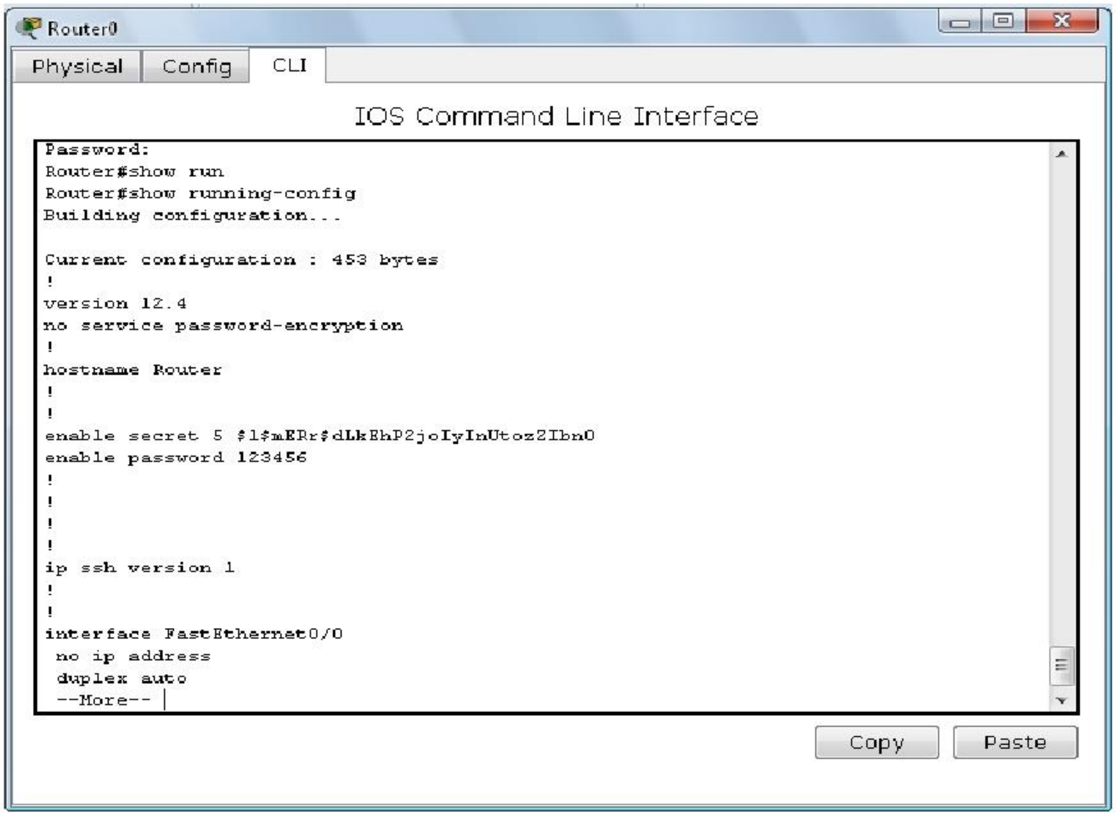
3- Secret password for more protection of privilege mode (more priority than enable pass and its encrypted pass.



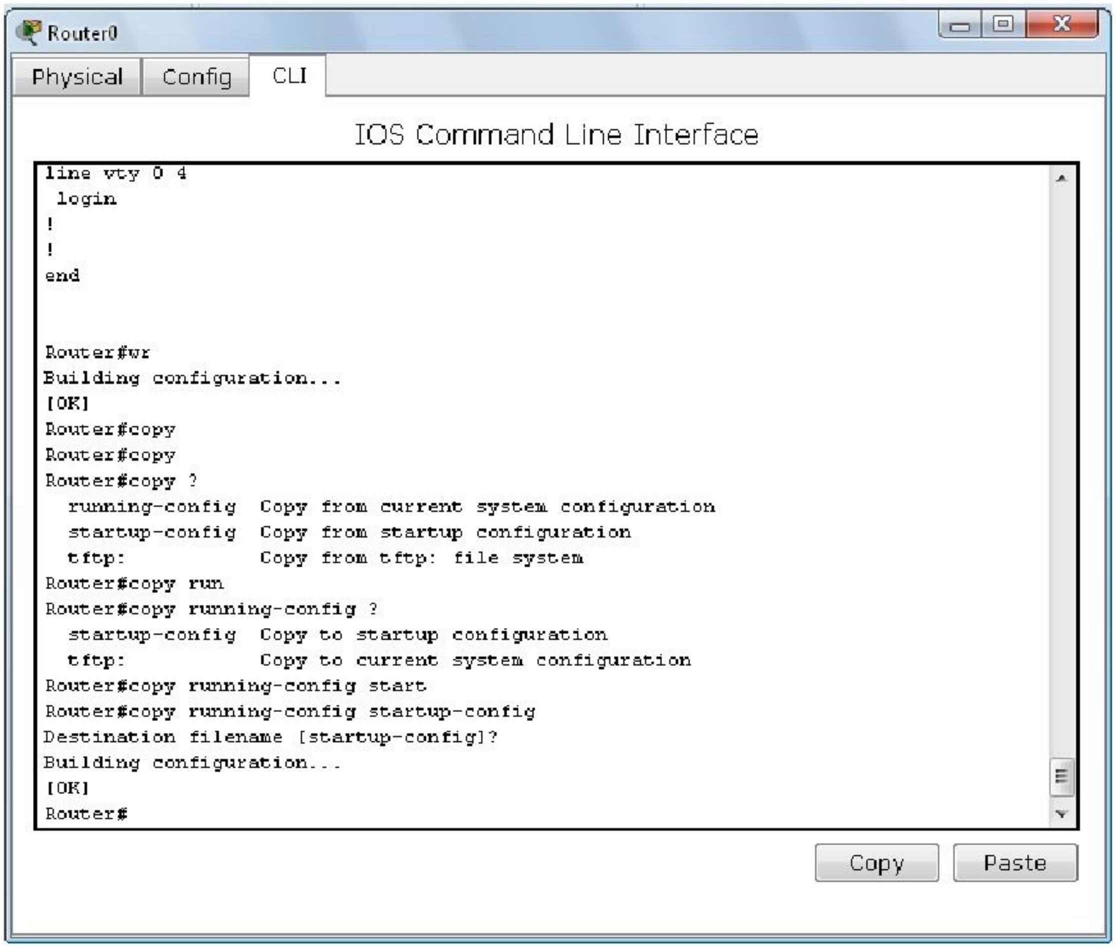
Now we try the passwords set:



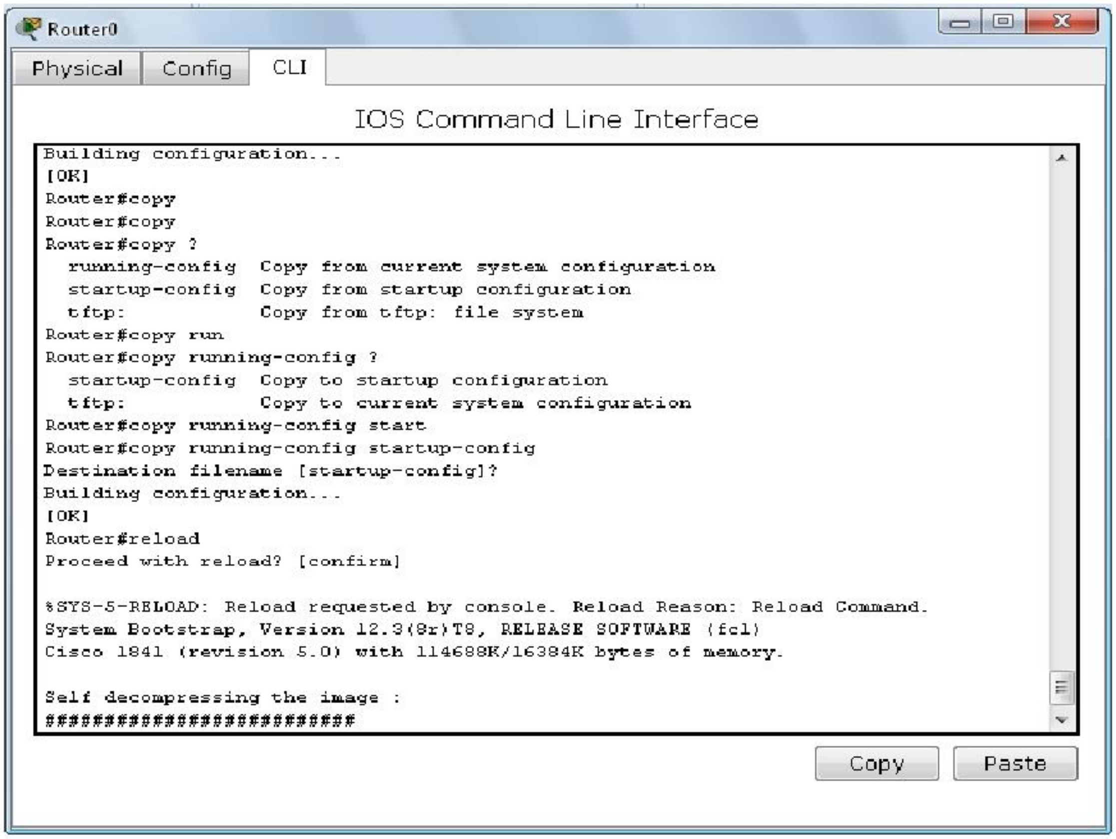
We notice that only secret pass is encrypted:

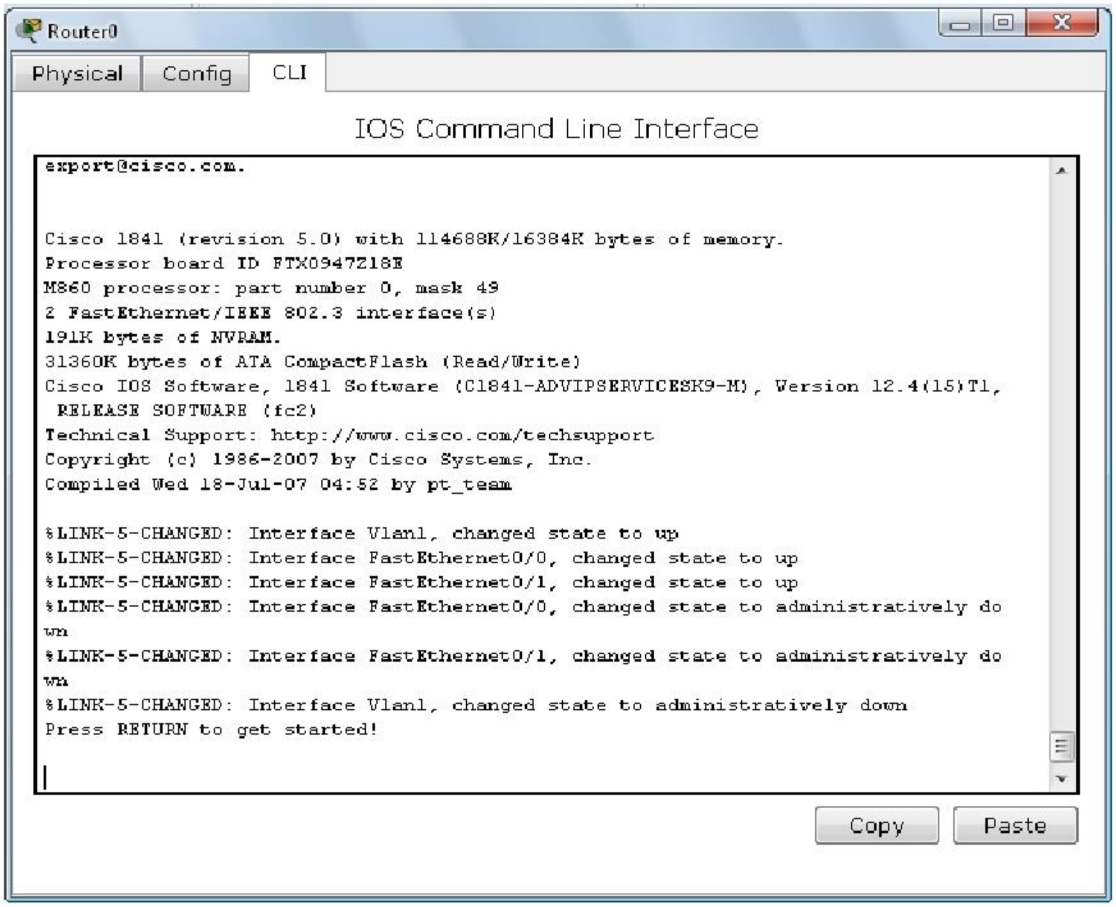


To save the configuration we did in the NVRAM we use (wr or copy commands).



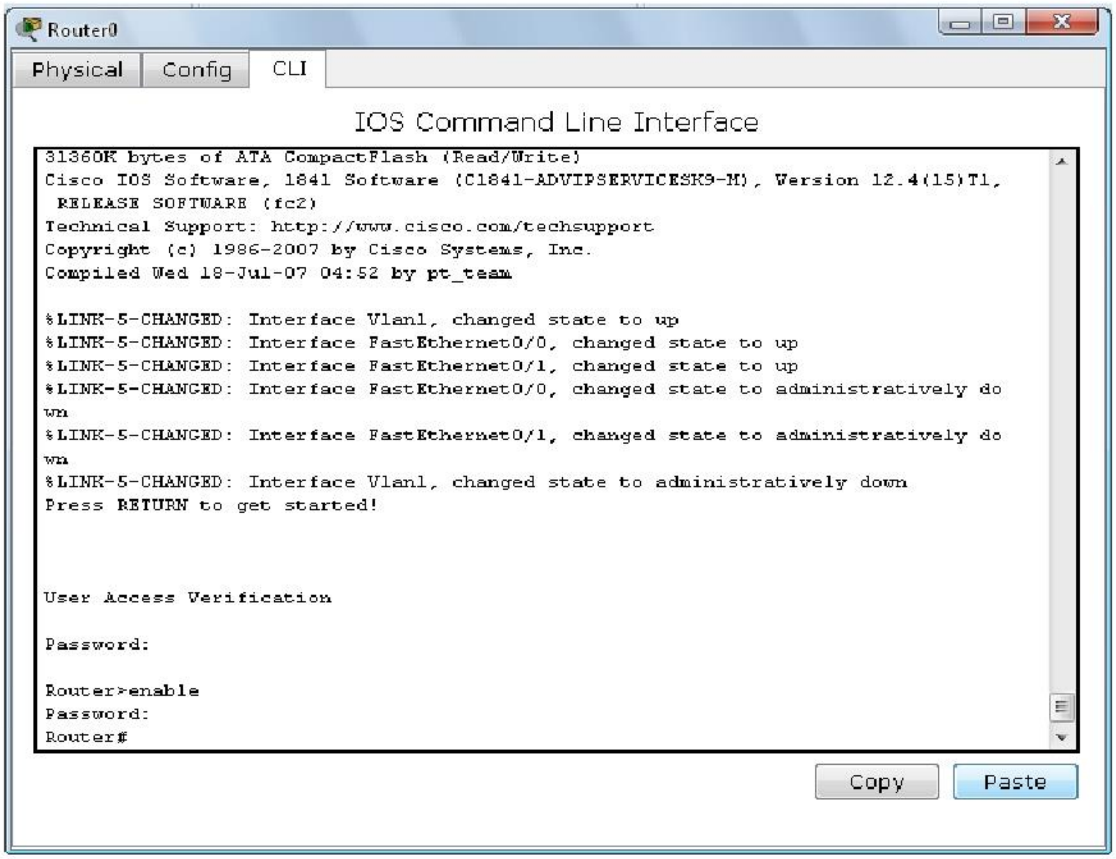
To restart we use the command (reload):

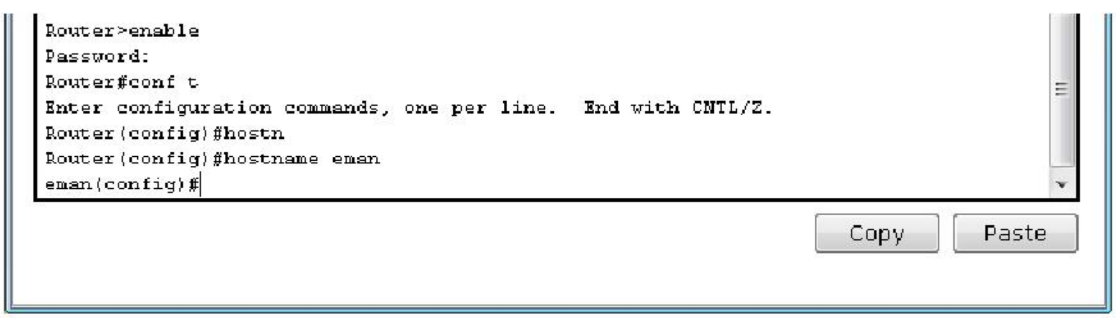

Router0
Physical
Config
CLI
IOS Command Line Interface
Building configuration...
[OK]
Router#copy
Router#copy
Router#copy ?
running-config Copy from current system configuration
startup-config Copy from startup configuration
tftp: Copy from tftp: file system
Router#copy run
Router#copy running-config ?
startup-config Copy to startup configuration
tftp: Copy to current system configuration
Router#copy running-config start
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#reload
Proceed with reload? [confirm]
%SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload Command.
System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Self decompressing the image :
##########################
Copy
Paste


Router0
Physical
Config
CLI
IOS Command Line Interface
export@cisco.com.
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Processor board ID FTX0947Z18E
M860 processor: part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
191K bytes of NVRAM.
31360K bytes of ATA CompactFlash (Read/Write)
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
%LINK-5-CHANGED: Interface Vlan1, changed state to administratively down
Press RETURN to get started!
Copy
Paste

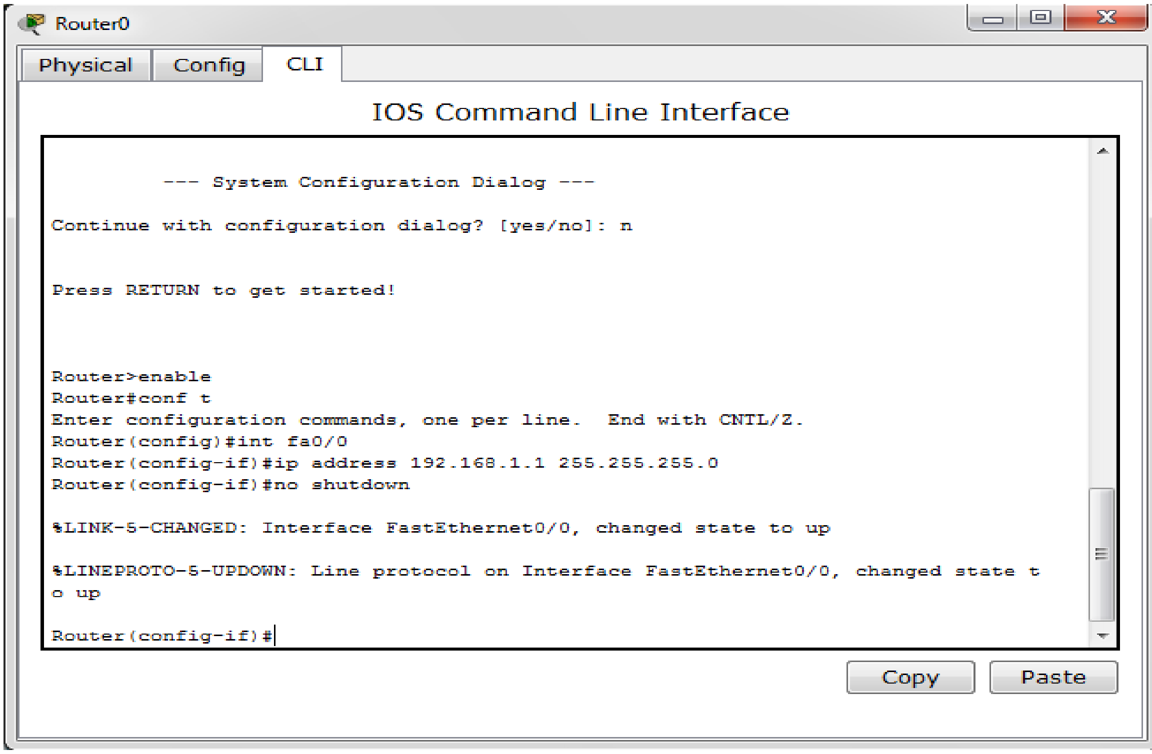
CSE-331 Computer Networks

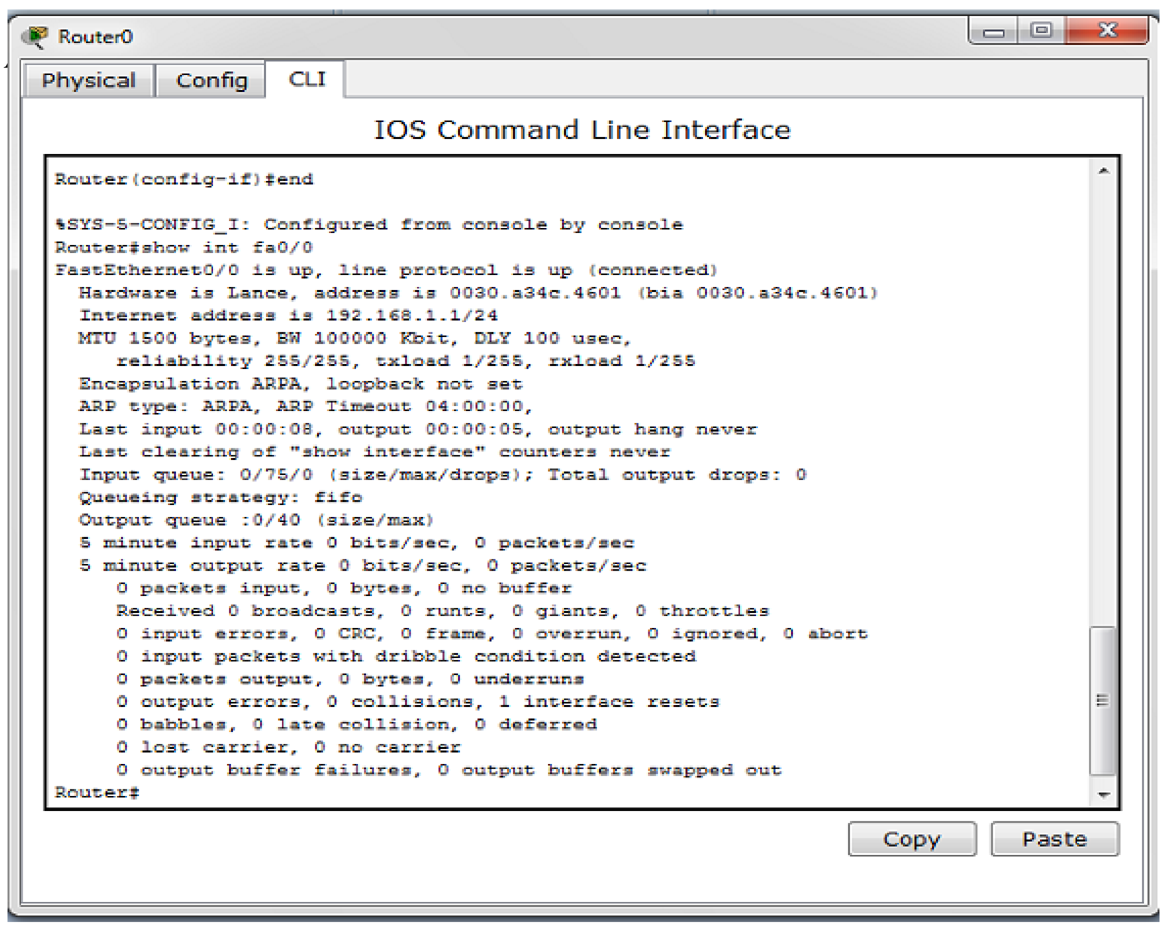
To change the host name of router:





Now to assign the IP's to the interface of the router:





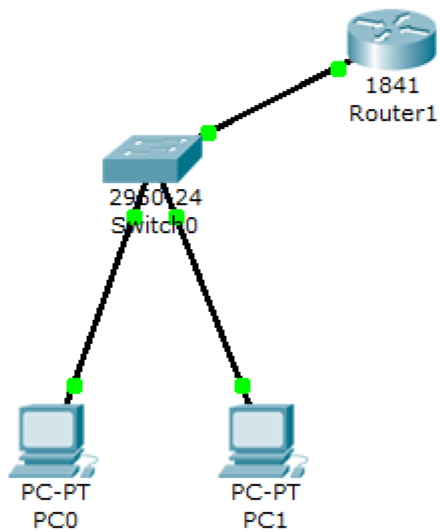
CSE-331 Computer Networks

Task 1:

Choose a router in packet tracer. Name it with your registration # e.g., 2018-CS-XX. Set its Time and Date as today. Enable user mode and privileged mode passwords with your name.

Task 2:

Design a network having one router, a switch and two PC's as shown in the figure. Configure the router and also assign the **Class B** IP addresses to the PC's and ping the default gateway. Also save the current configuration of the router.



Lab manual 6 **CLO3**

Task	0	2	4	8	10
Create topology and apply all steps	Student establish some connections	Student assign IP address to all devices as given in figure	Student does static routing accurately	Student assign correct gateway to each network	"Ping" command is working on all devices

IP naming:

Since IP addresses are rather difficult to remember (and are not particularly descriptive), the Internet also allows you to specify a device/hope by a *name* rather than a number string. For example, the machine at CS department with the IP address 18.72.0.3 can also be referred to as: *bitsy.cs.uet.edu*. This whole string is known as the computer's *host name*. In this string, the first part ("bitsy") is the name of the machine itself, while everything else ("cs.uet.edu") is the *domain name*.

We commonly use two methods for IP Naming.

Using Host
Table Using
DNS Server

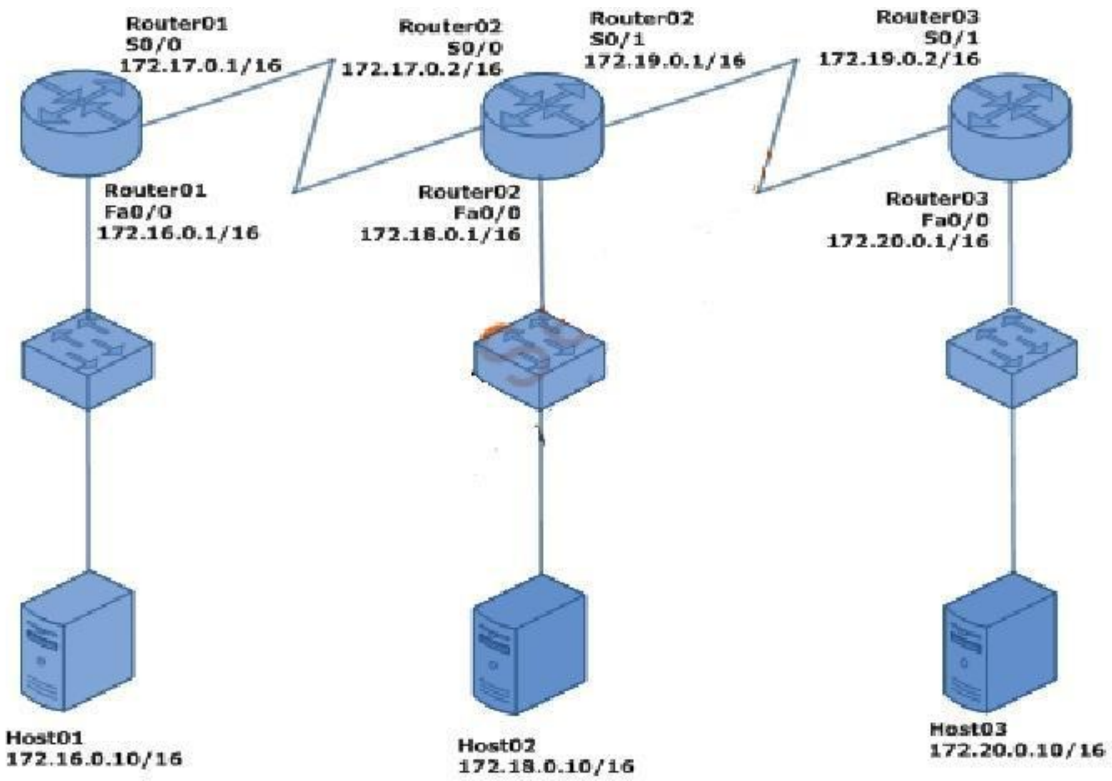
IP naming using host table.

It has the following steps:

- Configure host names and interfaces.
- Assign IP addresses to all interfaces (in use) of all routers Set VTY and enable passwords on all routers
- Add static routes on all routers
- Create host tables on all routers

Activity 1:

The following diagram shows our lab setup. We have three routers, three switches and three hosts connected as below. The host names, IP addresses and the interfaces of the routers are shown in diagram. The IP addresses of the hosts are also shown in the diagram.



Step 1: Hostname, interface and IP address configurations in all routers.

- Configurations in Router01

Connect to Router01 console and use the following IOS commands to configure host name as Router01.

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router01
Router01(config)#

For Router interface configuration type "interface interface_name".

Router01(config)#interface fa0/0
router01(config-if)#

Assign an IP address to an interface. Run the following command from interface configuration mode.

Router01(config-if)#ip address 172.16.0.1 255.255.0.0

You have to enter both IP address and subnet mask.

Now enable router01 interface. Run "no shutdown" command from interface configuration mode.

Router01(config-if)#no shutdown
Router02(config-if)#exit
LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Note: To disable a router interface we run "shutdown" command from interface configuration mode.

Use the following IOS commands to open the serial interface S0/0 configuration mode on Router01 and configure IP address as 172.17.0.1/16. You have to set a clock rate also using the "clock rate" command on S0/0 interface, since this is the DCE side.

Router01(config)#interface s0/0/0
Router01(config-if)#clock rate 64000
Router01(config-if)#ip address 172.17.0.1 255.255.0.0
Router01(config-if)#no shutdown
Router02(config-if)#exit

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router.

- Configuration in Router02

Connect to Router02 console and use the following IOS commands to configure host name as Router02.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router02
Router02(config)#
```

Use the following IOS commands to open the fast ethernet interface Fa0/0 configuration mode on Router02 and configure IP address as 172.18.0.1/16.

```
Router02(config)#interface fa0/0
Router02(config-if)#ip address 172.18.0.1 255.255.0.0
Router02(config-if)#no shutdown
Router02(config-if)#exit
```

Use the following IOS commands to open the serial interface S0/0 configuration mode on Router02 and configure IP address as 172.17.0.2/16.

```
Router02(config)#interface s0/0/0
Router02(config-if)#ip address 172.17.0.2 255.255.0.0
Router02(config-if)#no shutdown
Router02(config-if)#exit
```

Use the following IOS commands to open the serial interface S0/1 configuration mode on Router02 and configure IP address as 172.19.0.1/16. You have to set a clock rate also using the "clock rate" command on S0/1 interface, since this is the DCE side.

```
Router02(config)#interface s0/0/1
Router02(config-if)#clock rate 64000
Router02(config-if)#ip address 172.19.0.1 255.255.0.0
Router02(config-if)#no shutdown
Router02(config-if)#exit
```

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router.

- Configurations in Router03

Connect to Router03 console and use the following IOS commands to configure host name as Router03.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router03
Router03(config)#
```

Use the following IOS commands to open the fast ethernet interface Fa0/0 configuration mode on Router03 and configure IP address as 172.20.0.1/16.

Router03(config)#interface fa0/0
Router03(config-if)#ip address 172.20.0.1 255.255.0.0
Router03(config-if)#no shutdown
Router02(config-if)#exit

Use the following IOS commands to open the serial interface S0/1 configuration mode on Router03 and configure IP address as 172.19.0.2/16.

outer03(config)#interface s0/0/1
Router03(config-if)#ip address 172.19.0.2 255.255.0.0
Router03(config-if)#no shutdown

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router..

Step 2: Configure Static Routes

Static Route can be configured by the following IOS commands.

• Router(config)#ip route destination_network subnet_mask default_gateway [administrative_distance] [permanent]

OR

• Router(config)# ip route destination_network subnet_mask interface_to_exit [administrative_distance] [permanent]

The permanent keyword will keep the static route in the routing table even when the interface the router uses for the static route fails.

- Static Routing configuration in Router01

Connect to Router01 console and use the following IOS commands to configure static routing in Router01. The "ip route" commands shown below states that to reach 172.18.0.0/16, 172.19.0.0/16 and 172.20.0.0/16 networks, handover the packets to the gateway ip address 172.17.0.2. The networks 172.16.0.0/16 and 172.17.0.0/16 are conneted directly to Router01.

Router01>enable
Router01#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router01(config)#ip route 172.18.0.0 255.255.0.0 172.17.0.2
Router01(config)#ip route 172.19.0.0 255.255.0.0 172.17.0.2
Router01(config)#ip route 172.20.0.0 255.255.0.0 172.17.0.2

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router.

To view the routing table in Router01, run "show ip route" command in Router01 as shown below.

```
Router01#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 172.16.0.0/16 is directly connected, FastEthernet0/0 C
172.17.0.0/16 is directly connected, Serial0/0
S 172.18.0.0/16 [1/0] via 172.17.0.2
S 172.19.0.0/16 [1/0] via 172.17.0.2
S 172.20.0.0/16 [1/0] via 172.17.0.2
```

The "S" character at the beginning of a line in routing table shows that it is a static route and "C" character shows that it is a directly connected network.

- Static Routing configuration in Router02

Connect to Router02 console and use the following IOS commands to configure static routing in Router02. The "ip route" commands shown below states that to reach 172.16.0.0/16 network, handover the packets to the gateway ip address 172.17.0.1 and to reach 172.20.0.0/16 network, handover the packets to the gateway ip address 172.19.0.2. The networks 172.17.0.0/16, 172.18.0.0/16 and 172.19.0.0/16 are connected directly to Router02.

```
Router02>enable
Router02#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router02(config)#ip route 172.16.0.0 255.255.0.0 172.17.0.1
Router02(config)#ip route 172.20.0.0 255.255.0.0 172.19.0.2
```

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router.

To view the routing table in Router02, run "show ip route" command in Router02 as shown below.

```
Router02#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
S 172.16.0.0/16 [1/0] via 172.17.0.1
```

C 172.17.0.0/16 is directly connected, Serial0/0
C 172.18.0.0/16 is directly connected, FastEthernet0/0 C
172.19.0.0/16 is directly connected, Serial0/1
S 172.20.0.0/16 [1/0] via 172.19.0.2

The "S" character at the beginning of a line in routing table shows that it is a static route and "C" character shows that it is a directly connected network.

- Static Routing configuration in Router03

Connect to Router03 console and use the following IOS commands to configure static routing in Router03. The "ip route" commands shown below states that to reach 172.16.0.0/16, 172.17.0.0/16 and 172.18.0.0/16 networks, handover the packets to the gateway ip address 172.19.0.1. The networks 172.19.0.0/16 and 172.20.0.0/16 are conneted directly to Router03.

```
Router03>enable
Router03#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router03(config)#ip route 172.16.0.0 255.255.0.0 172.19.0.1
Router03(config)#ip route 172.17.0.0 255.255.0.0 172.19.0.1
Router03(config)#ip route 172.18.0.0 255.255.0.0 172.19.0.1
```

Do remember to run the "copy running-config startup-config" command from enable mode, if you want to save the changes you have made in the router.

To view the routing table in Router03, run "show ip route" command in Router03 as shown below.

```
Router03#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
S  172.16.0.0/16  [1/0]  via  172.19.0.1
S  172.17.0.0/16  [1/0]  via  172.19.0.1
S 172.18.0.0/16 [1/0] via 172.19.0.1
C  172.19.0.0/16 is directly connected, Serial0/1
C  172.20.0.0/16 is directly connected, FastEthernet
```

The "S" character at the beginning of a line in routing table shows that it is a static route and "C" character shows that it is a directly connected network.

Verify the connectivity between networks using the ping command

To verify the static routes which we have configured and the connectivity between networks, run the ping command from Host01 (IP address: 172.16.0.10/16) to Host03 (IP address: 172.20.0.10/16).

```
C:\>ping 172.20.0.10

Pinging 172.20.0.10 with 32 bytes of data:

Reply from 172.20.0.10: bytes=32 time=172ms TTL=125
Reply from 172.20.0.10: bytes=32 time=235ms TTL=125
Reply from 172.20.0.10: bytes=32 time=187ms TTL=125
Reply from 172.20.0.10: bytes=32 time=187ms TTL=125

Ping statistics for 172.20.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 172ms, Maximum = 235ms, Average = 195ms
```

The ping reply from Host03 (IP address: 172.20.0.10/16) shows that the static routes are configured well in three routers and there is network connectivity between different networks

Step 3: Configure enable and vty passwords on these routers.

VTY ports are virtual TTY ports, used to Telnet or SSH into the router over the network. You can use them to connect to the router to make configuration changes or check the status. Most routers have five VTY ports, numbered 0 to 4.

That means you can have up to five concurrent network admins configuring the router at one time. However, you can easily generate more VTY lines.

For example, to create a total of 21 VTY lines (numbered 0 through 20), enter the following:

```
Router (config)# line 0 20
```

Here's an example:

```
Router(config)# line vty 0 4
Router(config-line)# password UrRoll# (USE A STRONG PASSWORD)
Router(config-line)# logging synchronous
Router(config-line)# exec-timeout 60 0
Router(config-line)# transport input <telnet OR ssh>
```

Now configure Enable Password (as done in previous labs) one by one on all routers.

Keep in mind that you can always use the "*clear line*" command to clear out a connection on a router line if you run into a problem.

Step 4: Creating a Host Table for Telnet Access

The host table defines a static name-to-address mapping on the router. This host table enables you to Telnet to the router defined in the host table by its name that is defined in the table.

To create an IP host table, enter global configuration mode and use the following command:

Router#ip host *name address*

For demonstration purposes, create a host table on R1 for each lab router that will map the router name to its loopback IP address. (You will assign them to the routers in Chapter 7, "Router Interface Configuration.") Example 6-12 shows the host table configuration.

Example 6-12. Configuring and IP Host Table. Mapping the Router Name to Its Loopback IP Address

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router01(config)#ip host Router02 172.17.0.2
Router01(config)#ip host Router02 172.19.0.1
Router01(config)#ip host Router03 172.19.0.2
Router01(config)#exit
Router01#
```

Now examine the host table as it appears in the configuration by doing a show running-config.

Verification using TELNET

Router01# telnet router02

Lab task:

1- Create a topology as given above and apply all the steps one by one.

Note: every end device should be able to send communication packets (ping) from to other devices

Lab No.7 CLO2

Aim: Subnetting

What Is Subnetting?

Subnetting is the process of stealing bits from the HOST part of an IP address in order to divide the larger network into smaller sub-networks called subnets. After subnetting, we end up with NETWORK SUBNET HOST fields. We always reserve an IP address to identify the subnet and another one to identify the broadcast subnet address.

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

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

Understand IP Addresses

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

Here is how binary octets convert to decimal: The right most bit, or least significant bit, of an octet holds a value of 2^0. The bit just to the left of that holds a value of 2^1. This continues until the left-most bit, or most significant bit, which holds a value of 2^7. So if all binary bits are a one, the decimal equivalent would be 255 as shown here:

1 1 1 1 1 1 1 1

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

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

0 1 0 0 0 0 0 1

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

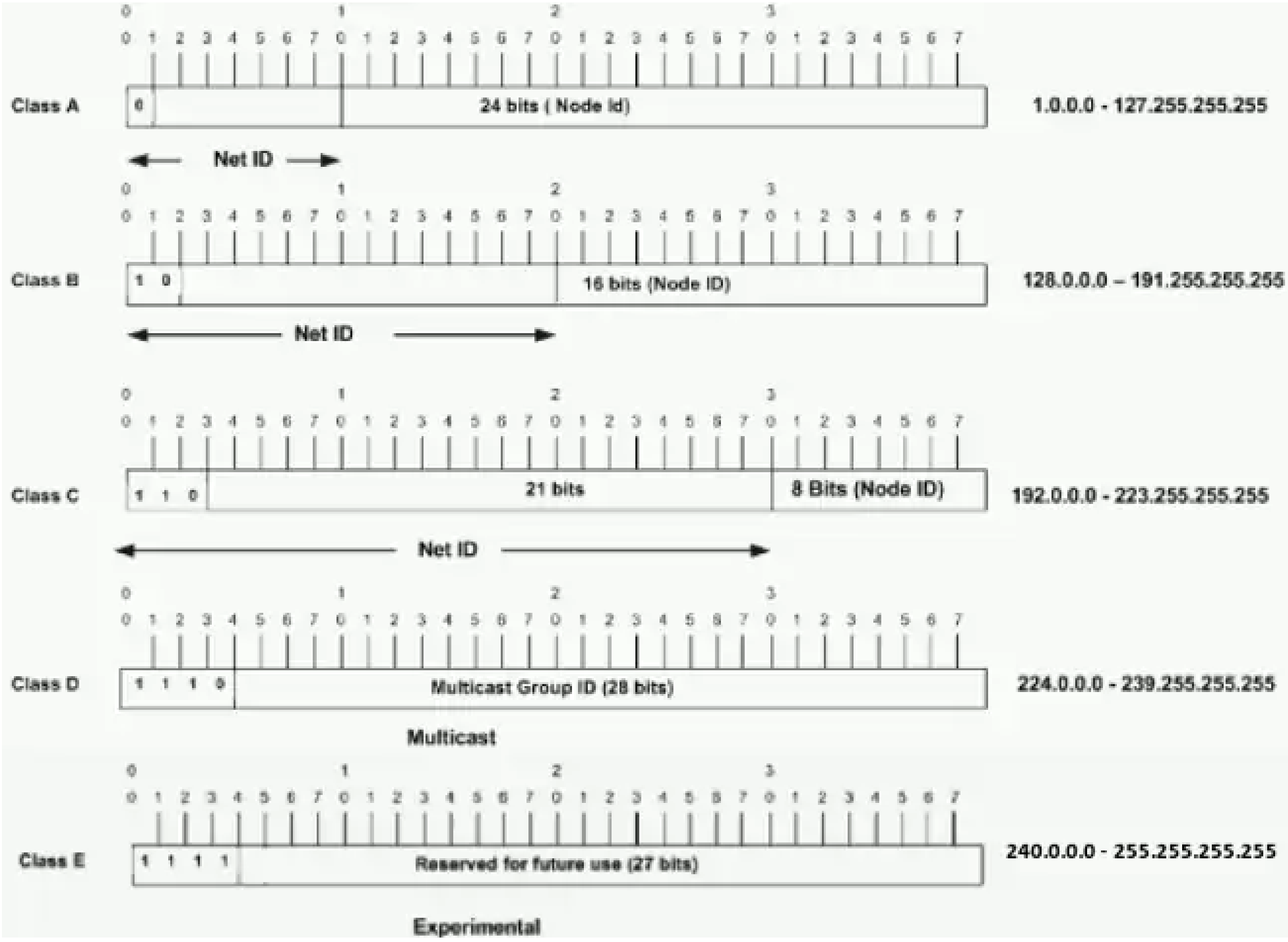
And this sample shows an IP address represented in both binary and decimal.

10. 1. 23. 19 (decimal)

00001010.00000001.00010111.00010011 (binary)

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

Given an IP address, its class can be determined from the three high-order bits (the three left-most bits in the first octet). [Figure 1](#) shows the significance in the three high order bits and the range of addresses that fall into each class. For informational purposes, Class D and Class E addresses are also shown.



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

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

In a Class C address, the first three octets are the network portion. The Class C example in Figure 1 has a major network address of 192.0.0.0 - 223.255.255.255. Octet 4 (8 bits) is for local subnets and hosts - perfect for networks with less than 254 hosts.

Network Masks

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

Class A: 255.0.0.0

Class B: 255.255.0.0

Class C: 255.255.255.0

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

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

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

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

----

net id | host id

netid = 00001000 = 8

hostid = 00010100.00001111.00000001 = 20.15.1

Understand Subnetting

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

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, that connects *n* networks/subnetworks has *n* distinct IP addresses, one for each network / subnetwork that it interconnects.

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

204.17.5.0 - 11001100.00010001.00000101.00000000

255.255.255.224 - 11111111.11111111.11111111.11100000

--|sub|------

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

204.17.5.0 255.255.255.224 host address range 1 to 30

204.17.5.32 255.255.255.224 host address range 33 to 62

204.17.5.64 255.255.255.224 host address range 65 to 94

204.17.5.96 255.255.255.224 host address range 97 to 126

204.17.5.128 255.255.255.224 host address range 129 to 158

204.17.5.160 255.255.255.224 host address range 161 to 190

204.17.5.192 255.255.255.224 host address range 193 to 222

204.17.5.224 255.255.255.224 host address range 225 to 254

Lab 8

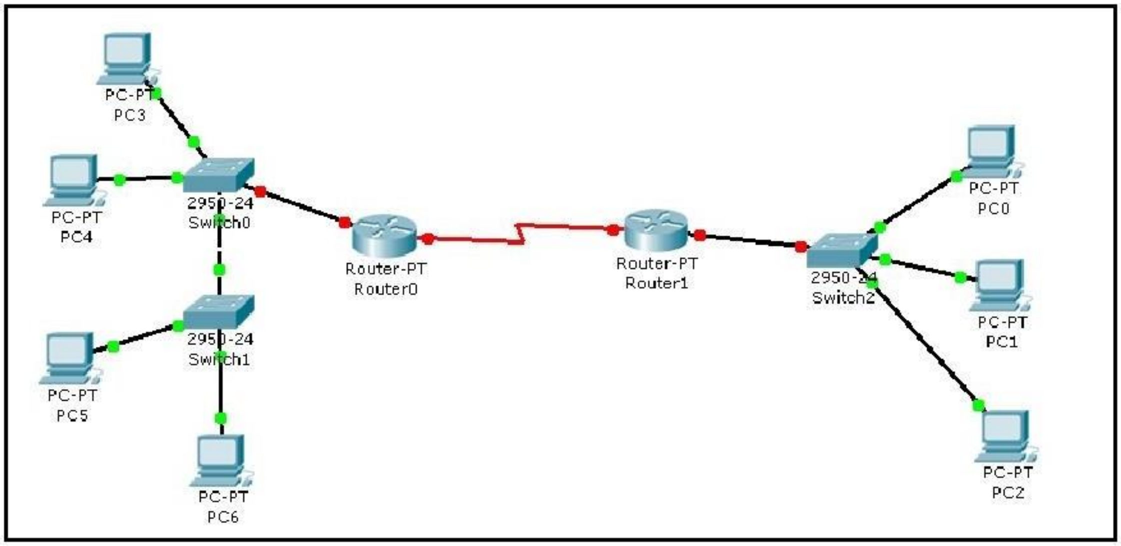
Dynamic Routing
(RIP, OSPF,EIGRP)

Using packet tracer in addressing :

1- We connect the figure .

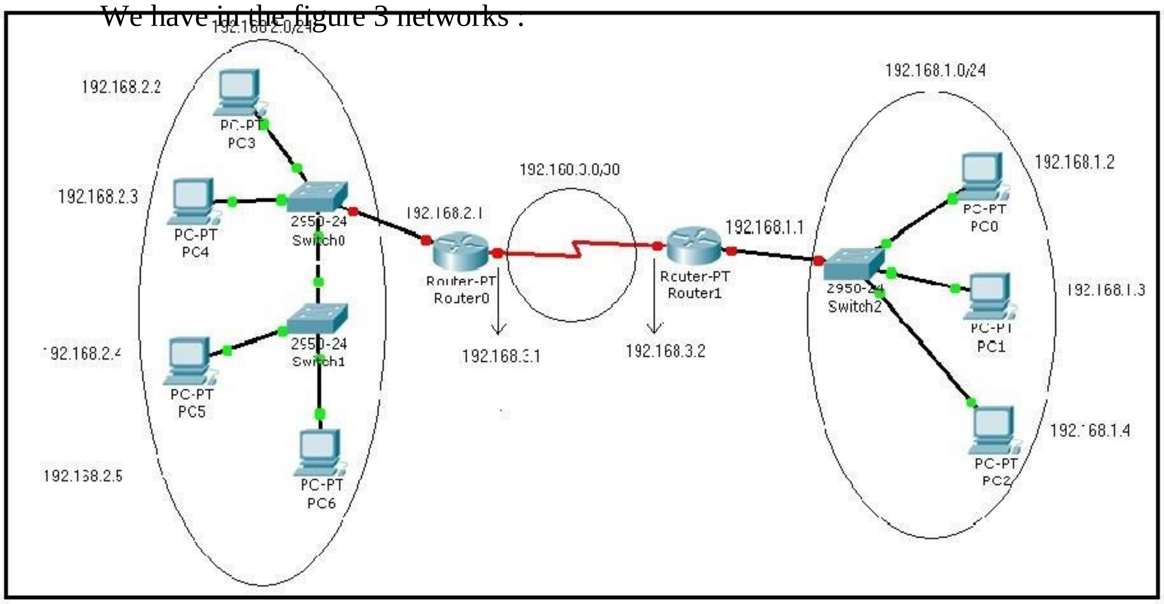
The cables used are :

- Straight : between switch, pc and router, switch.
- Cross : between switch, switch .
- Serial : betweeen the two routers .

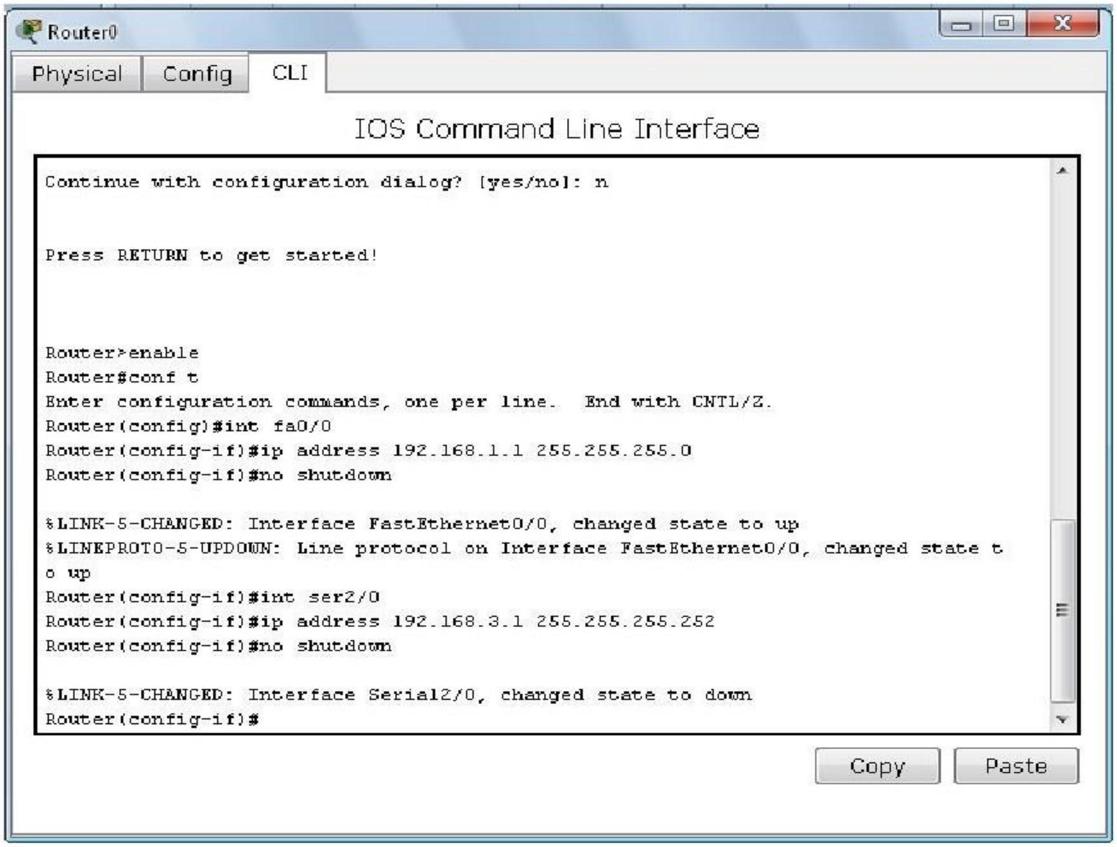


2- We put the IP address for each device as the following :

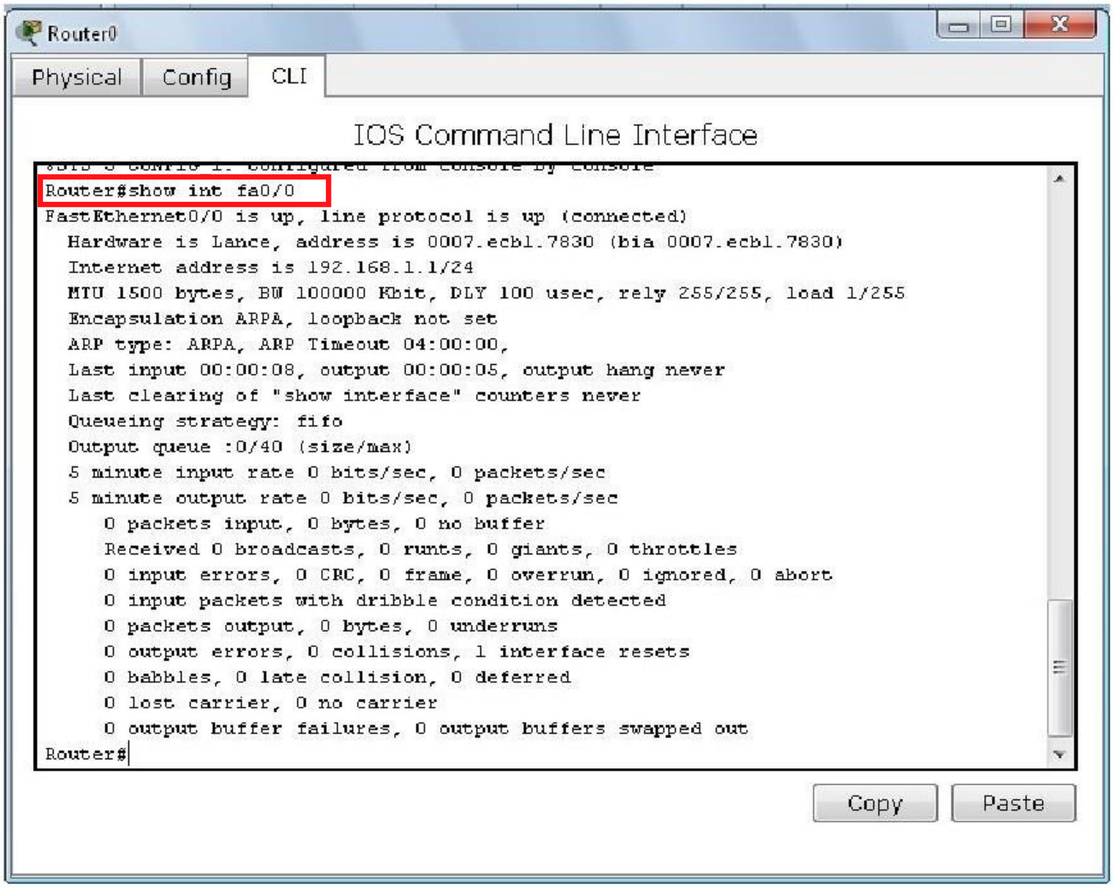
We have in the figure 3 networks :

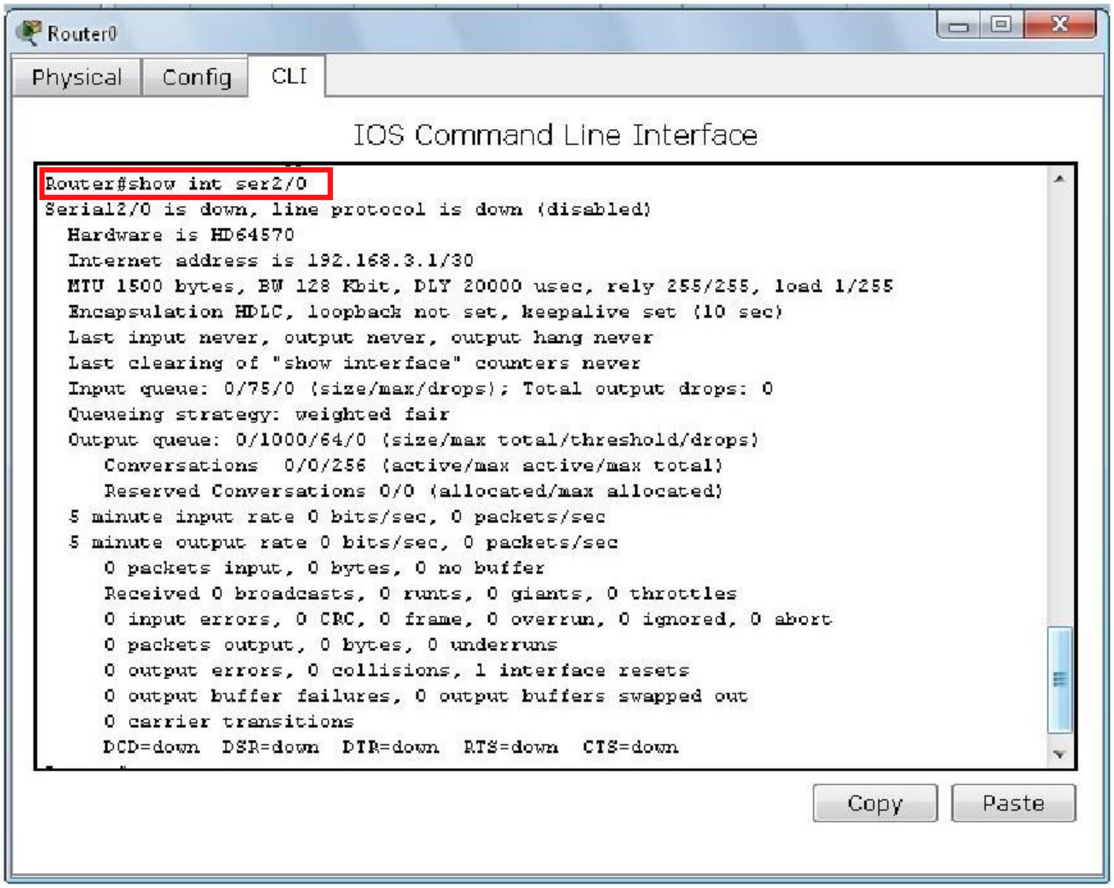


To disrtribute the Ip's on the router interfaces we do the following :

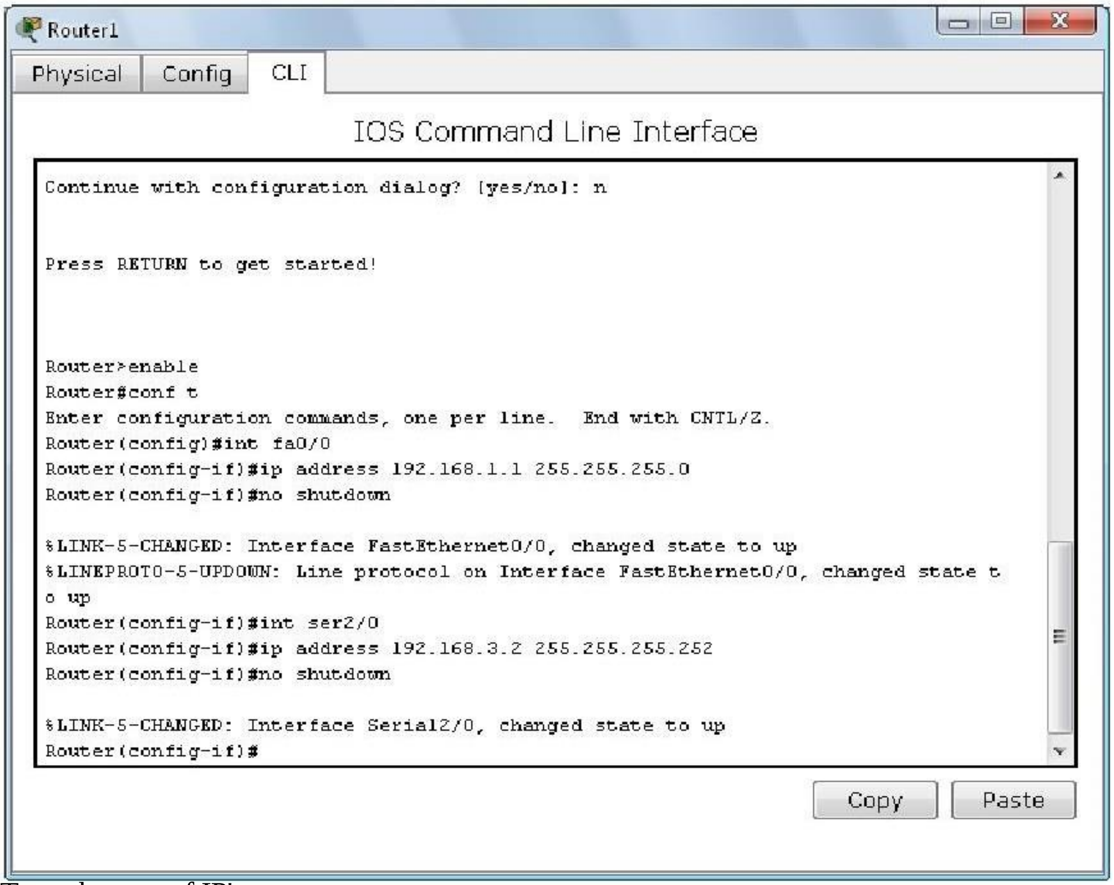


To make sure of the IP :

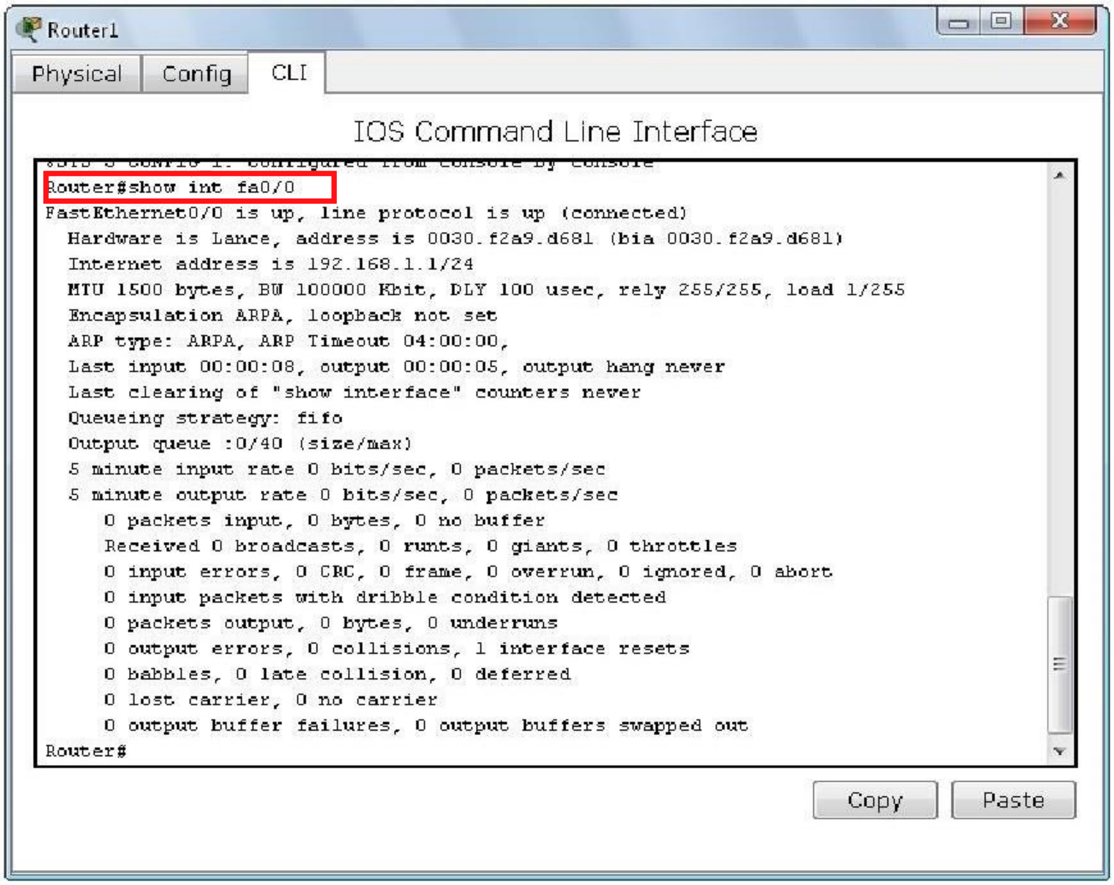

Router0
Physical
Config
CLI
IOS Command Line Interface
Router#show int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0007.ecb1.7830 (bia 0007.ecb1.7830)
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00,
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 input packets with dribble condition detected
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
Router#
Copy
Paste

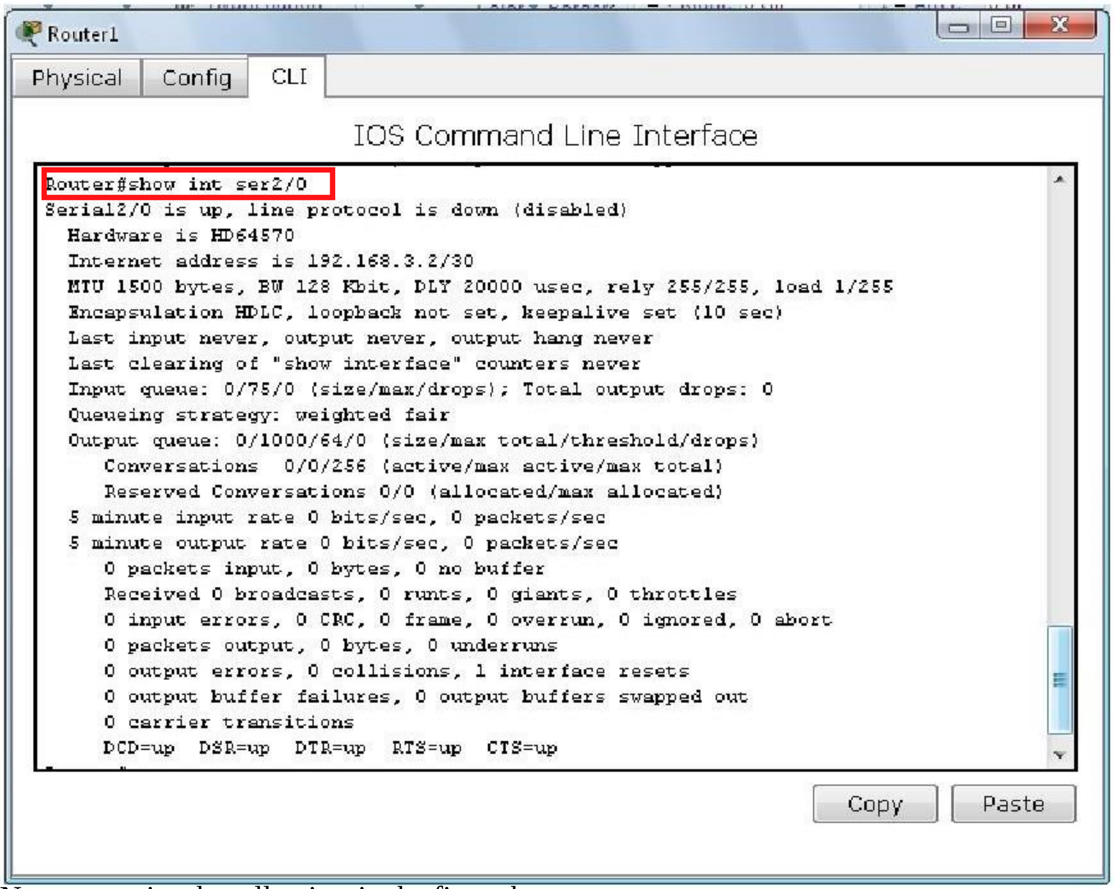


For the second router :

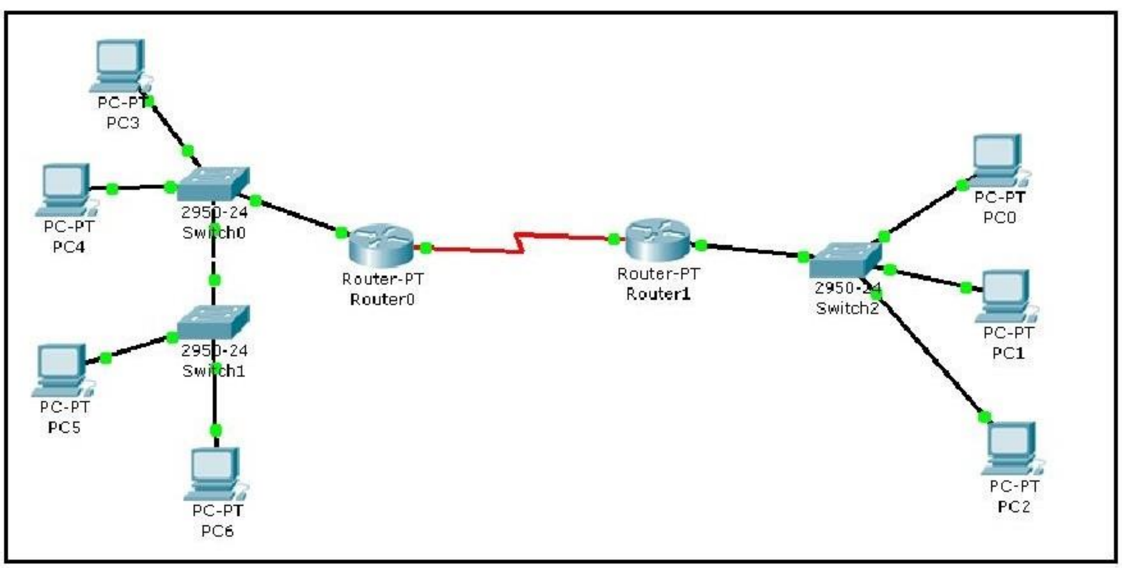


To make sure of IP's :

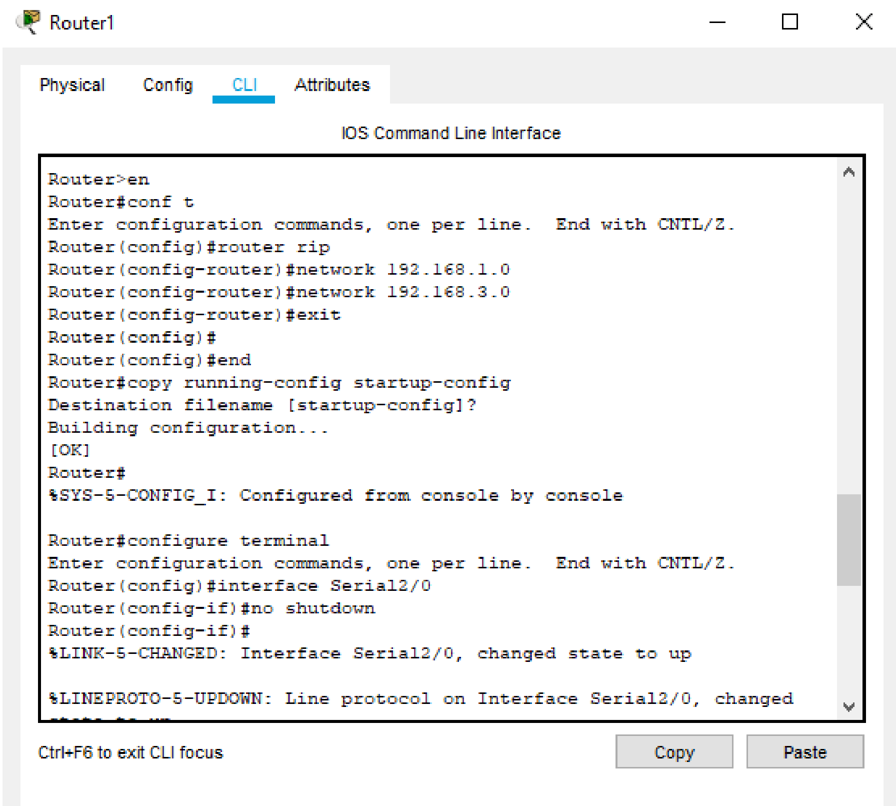

Router1
Physical
Config
CLI
IOS Command Line Interface
Router#show int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0030.f2a9.d681 (bia 0030.f2a9.d681)
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00,
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 input packets with dribble condition detected
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
Router#
Copy
Paste

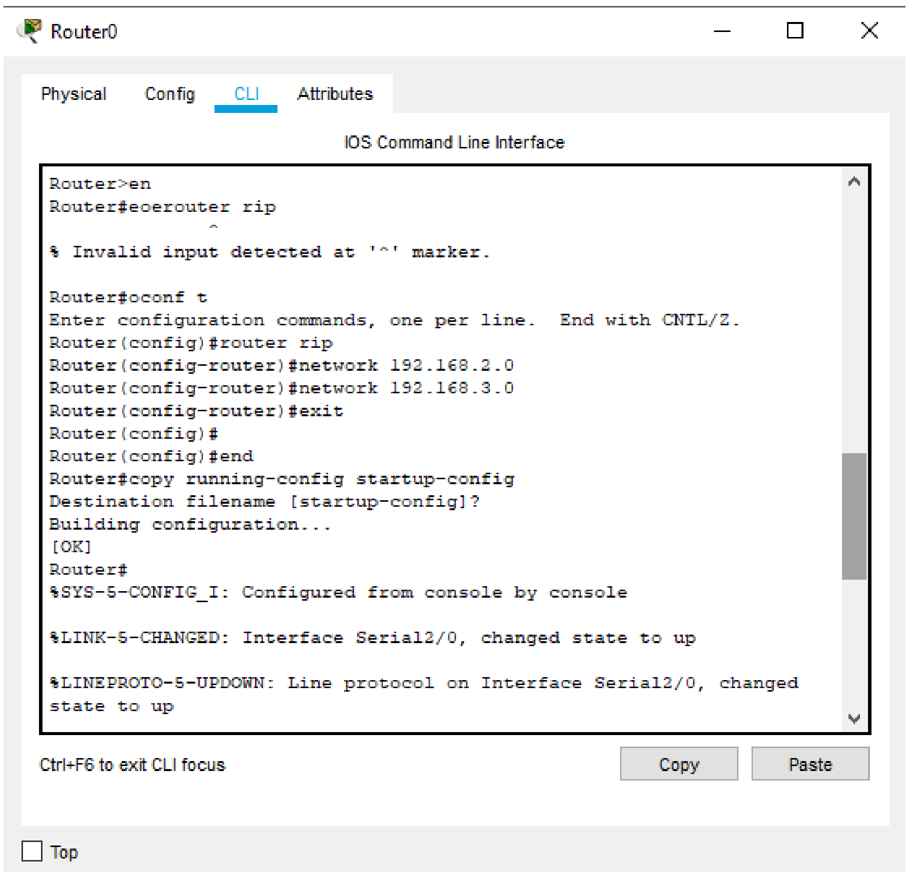


Now we notice that all points in the figure became green :

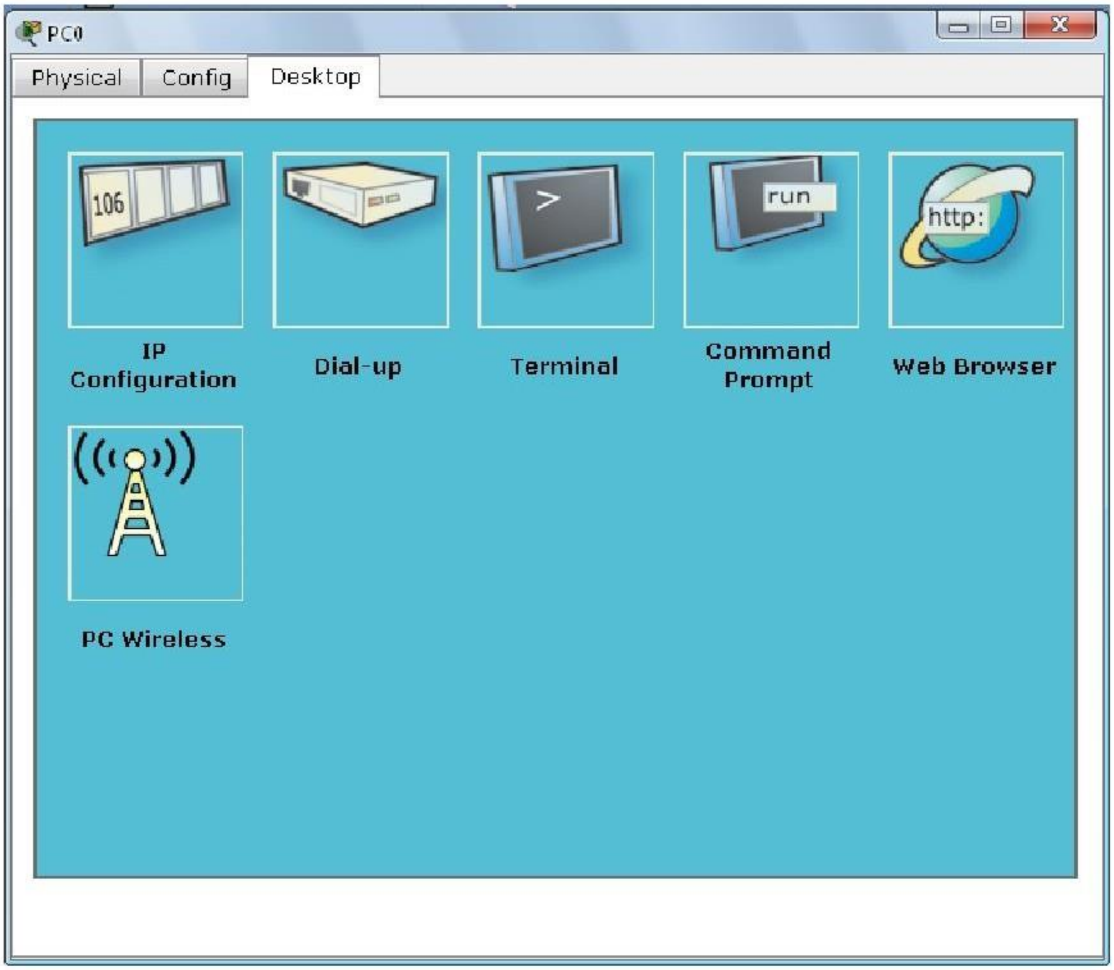


Now, configure RIP on both networks





How to ping on default gate way : For PC0



We chose command prompt and ping on the default gateway :

```
PC>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=63ms TTL=255
Reply from 192.168.1.1: bytes=32 time=62ms TTL=255
Reply from 192.168.1.1: bytes=32 time=47ms TTL=255
Reply from 192.168.1.1: bytes=32 time=60ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 47ms, Maximum = 63ms, Average = 58ms
```

Task:

Configure dynamic routing for the above topology using **RIP**

Now ping all the interfaces. Were all the ping’s successful….?

Lab 9 **CLO4**

Router Configurations

(Access Control Lists)

Source: www.cisco.com

Access Control Lists

- Access Control Lists used to implement security in routers
  - powerful tool for network control
  - filter packets flow in or out of router interfaces
  - restrict network use by certain users or devices
  - deny or permit traffic

- Rules Followed When Traffic Is Compared To An Access Control List
  - Is done in sequential order; line 1, line 2, line 3 e.t.c
  - Is compared with the access list until a match is made; then NO further comparisons are made
  - There is an implicit "deny" at the end of each access list; if a packet does not match in the access list, it will be discarded

- Using Access Control Lists
  - Standard IP Access Lists (1 - 99)
    - simpler address specifications
    - generally permits or denies entire protocol suite
  - Extended IP Access Lists (100 - 199)
    - more complex address specification
    - generally permits or denies specific protocols

- Syntax of using access lists
  - Standard IP Access List Configuration Syntax
    - access-list access-list-number {permit | deny} source {source-mask}
    - ip access-group access-list-number {in | out}
  - Extended IP Access List Configuration Syntax
    - access-list access-list-number {permit | deny} protocol source {source-mask} destination {destination-mask}
    - ip access-group access-list-number {in | out}

- <u>Where to place access control lists</u>
    - Place **Standard IP** access list close to **destination**
    - Place **Extended IP** access lists close to the **source** of the traffic you want to manage

- <u>Using Wild Cards</u>
    - Are used with access lists to specify a host, network or part of a network
    - To specify an address range, choose the next largest block size e.g.
        - to specify 34 hosts, you need a 64 block size
        - to specify 18 hosts, you need a 32 block size
        - to specify 2 hosts, you need a 4 block size

- <u>Wild Card Masks</u>
    - Are used with the host/network address to tell the router a range of addresses to filter
        - Examples:
        - to specify a host: 81.199.108.1 0.0.0.0
        - to specify a small subnet: 81.199.108.8 – 81.199.108.15 (would be a /29)
        - Block size is 8, and wildcard is always one number less than the block size
        - Cisco access list then becomes: 81.199.108.8 0.0.0.7
    - to specify all hosts on a Class C network: 81.199.108.0 0.0.0.255

- **<u>What are wild card masks</u>**
    - Short cut method to a quick calculation of a network subnet to wildcard:
        - 255 – {netmask bits on subnet mask}
        - to create wild card mask for 81.199.108.160 255.255.255.240
        - 81.199.108.160 0.0.0.15 {255 – 240}
        - to create wild card mask for 81.199.108.0 255.255.252.0
          81.199.108.0 0.0.3.255

- **<u>Examples</u>**
    - Router(config)#Access-list access-list-number {permit|deny}{test conditions}
    - Router(config)#{protocol} access-group access-list-number
        - e.g check for IP subnets 81.199.108.80 to 81.199.108.95
          81.199.108.80,
          Address and Wildcard Mask: **81.199.108.80 0.0.0.15**
    - Wildcard bits indicate how to check corresponding address bit
        - 0=check or match
        - 1=ignore
    - Matching Any IP Address

- 255.255.255.255
- or abbreviate the expression using the keyword any

o Matching a specific host

- 81.199.108.8 0.0.0.0
- or abbreviate the wildcard using the IP address preceded by the keyword host

Lab Task

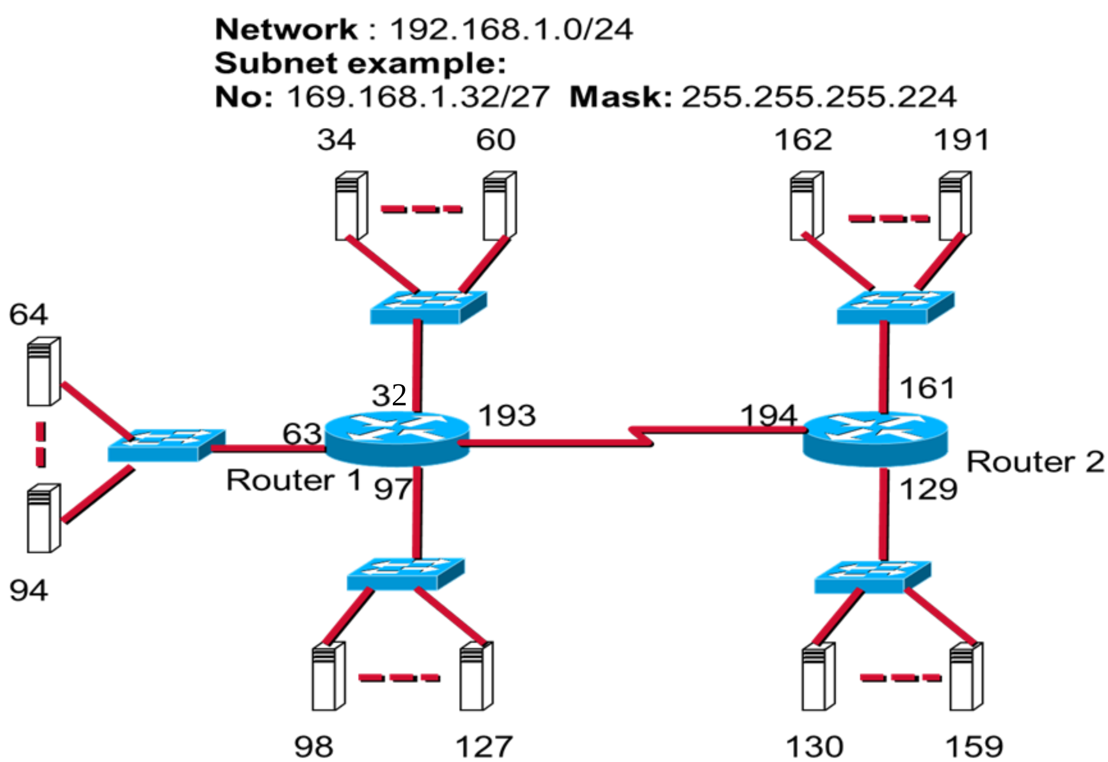
Task 1:

Use the topology created in last lab and do following:

1- attach one webserver on each network
2- assign it a valid ip address from the pool and then deny one hosts of the other network to have an access to this webserver.
3- Do similar for the other network

Task 2:

Considering Network designed



Implement the following access control lists on the two available routers:

Create an access list to deny hosts of network 192.168.1.32 to restrict the access to the other two networks 192.168.1.64 and 192.168.1.96
(config)#access-list 1 deny 192.168.1. 32 0.0.0.31

```
access-list 1 permit any any
  Interface fa0/1
  ip access-group 1 out
  Interface fa1/0
  ip access-group 1 out
```

Objective:

Introduction to OMNET++ and Inet

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. OMNeT++ is distributed under the Academic Public License.

OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

Components:

The main ingredients of OMNeT++ are:

- Simulation kernel library (C++)
- The NED topology description language
- Simulation IDE based on the Eclipse platform
- Interactive simulation runtime GUI (Qtenv)
- Command-line interface for simulation execution (Cmdenv)
- Utilities (makefile creation tool, etc.)
- Documentation, sample simulations, etc.

Models:

During the years OMNeT++ has been available, countless simulation models and model frameworks have been written for it by researchers in diverse areas: queuing, resource modeling, internet protocols, wireless networks, switched LANs, peer-to-peer networks, media streaming, mobile ad-hoc networks, mesh networks, wireless sensor networks, vehicular networks, NoCs, optical networks, HPC systems, cloud computing, SANs, and more. Most of these model frameworks are open source, developed as independent projects, and follow their own release cycles.

The INET Framework can be considered the standard protocol model library of OMNeT++. INET contains models for the Internet stack and many other protocols and components. The INET Framework is maintained by the OMNeT++ team for the community, utilizing patches and new models contributed by members of the community. Several other simulation frameworks take INET as a base, and extend it into specific directions, such as vehicular networks (Veins, CoRE), overlay/peer-to-peer networks (OverSim), or LTE (SimuLTE).

Platforms:

The OMNeT++ simulation kernel is standard C++ and runs basically on all platforms where a modern C++ compiler is available. The Simulation IDE requires Windows, Linux, or macOS.

Installation

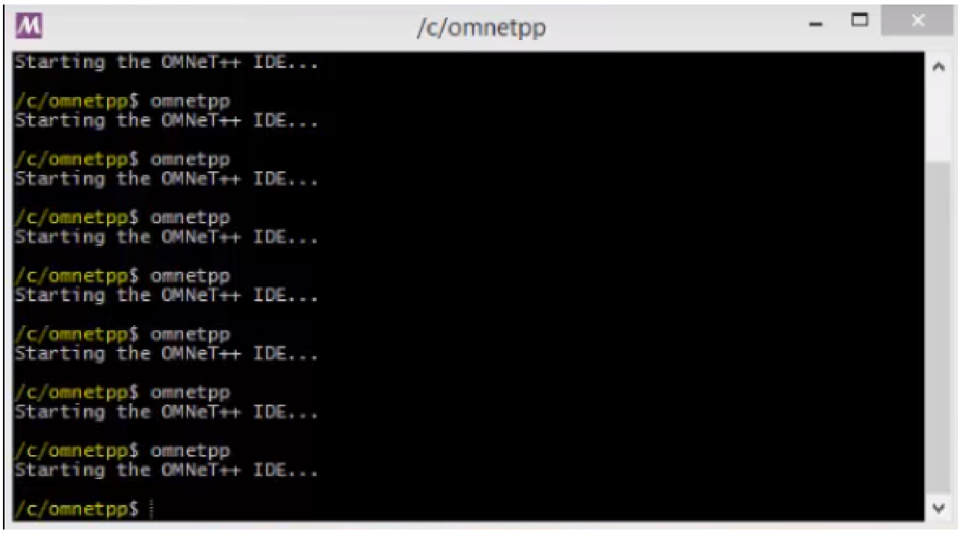
There are several ways to install the INET Framework:

- Let the OMNeT++ IDE download and install it for you. This is the easiest way. Just accept the offer to install INET in the dialog that comes up when you first start the IDE or choose Help ‣ Install Simulation Models any time later.
- From INET Framework web site, http://inet.omnetpp.org. The IDE always installs the last stable version compatible with your version of OMNeT++. If you need some other version, they are available for download from the web site. Installation instructions are also provided there.
- From GitHub. If you have experience with git, clone the INET Framework project (inet-framework/inet), check out the revision of your choice, and follow the INSTALL file in the project root.

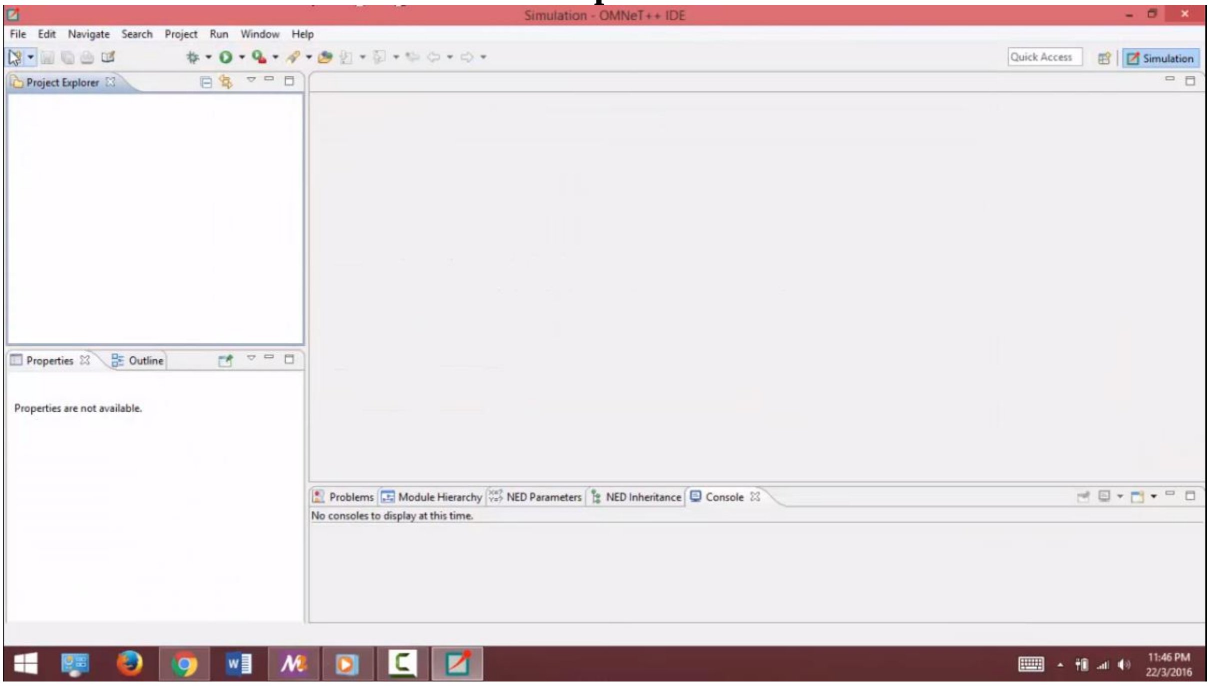
First project on OMNET++

To create a project in OMNET++ follow the given steps:

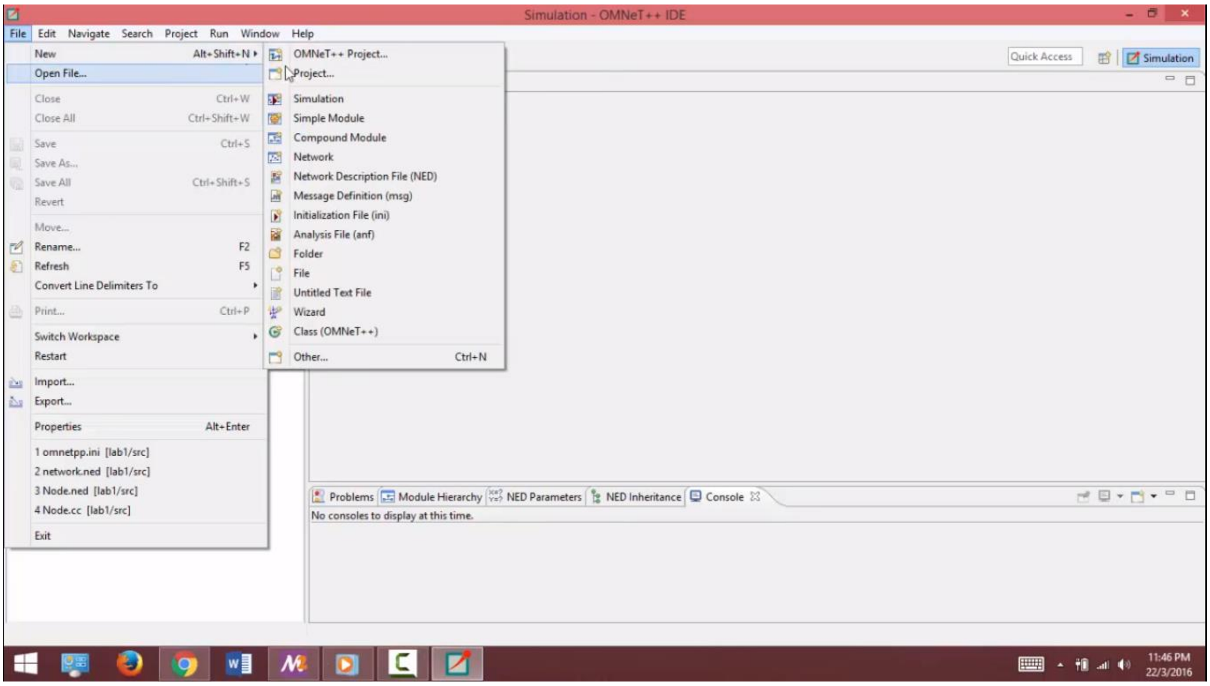
Write omnetpp in the cmd of omnet to launch the interface



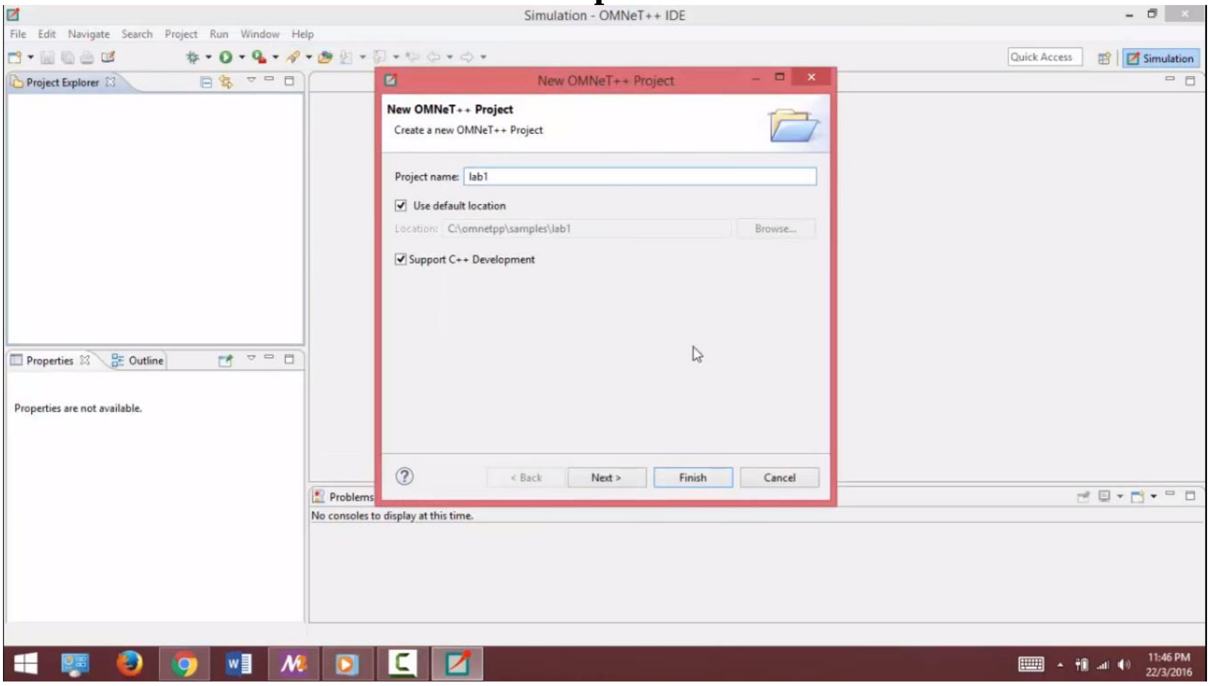
This is the interface of OMNET++



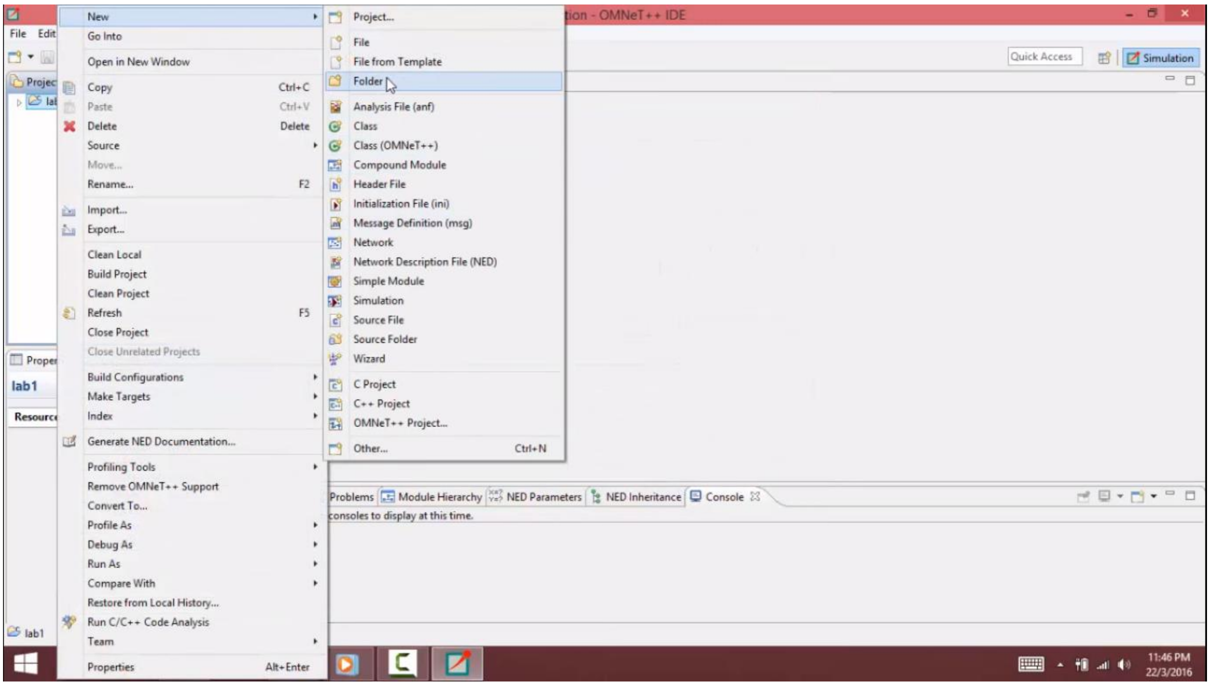
Select file->open file-> OMNet++ Project



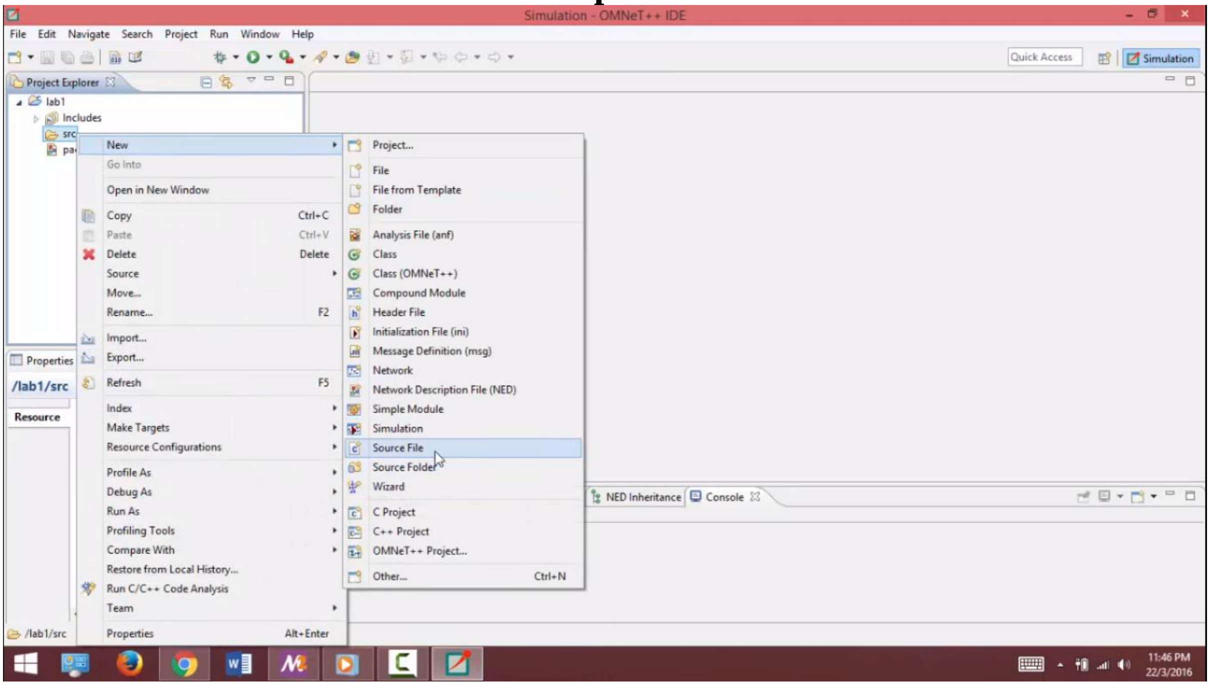
Write name of your project and click Next and then Finish



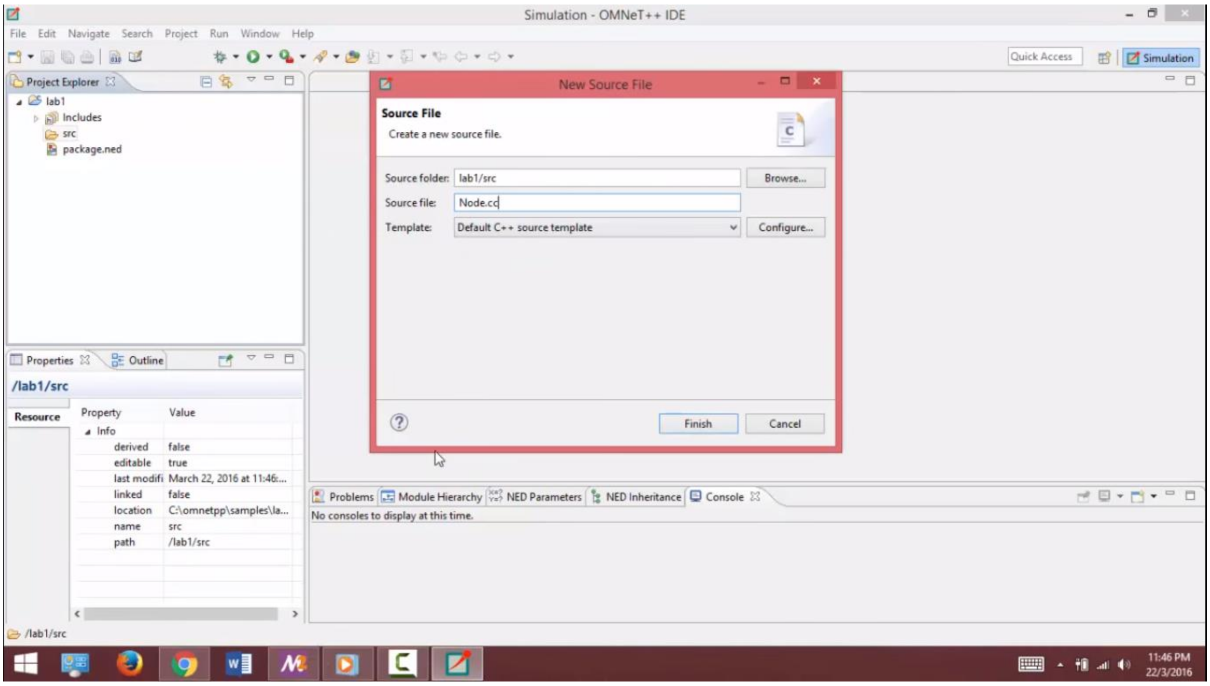
Right click on the name of project and make a new folder, give it a name and click Finish.



Expand the project select the folder and create new source file

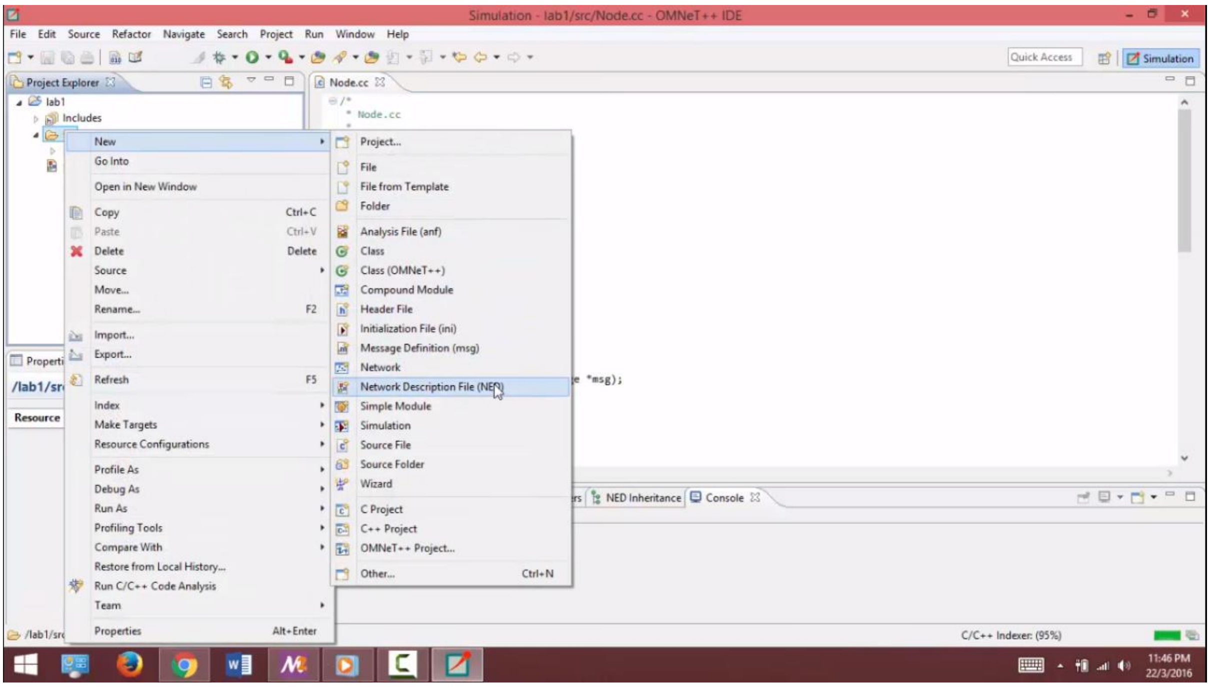


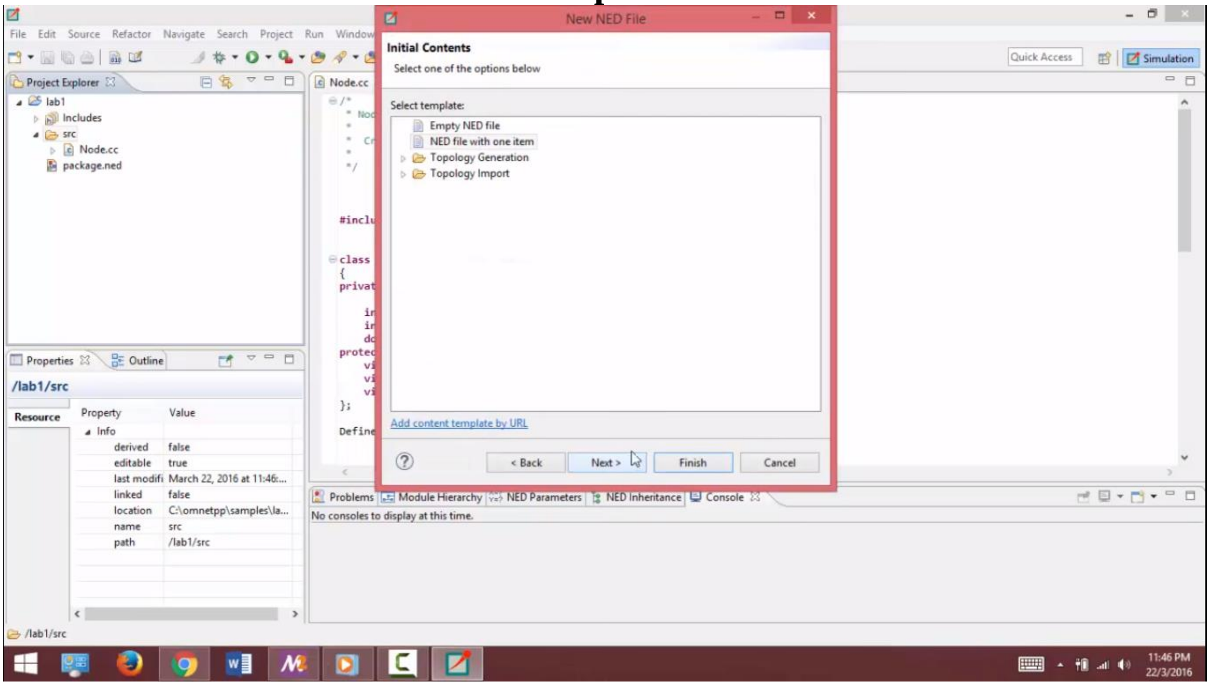
Name the file with extension of *.cc* and click Finish.

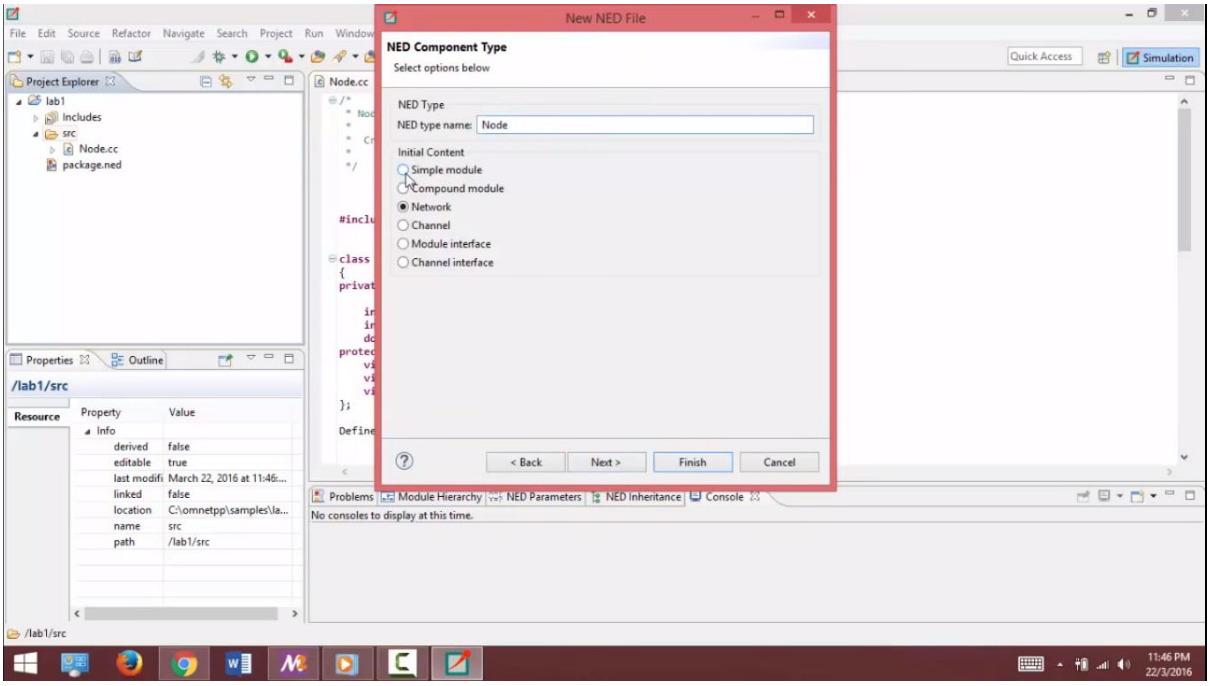


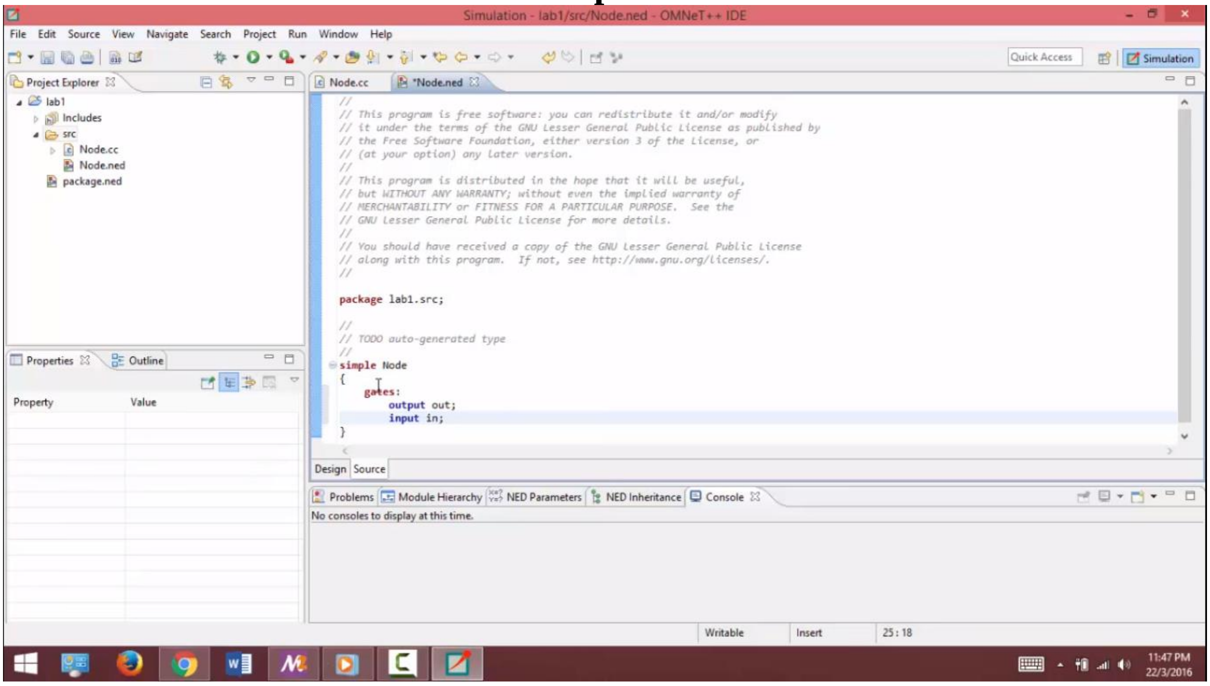
Click on package.ned and add following code there

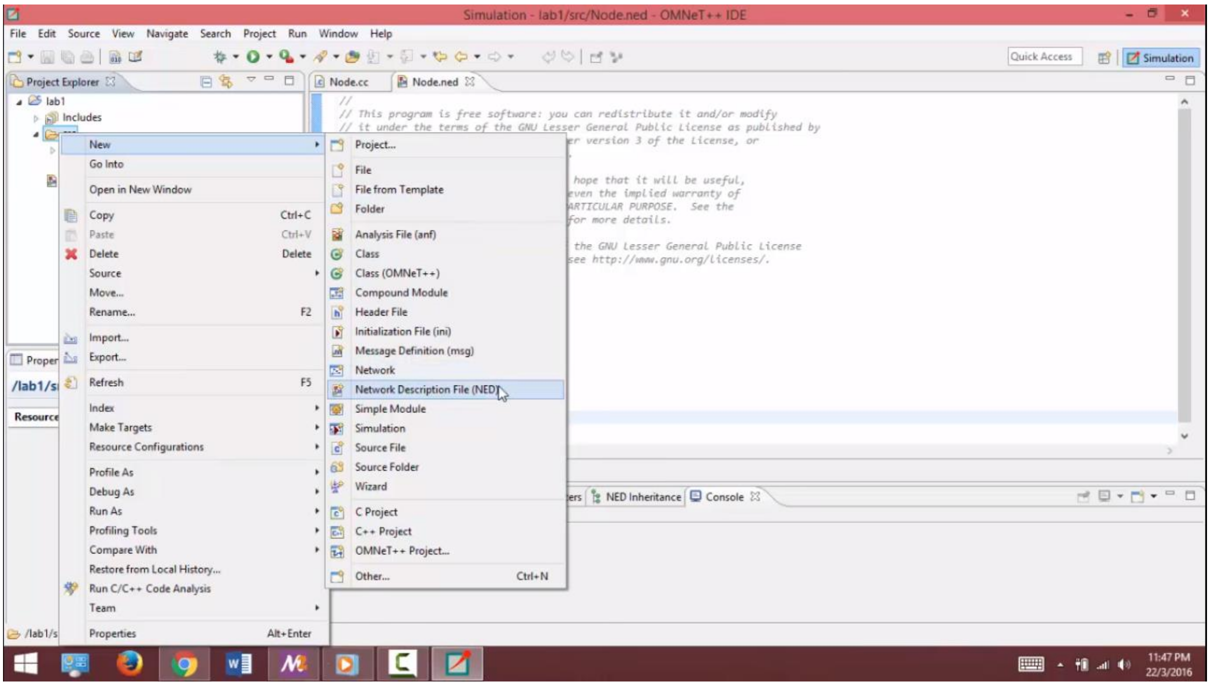
```cpp
#include<omnetpp.h>

class Node:public cSimpleModule
{
private:

    int no_sent;
    int no_rcvd;
    double time_interval;
protected:
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
    virtual void finish();
};

Define_Module(Node);

void Node::initialize()
{

    no_sent= 0;
    no_sent= 0;
    time_interval= 0.1;

    cMessage *msg= new cMessage();
    scheduleAt(0.01,msg);


}
void Node::handleMessage(cMessage *msg)
{

    if(msg->isSelfMessage())
    {
    cMessage *out_msg= new cMessage();
    send(out_msg,"out");
    no_sent++;
    scheduleAt(simTime()+time_interval,msg);

    }
    else
    {
        no_rcvd++;
        delete(msg);
    }

}
void Node::finish()
{

    recordScalar("Number of received messages",no_rcvd);
    recordScalar("Number of sent messages",no_sent);
}
```

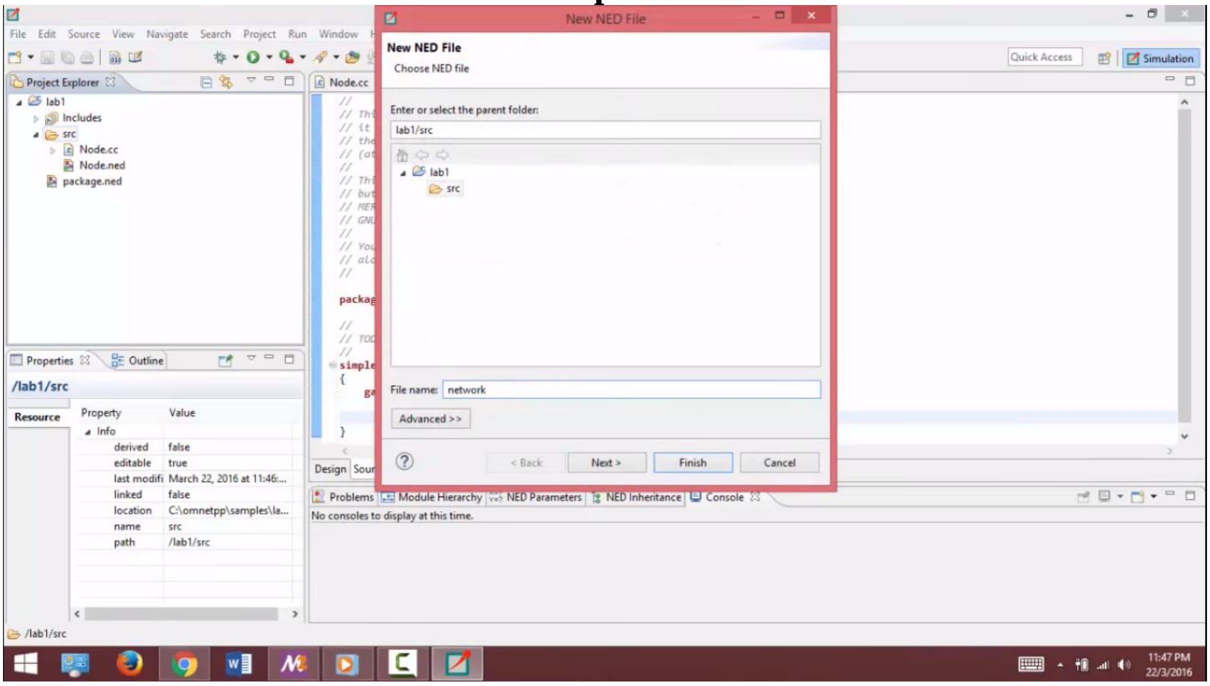


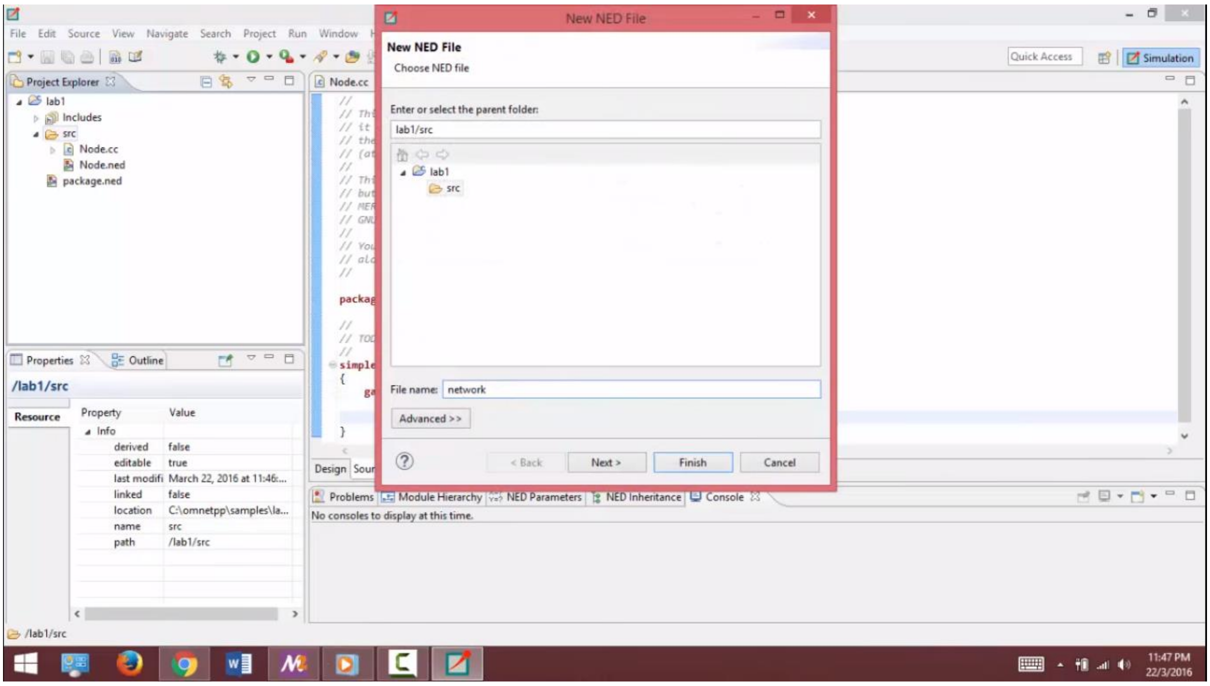

New NED File
Initial Contents
Select one of the options below
Select template:
Empty NED file
NED file with one item
Topology Generation
Topology Import
Add content template by URL
< Back
Next >
Finish
Cancel
Project Explorer
lab1
Includes
src
Node.cc
package.ned
Properties
Outline
/lab1/src
Resource
Property
Value
Info
derived
false
editable
true
linked
false
location
C:\omnetpp\samples\la...
name
src
path
/lab1/src
Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
No consoles to display at this time.
Quick Access
Simulation

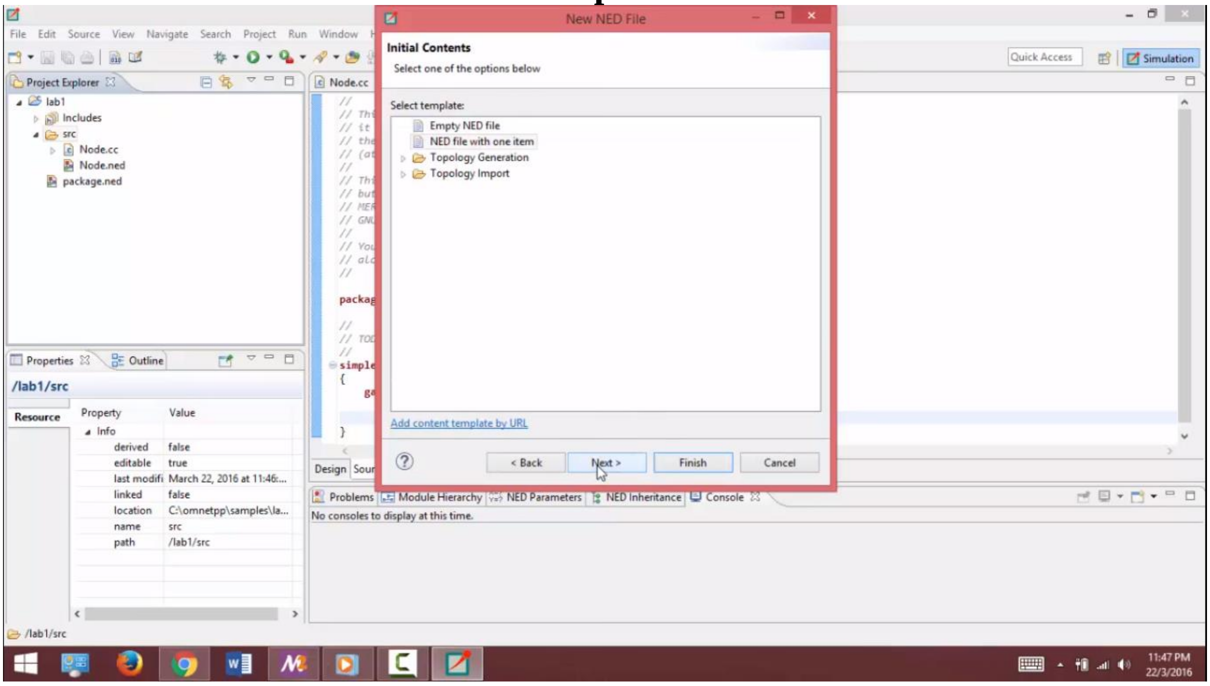

New NED File
NED Component Type
Select options below
NED Type
NED type name:
Node
Initial Content
Simple module
Compound module
Network
Channel
Module interface
Channel interface
< Back
Next >
Finish
Cancel
Project Explorer
lab1
Includes
src
Node.cc
package.ned
Properties
Outline
/lab1/src
Resource
Property
Value
Info
derived
false
editable
true
linked
false
location
C:\omnetpp\samples\la...
name
src
path
/lab1/src
Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
No consoles to display at this time.
Quick Access
Simulation

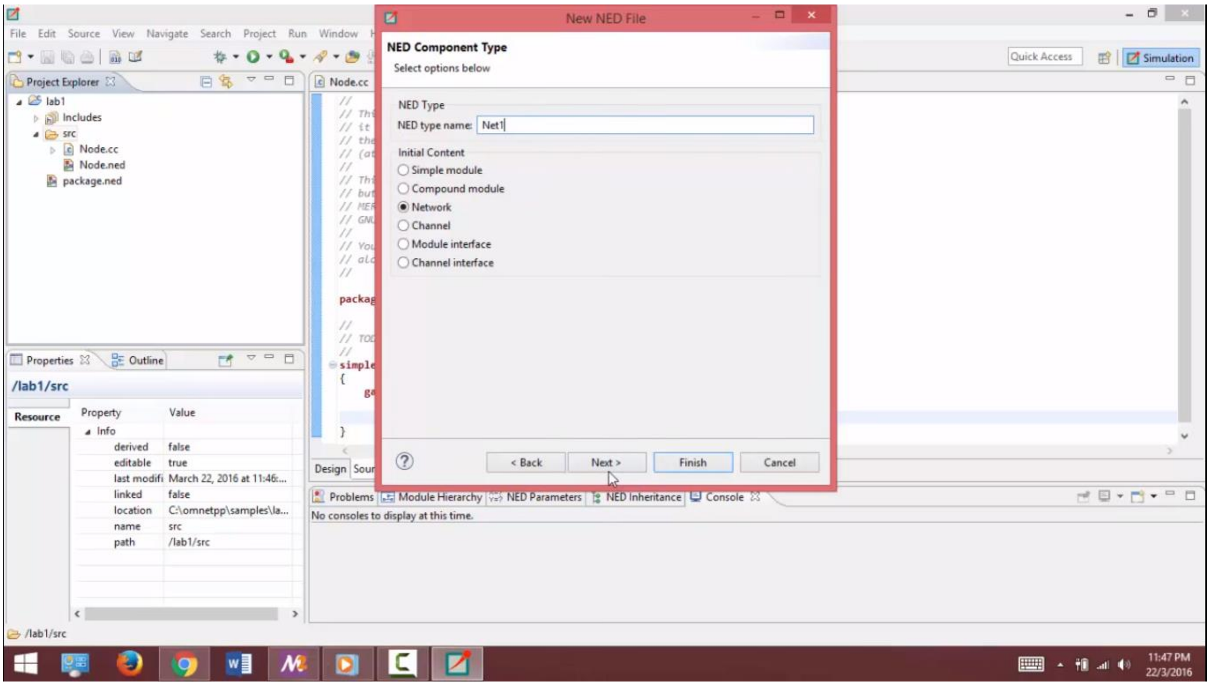

Simulation - lab1/src/Node.ned - OMNeT++ IDE
File Edit Source View Navigate Search Project Run Window Help
Project Explorer
lab1
Includes
src
Node.cc
Node.ned
package.ned
Node.cc
*Node.ned
package lab1.src;
// TODO auto-generated type
simple Node
{
gates:
output out;
input in;
}
Design Source
Problems Module Hierarchy NED Parameters NED Inheritance Console
No consoles to display at this time.
Writable Insert 25 : 18


Simulation - lab1/src/Node.ned - OMNeT++ IDE
New
Go Into
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Refresh F5
Index
Make Targets
Resource Configurations
Profile As
Debug As
Run As
Profiling Tools
Compare With
Restore from Local History...
Run C/C++ Code Analysis
Team
Properties Alt+Enter
Project...
File
File from Template
Folder
Analysis File (anf)
Class
Class (OMNeT++)
Compound Module
Header File
Initialization File (ini)
Message Definition (msg)
Network
Network Description File (NED)
Simple Module
Simulation
Source File
Source Folder
Wizard
C Project
C++ Project
OMNeT++ Project...
Other... Ctrl+N

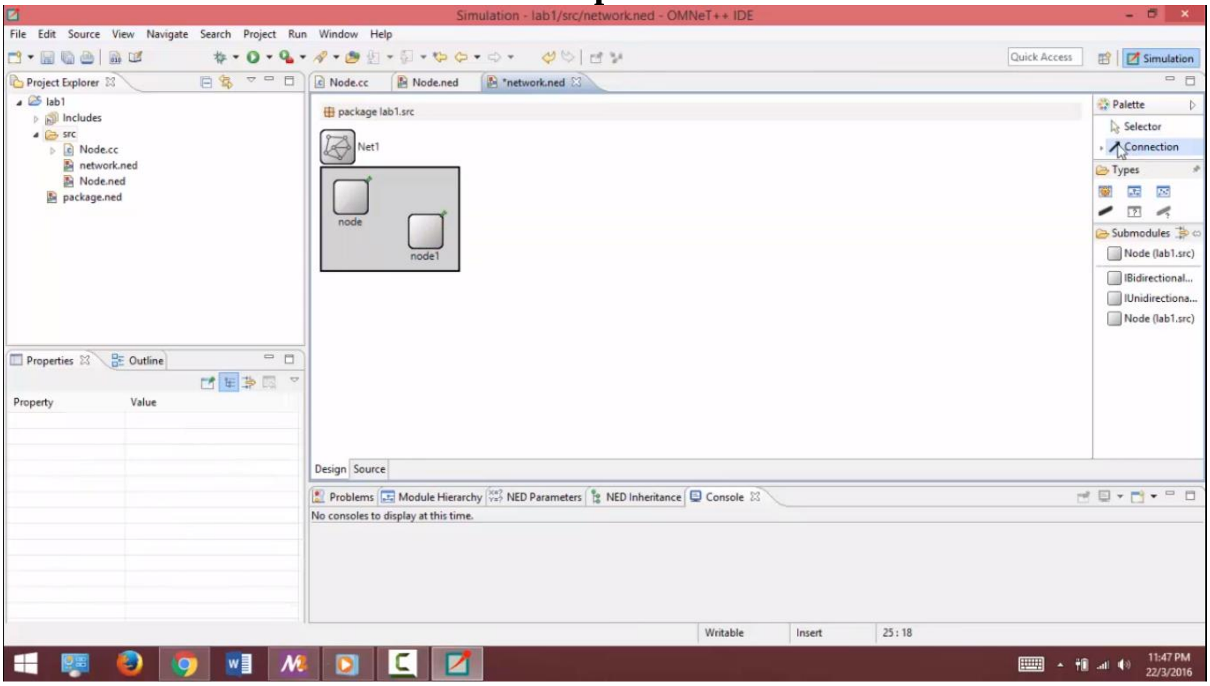

New NED File
Choose NED file
Enter or select the parent folder:
lab1/src
File name: network
Advanced >>
< Back
Next >
Finish
Cancel

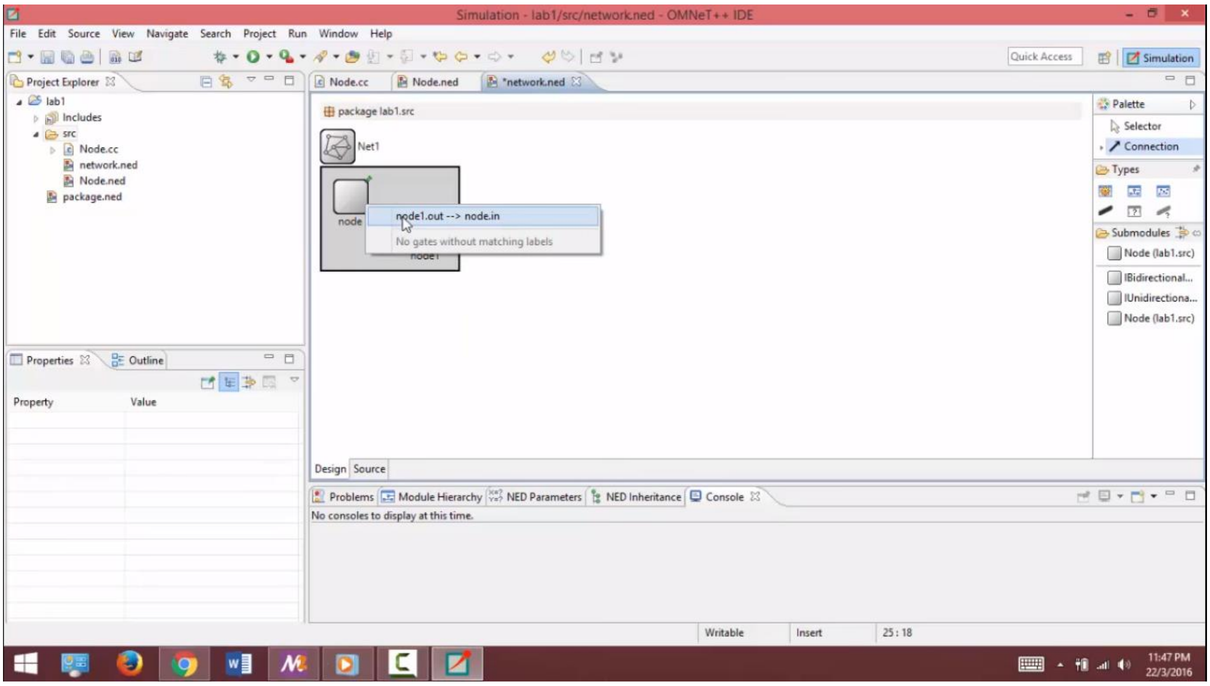

New NED File
Choose NED file
Enter or select the parent folder:
lab1/src
File name: network
Advanced >>
< Back
Next >
Finish
Cancel



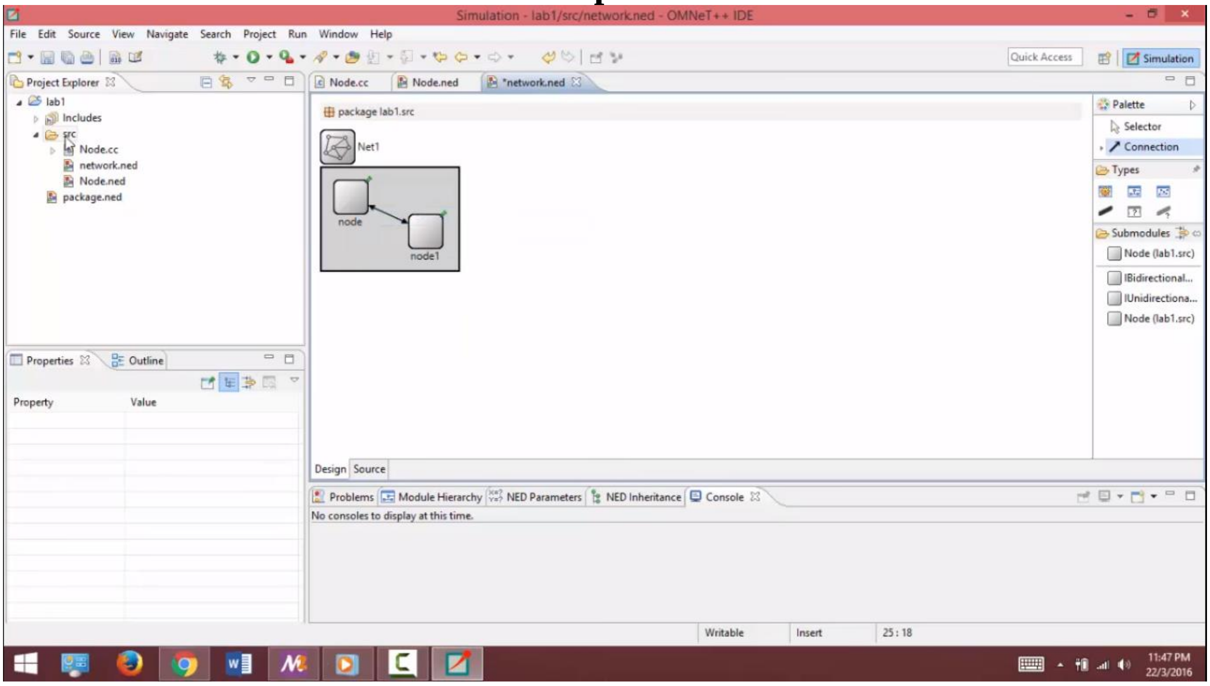


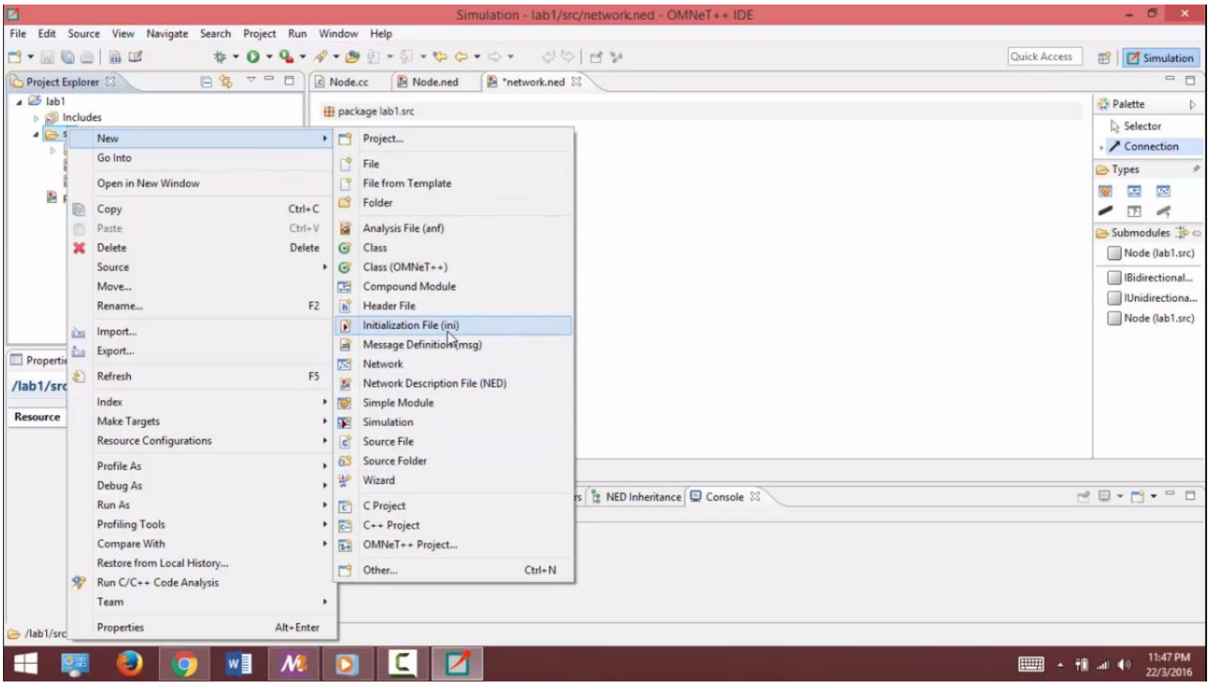
Drag Node(lab1.src) (twice) into the network and connect them both ways with the connection as shown below.

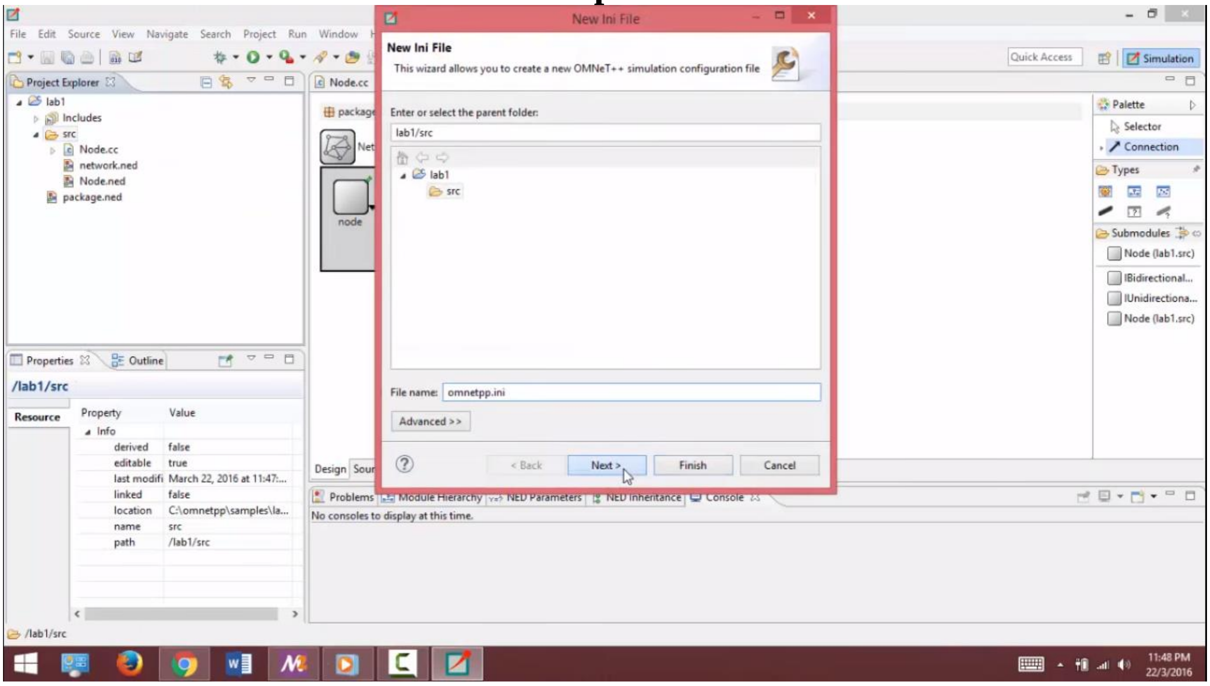

Simulation - lab1/src/network.ned - OMNeT++ IDE
File Edit Source View Navigate Search Project Run Window Help
Quick Access
Simulation
Project Explorer
lab1
Includes
src
Node.cc
network.ned
Node.ned
package.ned
Node.cc
Node.ned
*network.ned
package lab1.src
Net1
node
node1
Palette
Selector
Connection
Types
Submodules
Node (lab1.src)
IBidirectional...
IUnidirectiona...
Node (lab1.src)
Properties
Outline
Property
Value
Design
Source
Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
No consoles to display at this time.
Writable
Insert
25 : 18
11:47 PM
22/3/2016

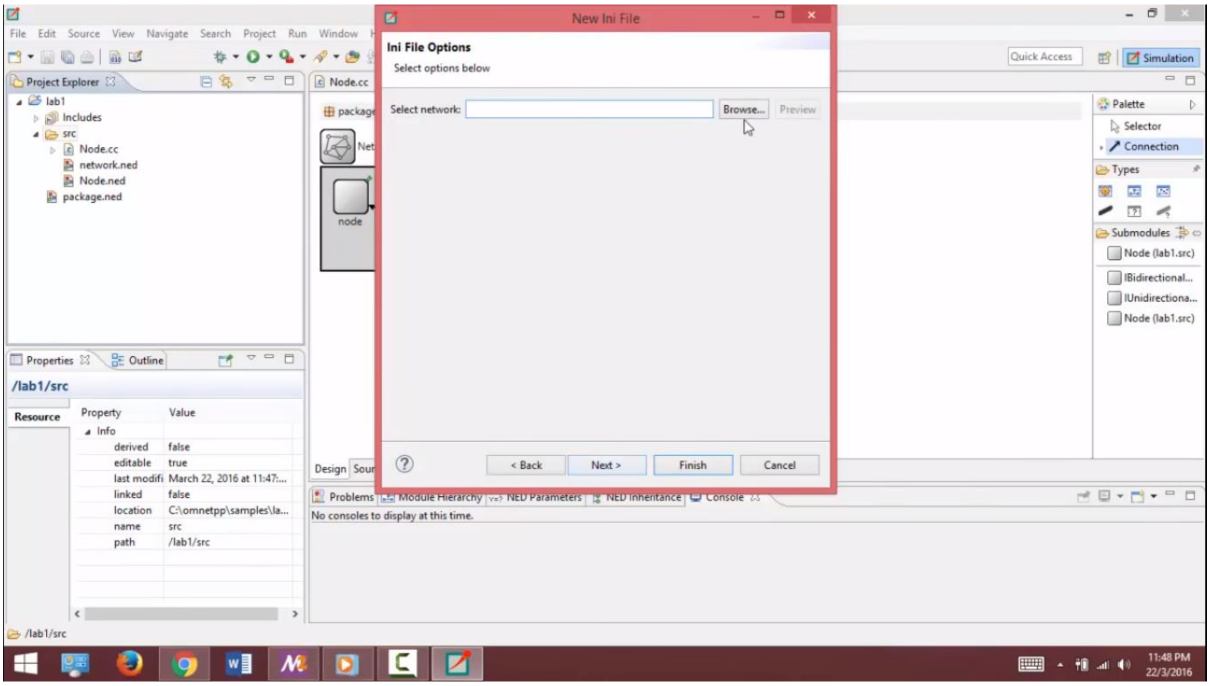

Simulation - lab1/src/network.ned - OMNeT++ IDE
File Edit Source View Navigate Search Project Run Window Help
Quick Access
Simulation
Project Explorer
lab1
Includes
src
Node.cc
network.ned
Node.ned
package.ned
Node.cc
Node.ned
*network.ned
package lab1.src
Net1
node
node1.out --> node.in
No gates without matching labels
Palette
Selector
Connection
Types
Submodules
Node (lab1.src)
IBidirectional...
IUnidirectiona...
Node (lab1.src)
Properties
Outline
Property
Value
Design
Source
Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
No consoles to display at this time.
Writable
Insert
25 : 18
11:47 PM
22/3/2016

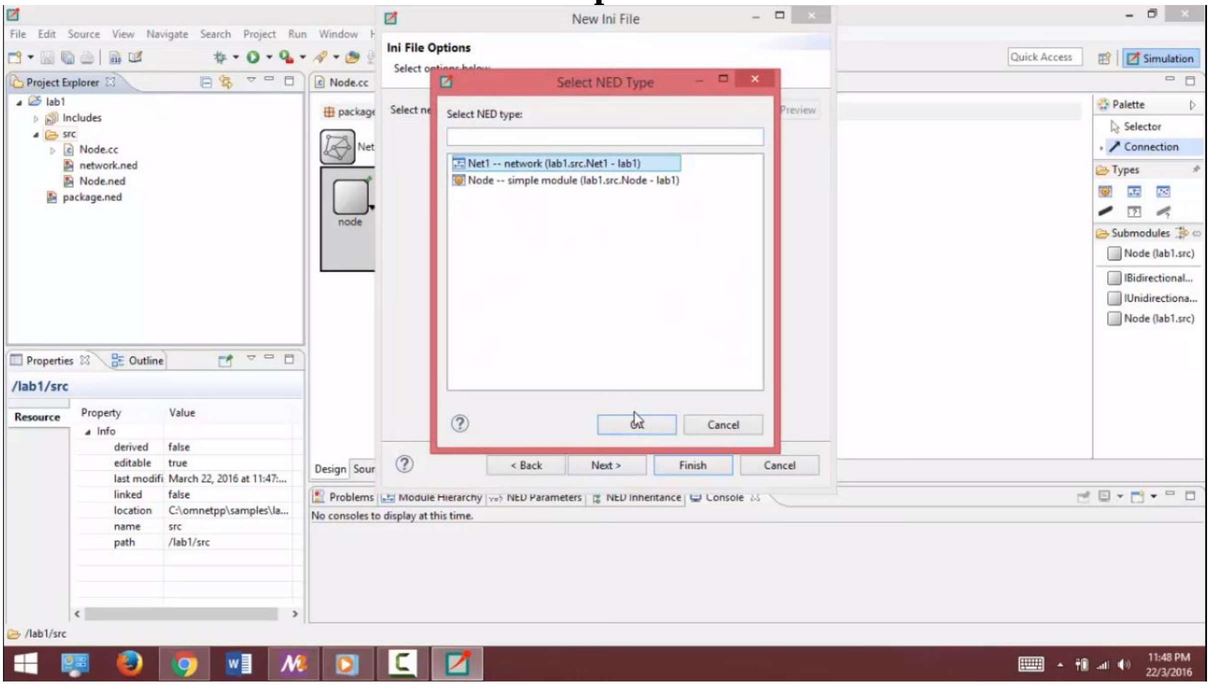
CSE-331 Computer Networks



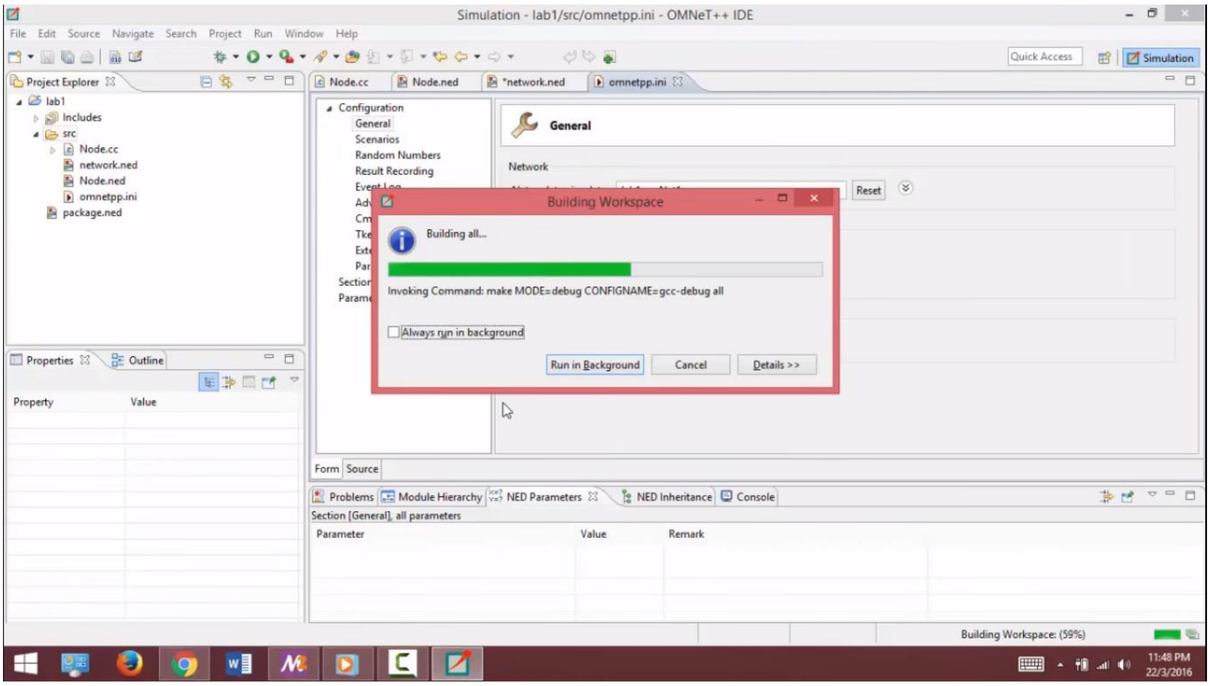


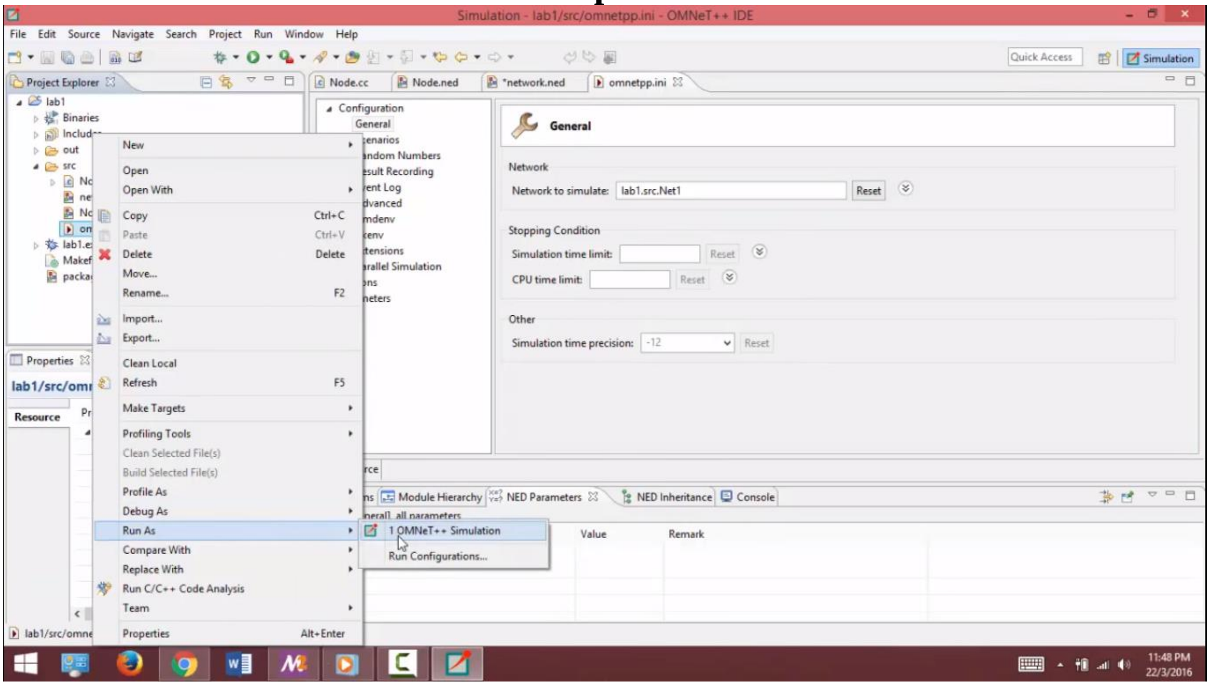

New Ini File
New Ini File
This wizard allows you to create a new OMNeT++ simulation configuration file
Enter or select the parent folder:
lab1/src
File name: omnetpp.ini
Advanced >>
< Back
Next >
Finish
Cancel

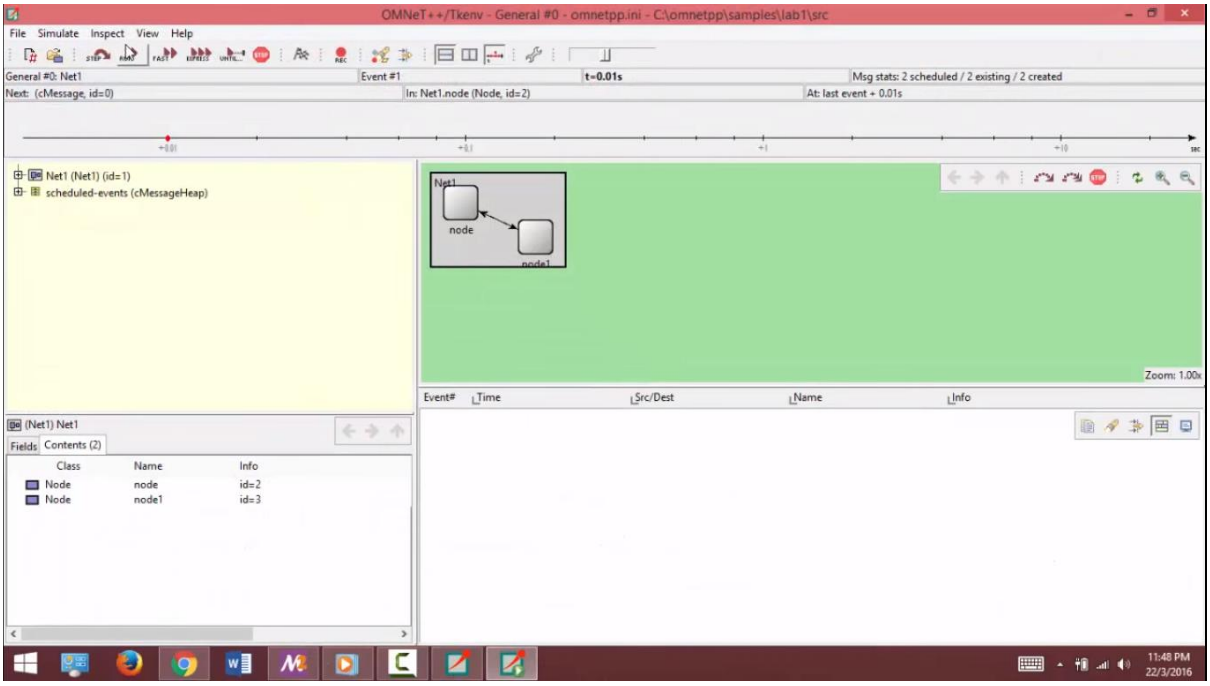

New Ini File
Ini File Options
Select options below
Select network:
Browse...
Preview
< Back
Next >
Finish
Cancel

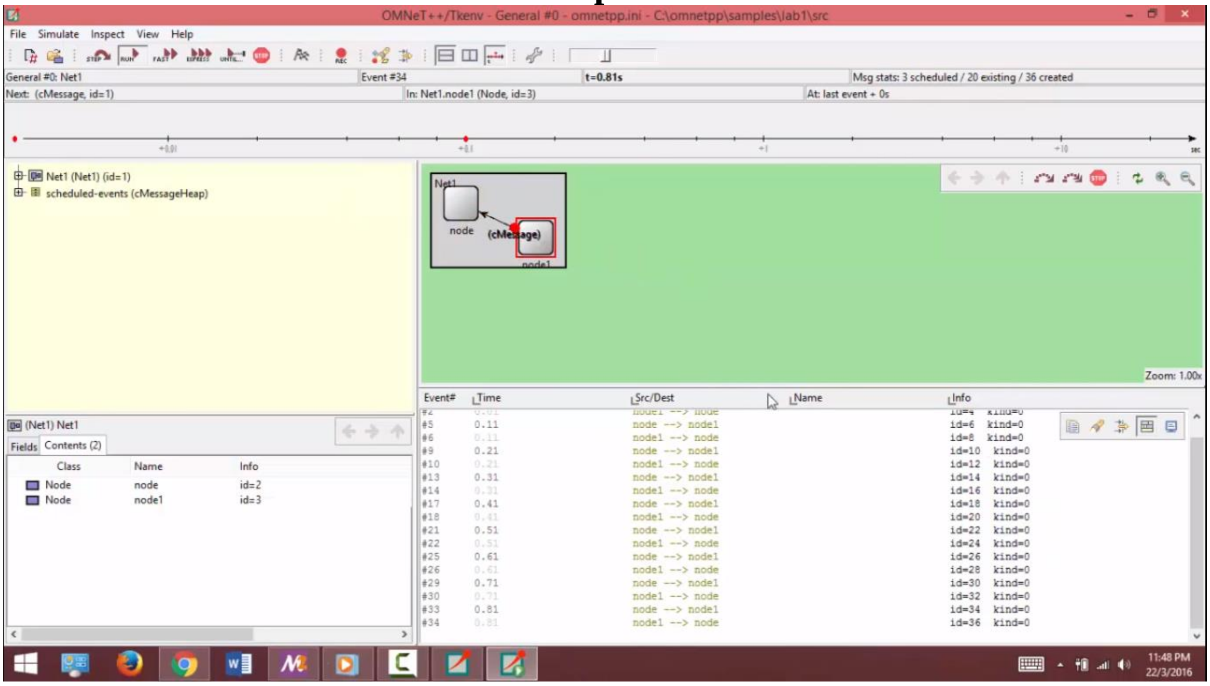


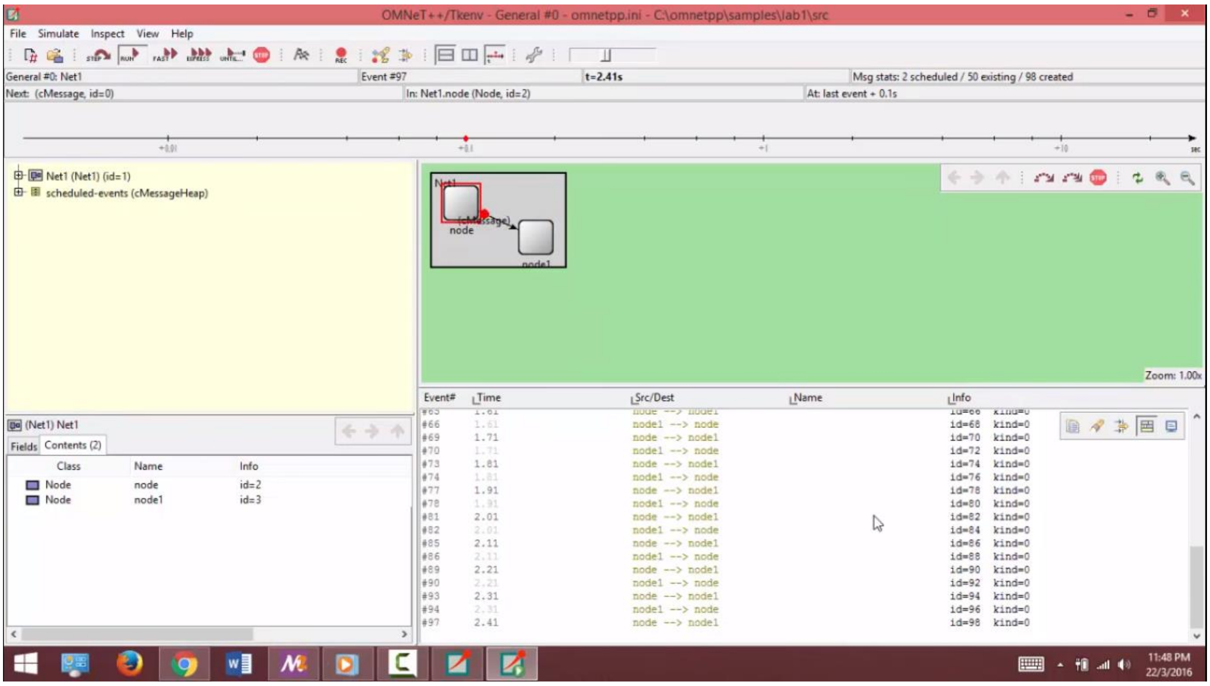
Now click ok, then Next and then Finish.




Simulation - lab1/src/omnetpp.ini - OMNeT++ IDE
File Edit Source Navigate Search Project Run Window Help
Project Explorer
Node.cc
Node.ned
*network.ned
omnetpp.ini
General
Network
Network to simulate: lab1.src.Net1
Stopping Condition
Simulation time limit:
CPU time limit:
Other
Simulation time precision:
New
Open
Open With
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename... F2
Import...
Export...
Clean Local
Refresh F5
Make Targets
Profiling Tools
Clean Selected File(s)
Build Selected File(s)
Profile As
Debug As
Run As
Compare With
Replace With
Run C/C++ Code Analysis
Team
Properties Alt+Enter
1 OMNeT++ Simulation
Run Configurations...
Module Hierarchy
NED Parameters
NED Inheritance
Console
Value
Remark


OMNeT++/Tkenv - General #0 - omnetpp.ini - C:\omnetpp\samples\lab1\src
File Simulate Inspect View Help
General #0: Net1
Event #1
t=0.01s
Msg stats: 2 scheduled / 2 existing / 2 created
Next: (cMessage, id=0)
In: Net1.node (Node, id=2)
At: last event + 0.01s
Net1 (Net1) (id=1)
scheduled-events (cMessageHeap)
node
node1
Zoom: 1.00x
Event#
Time
Src/Dest
Name
Info
(Net1) Net1
Fields
Contents (2)
Class
Name
Info
Node
node
id=2
Node
node1
id=3





You can see the message is being sent from one node to the other.

Understanding packet sniffers/protocol analyzers

(Using Wireshark) CLO2

Lab 11

Understanding HTTP working with Wireshark

In this lab, we will explore several aspects of the HTTP protocol: the basic GET/response interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded objects, and HTTP authentication and security. Before beginning these labs.[1]

1. The Basic HTTP GET/response interaction

Let us begin our exploration of HTTP by downloading a very simple HTML file - one that is very short and contains no embedded objects. Do the following:

1. Start up your web browser.
2. Start up the Wireshark packet sniffer (but don't yet begin packet capture). Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We are only interested in the HTTP protocol here, and do not want to see the clutter of all captured packets).
3. Wait a bit more than one minute (we will see why shortly), and then begin Wireshark packet capture.
4. Enter the following to your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
   Your browser should display the very simple, one-line HTML file.
5. Stop Wireshark packet capture.

Your Wireshark window should look similar to the window shown in Figure 1. If you are unable to run Wireshark on a live network connection, you can download a packet trace that was created when the steps above were followed.[2]

---

[1] References to figures and sections are for the 6th edition of our text, *Computer Networks, A Top-down Approach, 6th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2012.*

[2] Download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file http-ethereal-trace-1. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the http-ethereal-trace-1 trace file. The resulting display should look similar to Figure 1. (The Wireshark user interface displays just a bit differently on different operating systems, and in different versions of Wireshark).

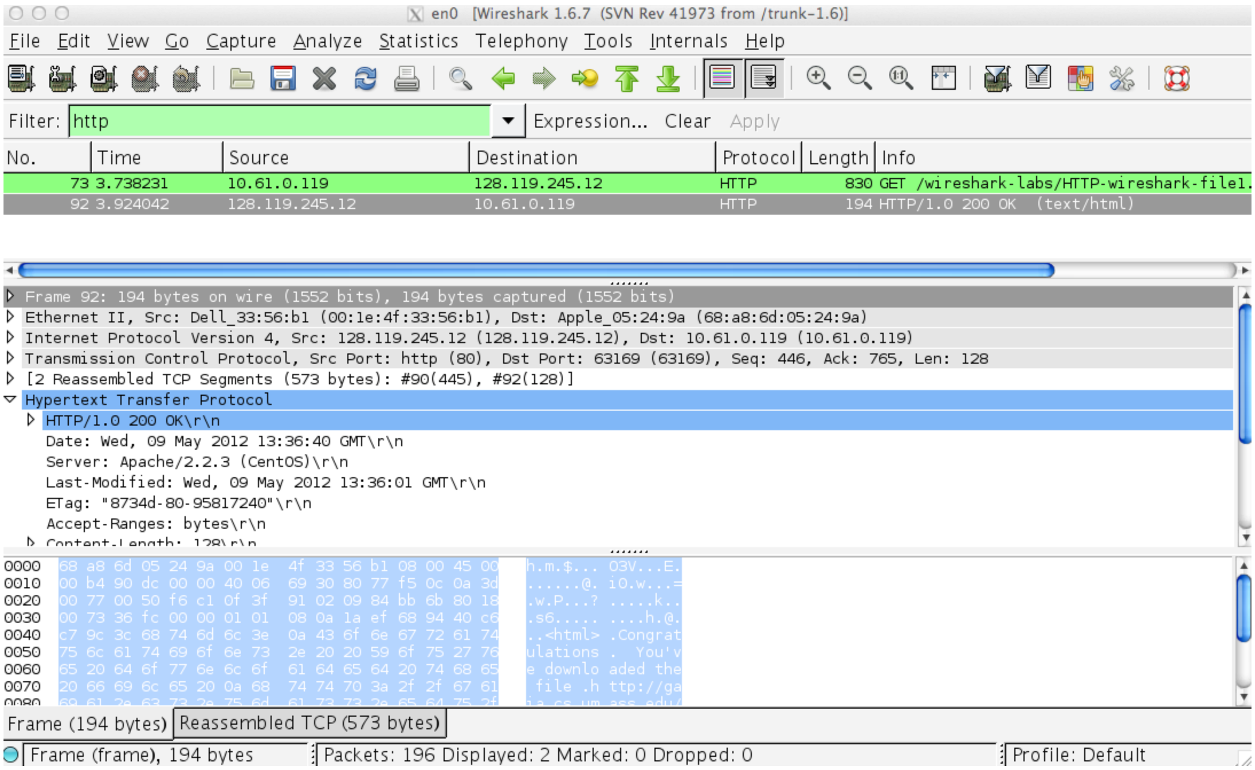


Figure 1: Wireshark Display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html has been retrieved by your browser

The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP OK message, which is highlighted in the packet-listing window). Since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well. We want to minimize the amount of non-HTTP data displayed (we're interested in HTTP here, and will be investigating these other protocols is later labs), so make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign or a right-pointing triangle (which means there is hidden, undisplayed information), and the HTTP line has a minus sign or a down-pointing triangle (which means that all information about the HTTP message is displayed).

(*Note:* You should ignore any HTTP GET and response for favicon.ico. If you see a reference to this file, it is your browser automatically asking the server if it (the server) has a small icon file that should be displayed next to the displayed URL in your browser. We will ignore references to this pesky file in this lab.).

By looking at the information in the HTTP GET and response messages, answer the following questions. When answering the following questions, you should print out the GET and response messages (see the online introductory Wireshark lab for an explanation of how to do this) and indicate where in the message you've found the information that answers the following questions.

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
2. What languages (if any) does your browser indicate that it can accept to the server?
3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?
4. What is the status code returned from the server to your browser?
5. When was the HTML file that you are retrieving last modified at the server?
6. How many bytes of content are being returned to your browser?
7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

In your answer to question 5 above, you might have been surprised to find that the document you just retrieved was last modified within a minute before you downloaded the document. That's because (for this particular file), the gaia.cs.umass.edu server is setting the file's last-modified time to be the current time and is doing so once per minute. Thus, if you wait a minute between accesses, the file will appear to have been recently modified, and hence your browser will download a "new" copy of the document.