CSE-331 Computer Networks
Lab No. 05

CLO3,4

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

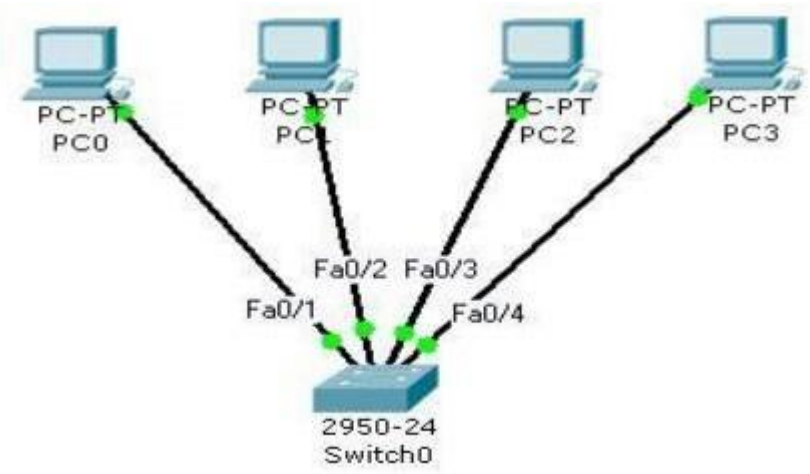
Aim:

Router Configuration using CLI

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.

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:

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

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 setting to full instead of Auto

Notice that the connection–immediately goes down. T bring it back up again, change the bandwidth and the duplex settings of the switch's Ethernet port to match current settings of PC0

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:

CSE-331 Computer Networks

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:

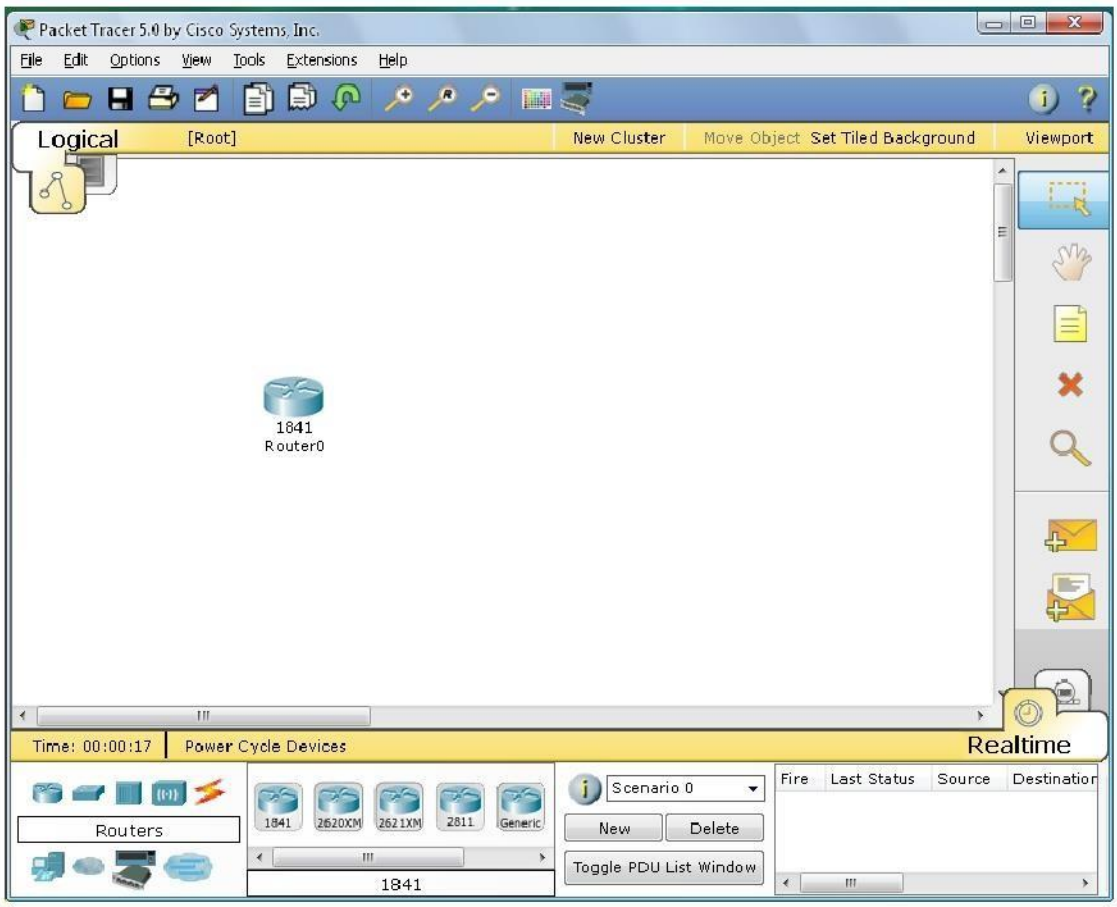
In any command mode, you can get a list of available commands by entering a question mark (?).

Router>?

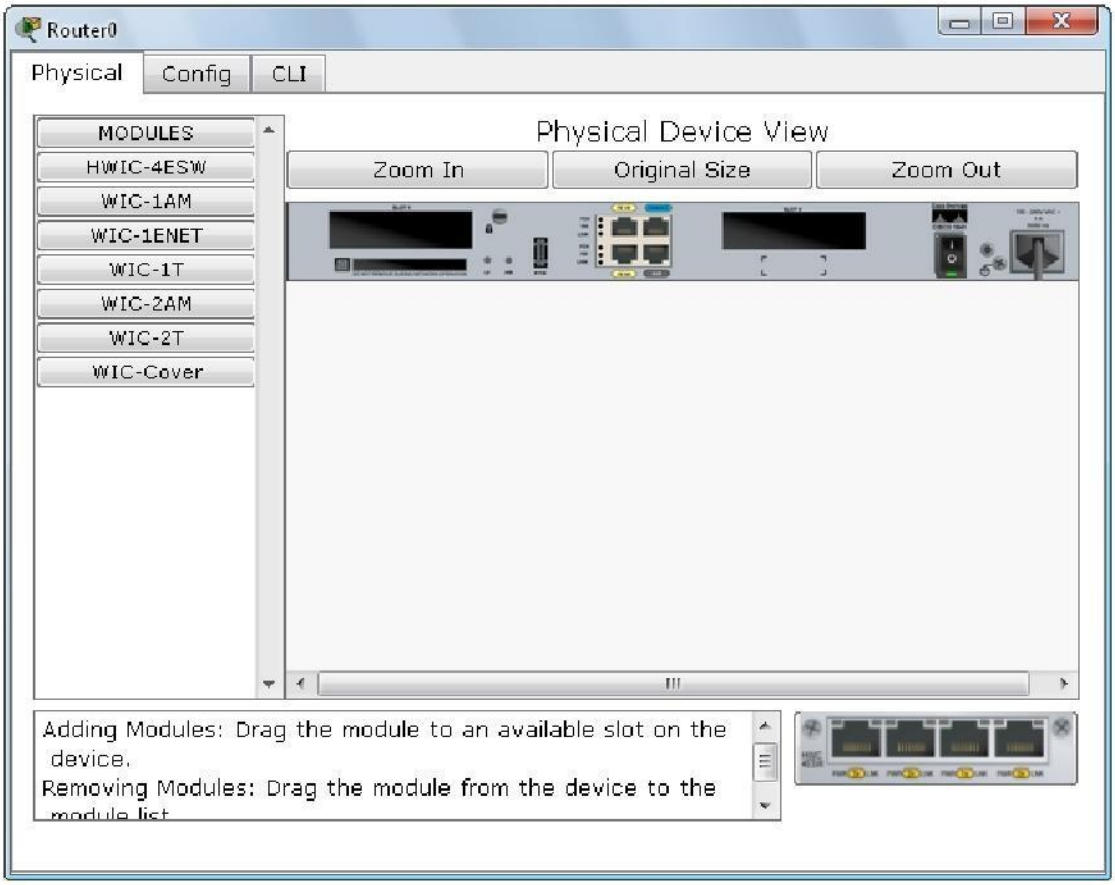
To obtain a list of command that begin with a particular character sequence, type in those characters followed immediately by the question mark (?).

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

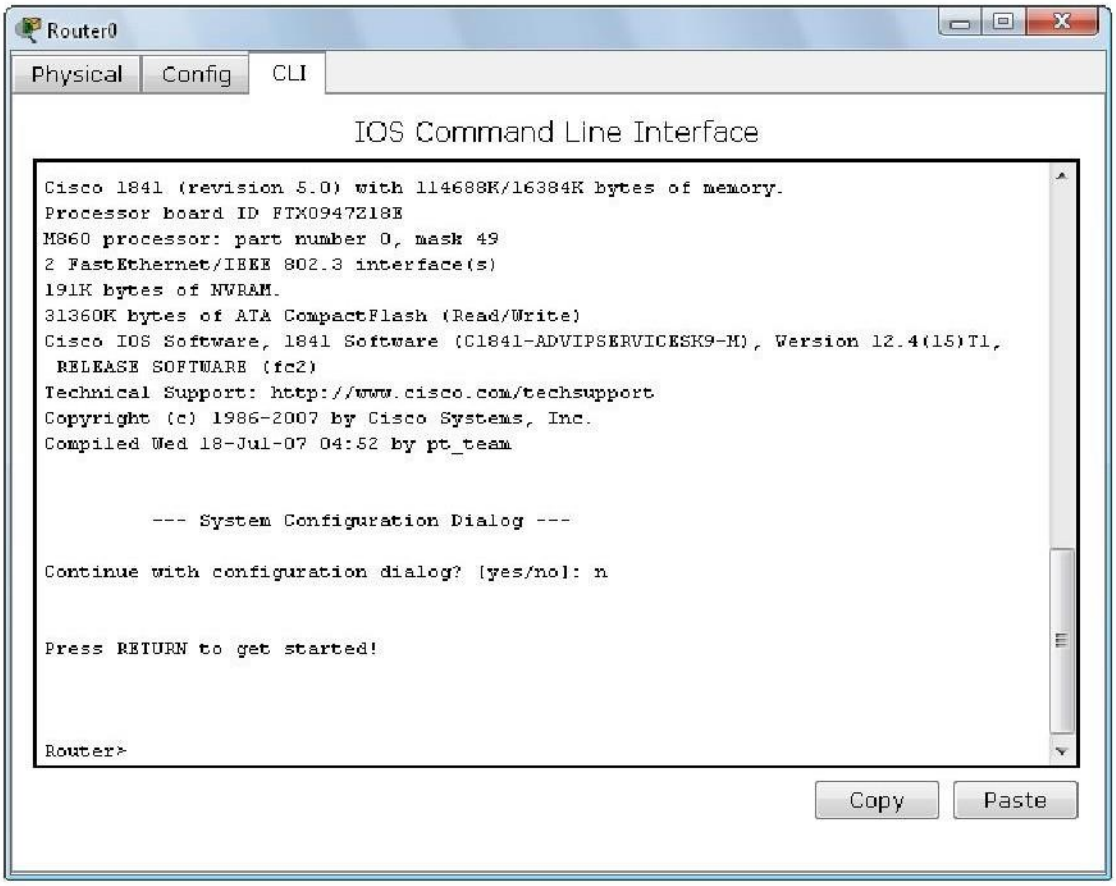
We choose a router:



Double click on the router chosen:

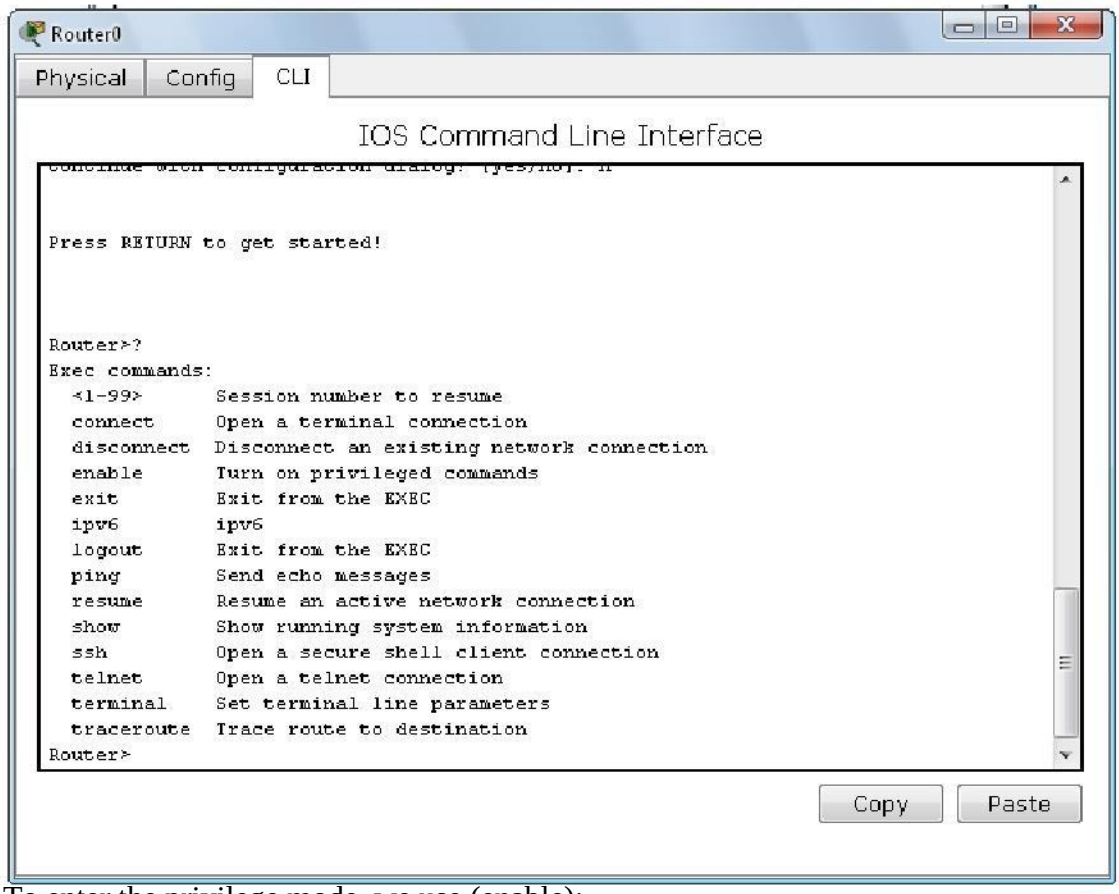


We enter CLI for router configuration:

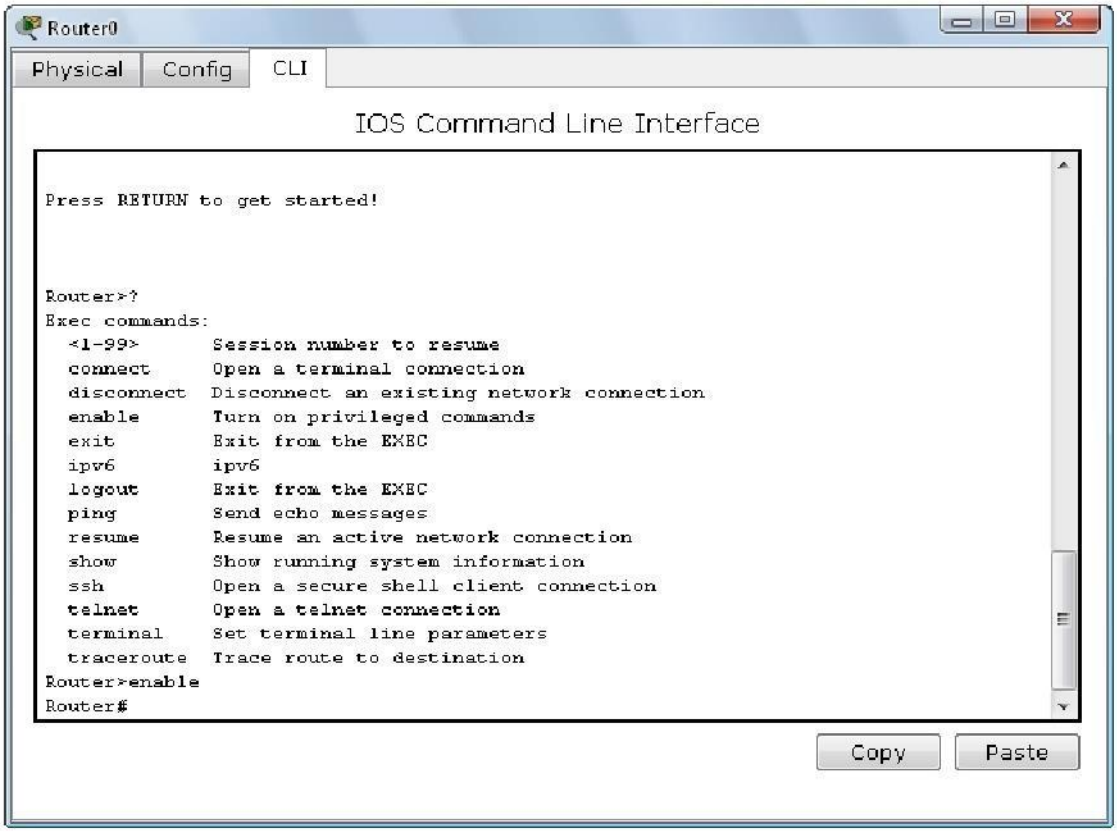


Now we are in the user mode:

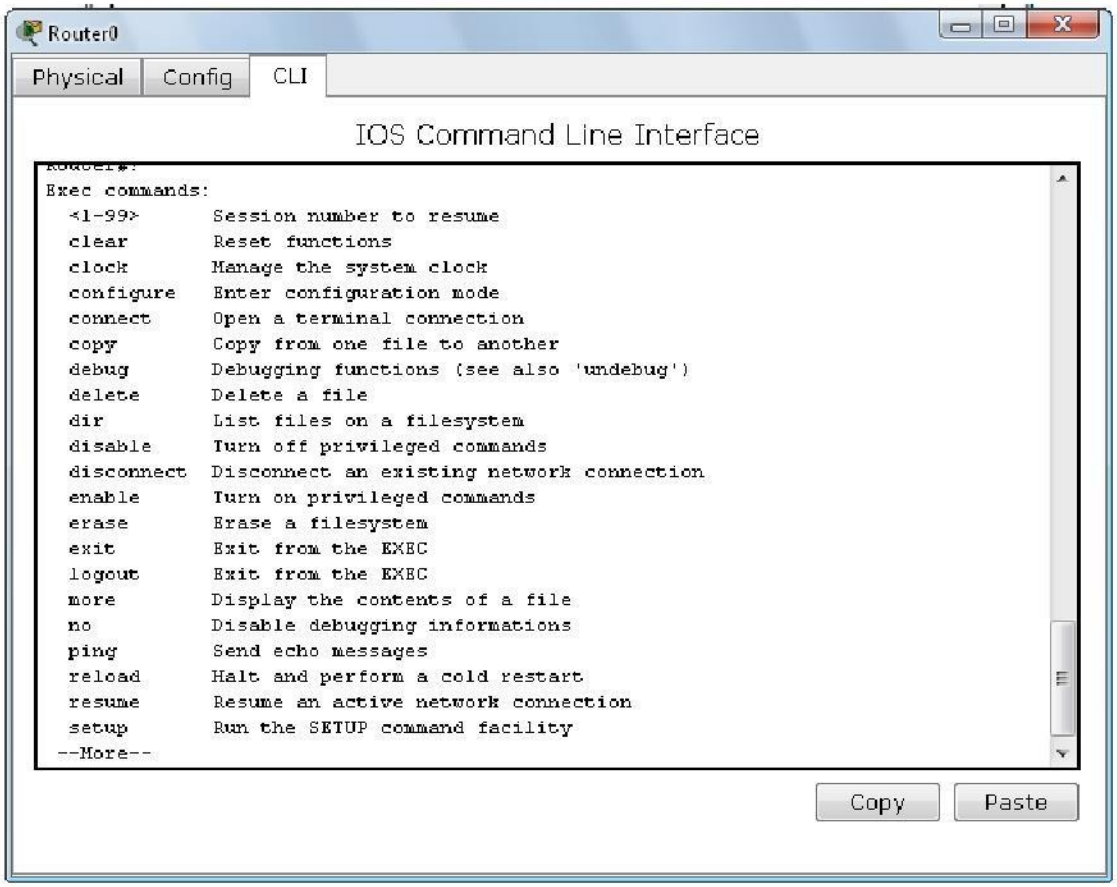
To know the commands in user mode we use (?) :



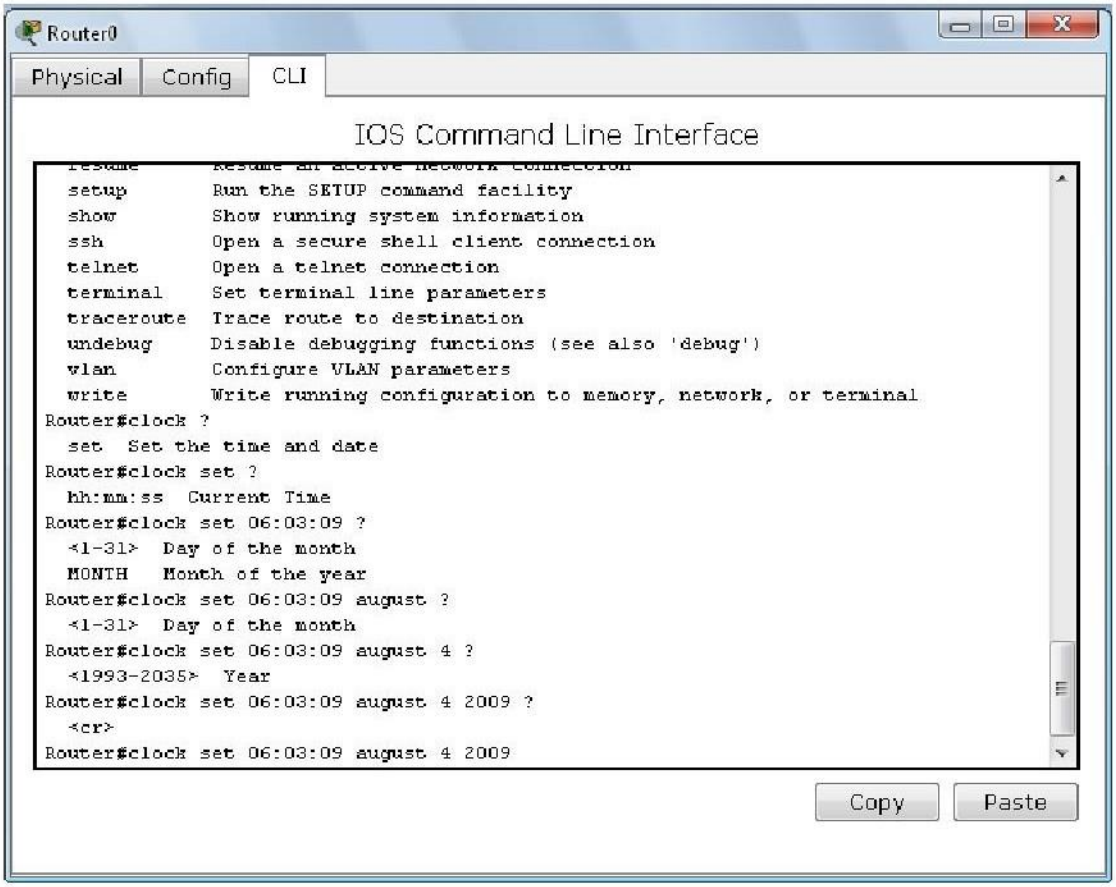
To enter the privilege mode, we use (enable):



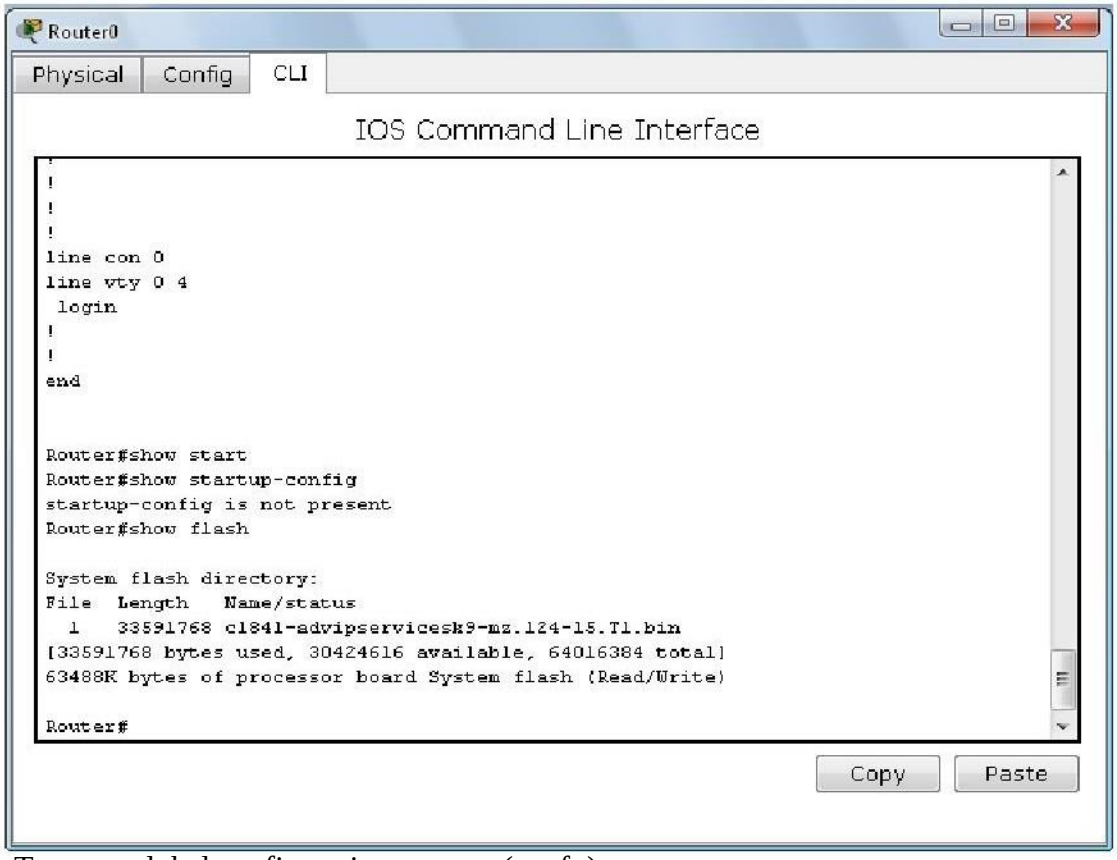
To see the commands in privileged mode we use (?):



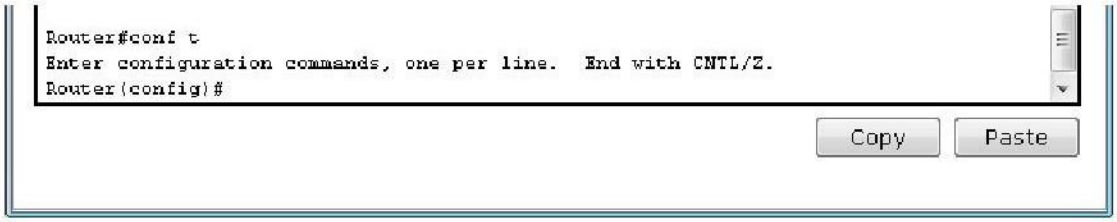
To manage the system clock, we use (clock):



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

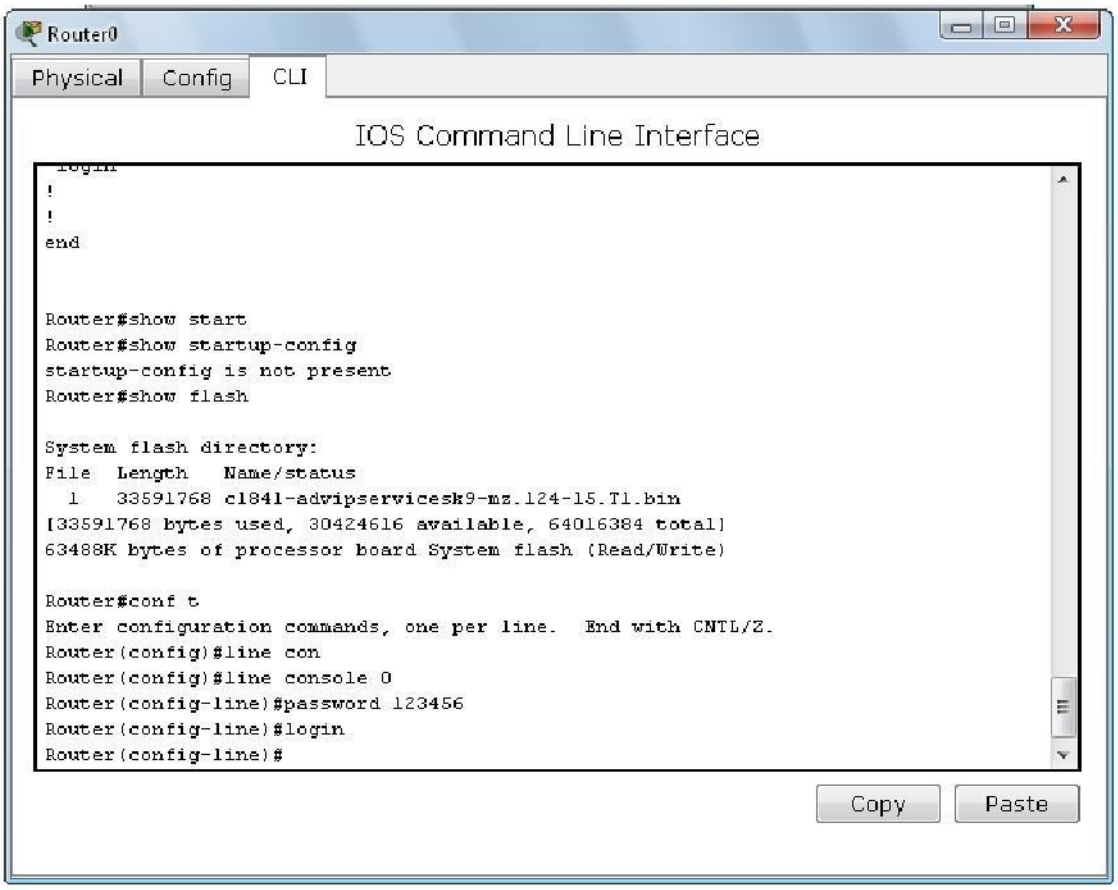


To go to global configuration, we use (conf t):

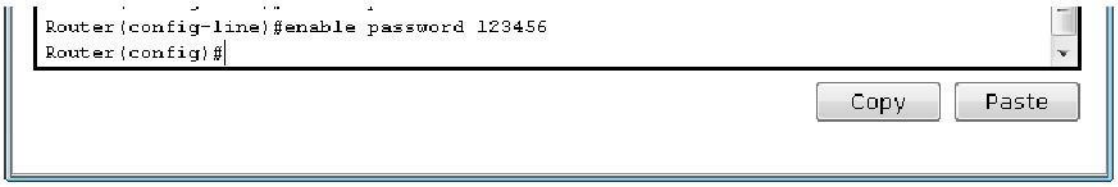


Passwords:

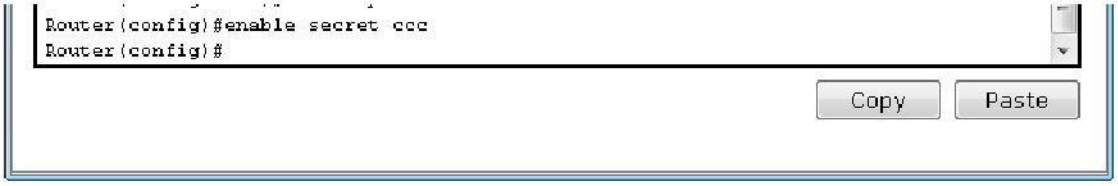
1- Line console password to protect the user mode:



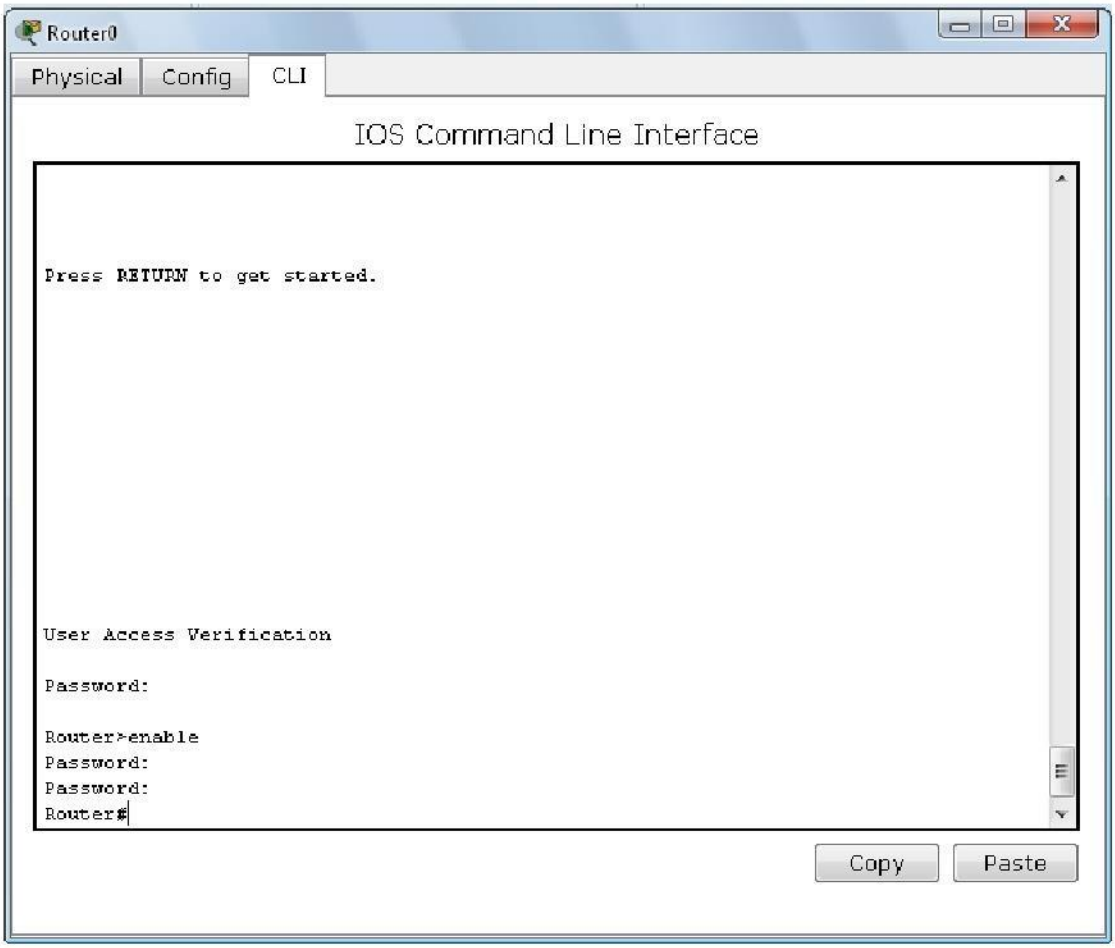
2- Enable password to protect the privilege mode:



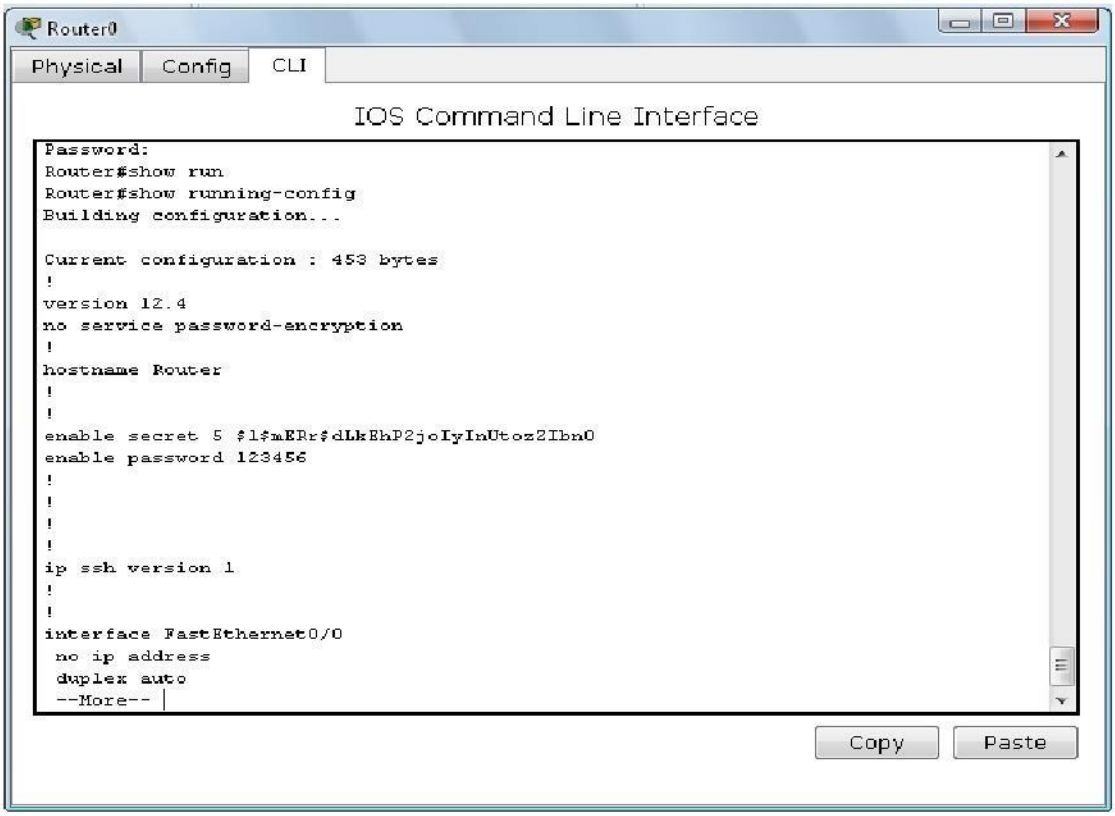
3- Secret password for more protection of privilege mode (more priority than enable pass and its encrypted pass.



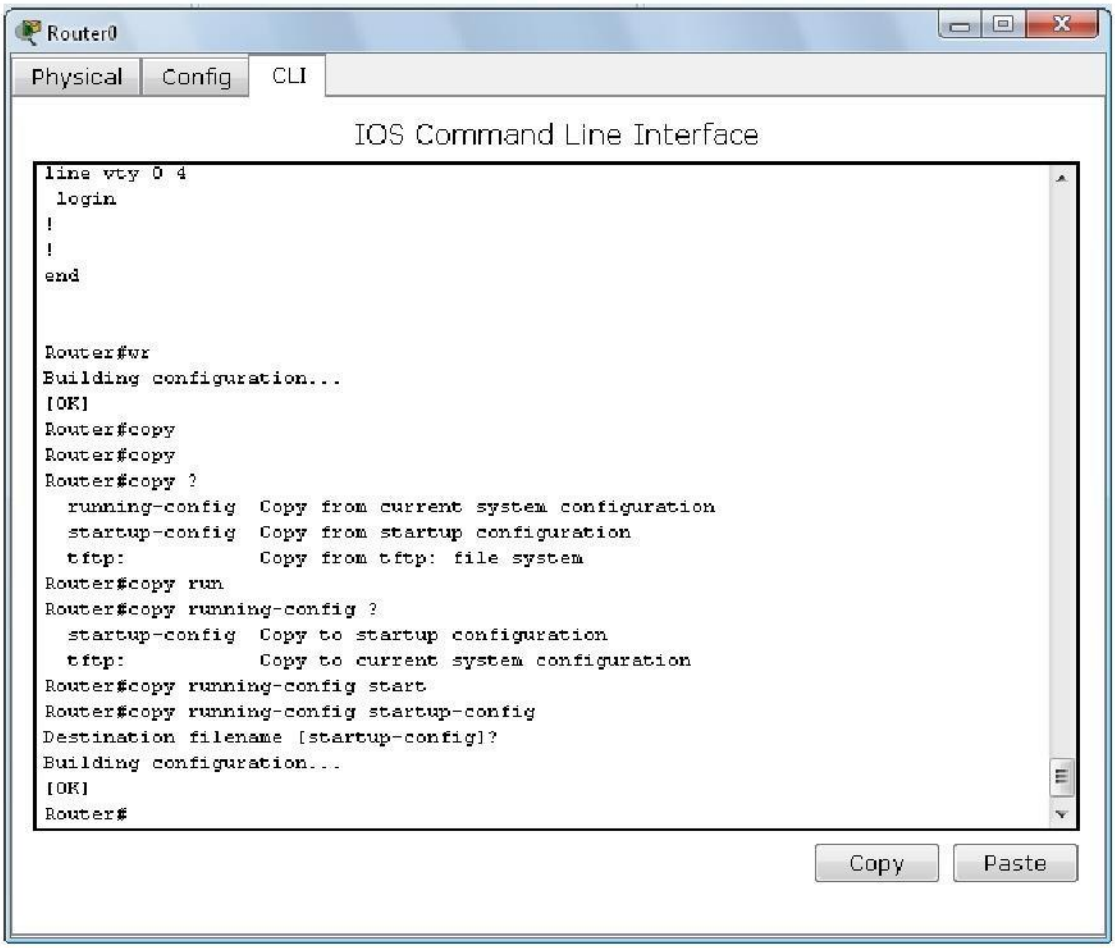
Now we try the passwords set:



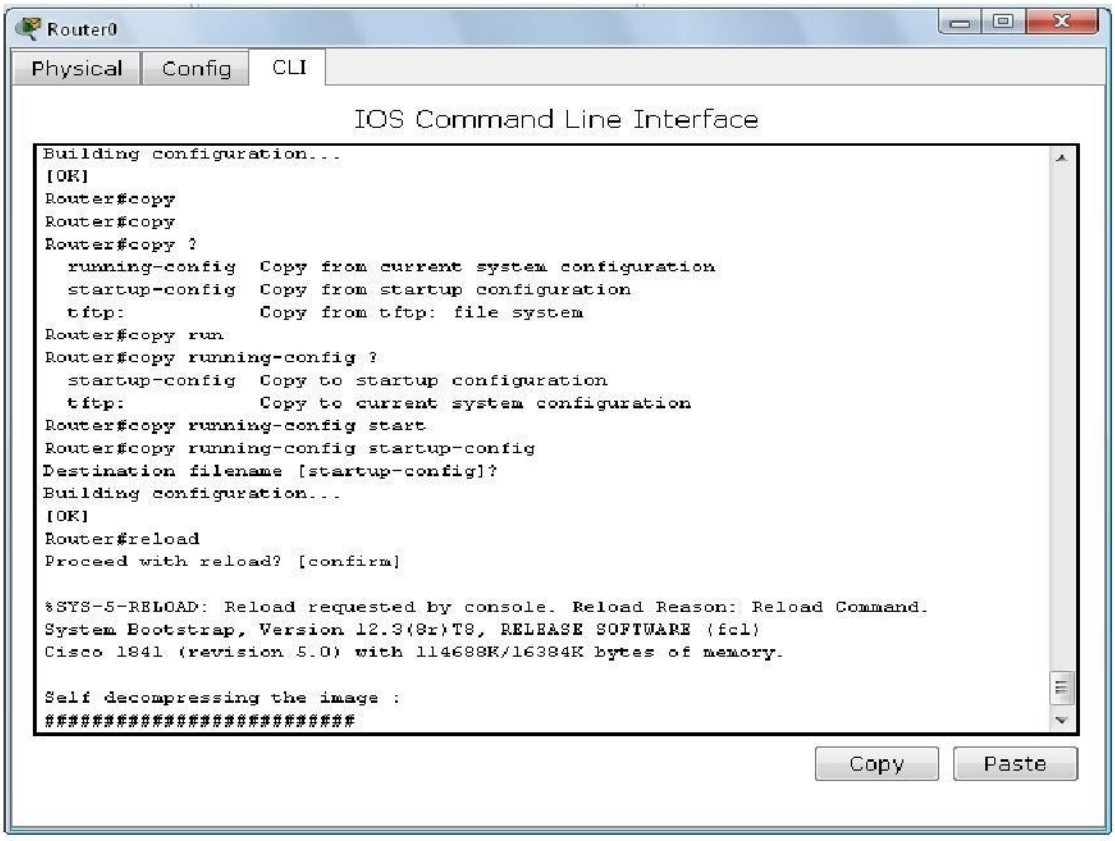
We notice that only secret pass is encrypted:

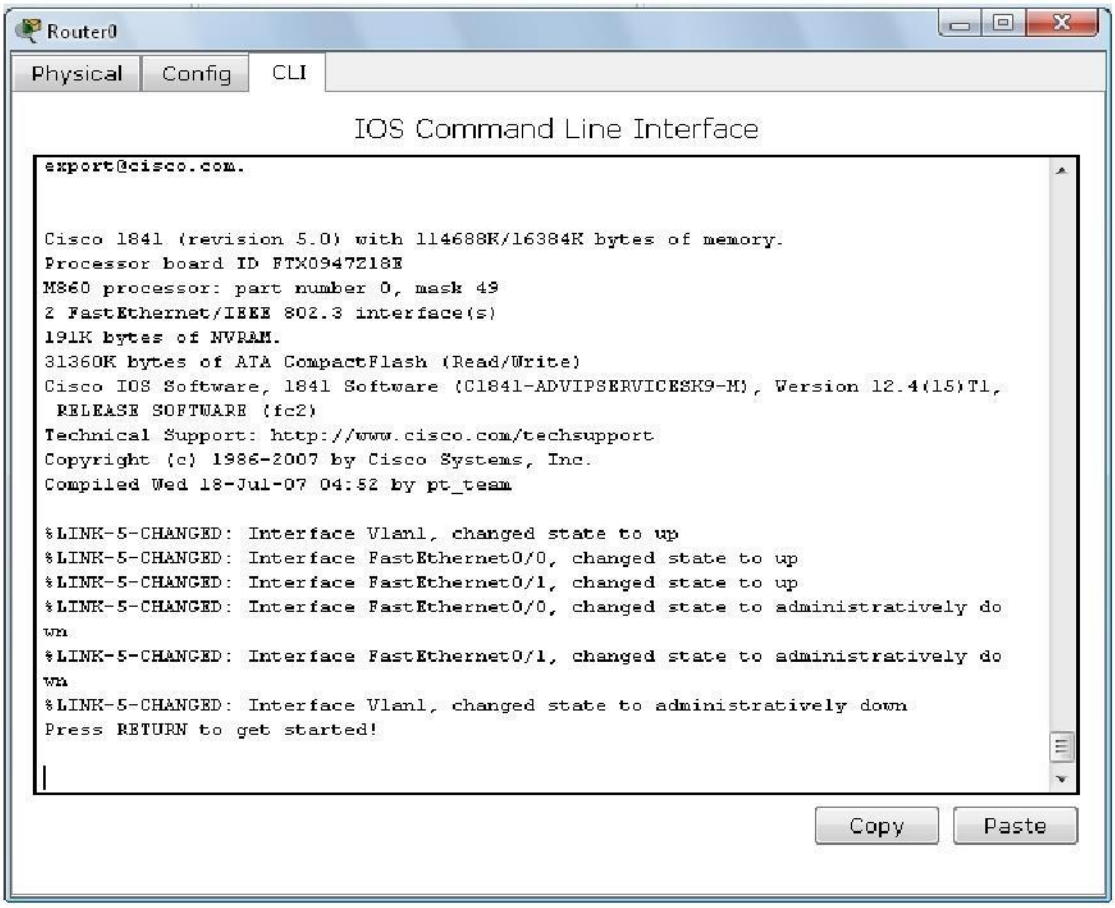


To save the configuration we did in the NVRAM we use (wr or copy commands).

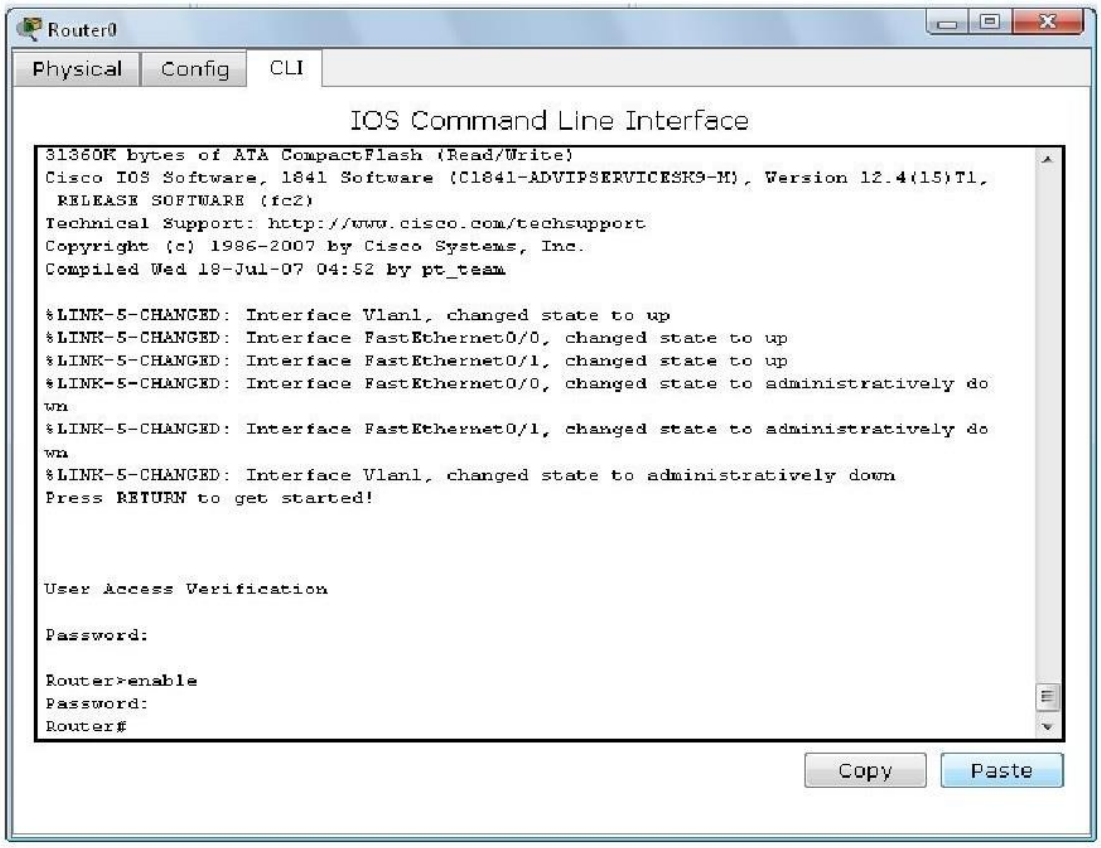


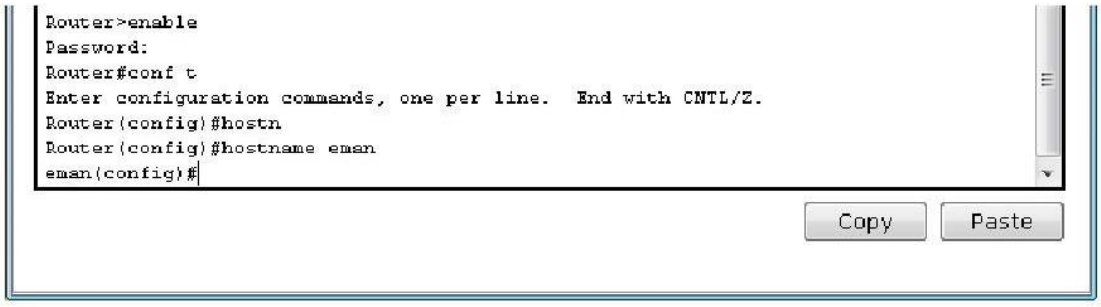
To restart we use the command (reload):


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

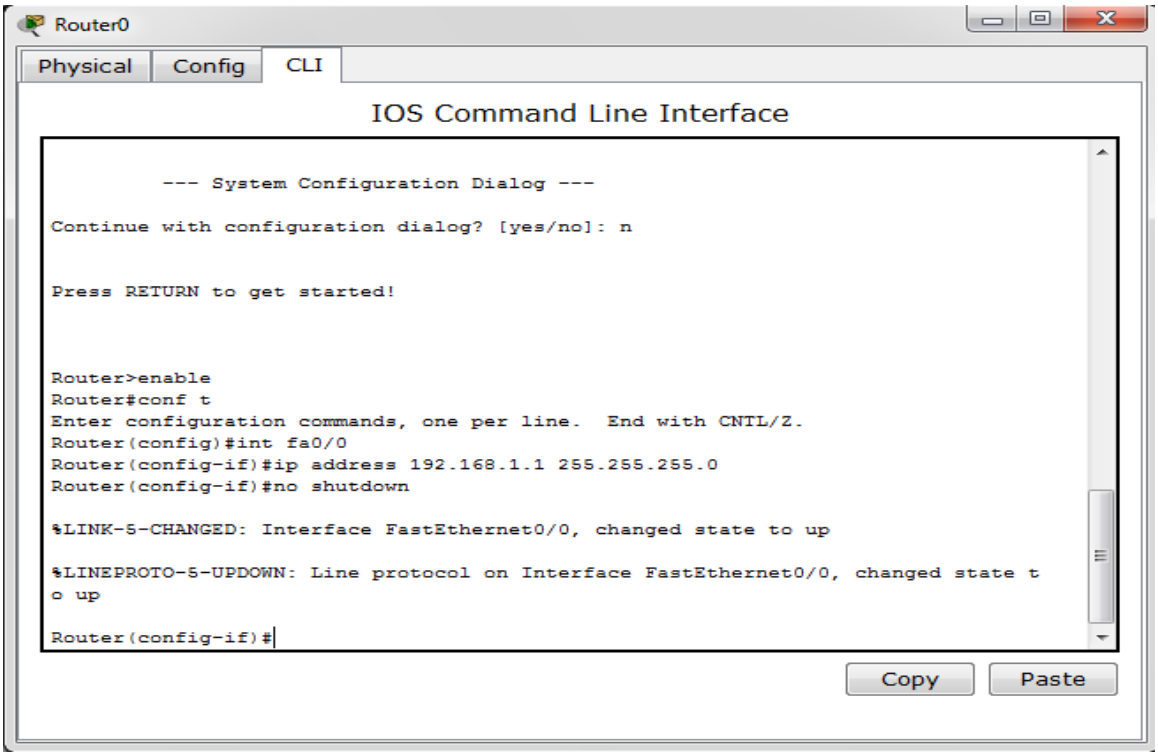

Router0
Physical
Config
CLI
IOS Command Line Interface
export@cisco.com.
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Processor board ID FTX0947Z18E
M860 processor: part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
191K bytes of NVRAM.
31360K bytes of ATA CompactFlash (Read/Write)
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 do
wn
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively do
wn
%LINK-5-CHANGED: Interface Vlan1, changed state to administratively down
Press RETURN to get started!
Copy
Paste

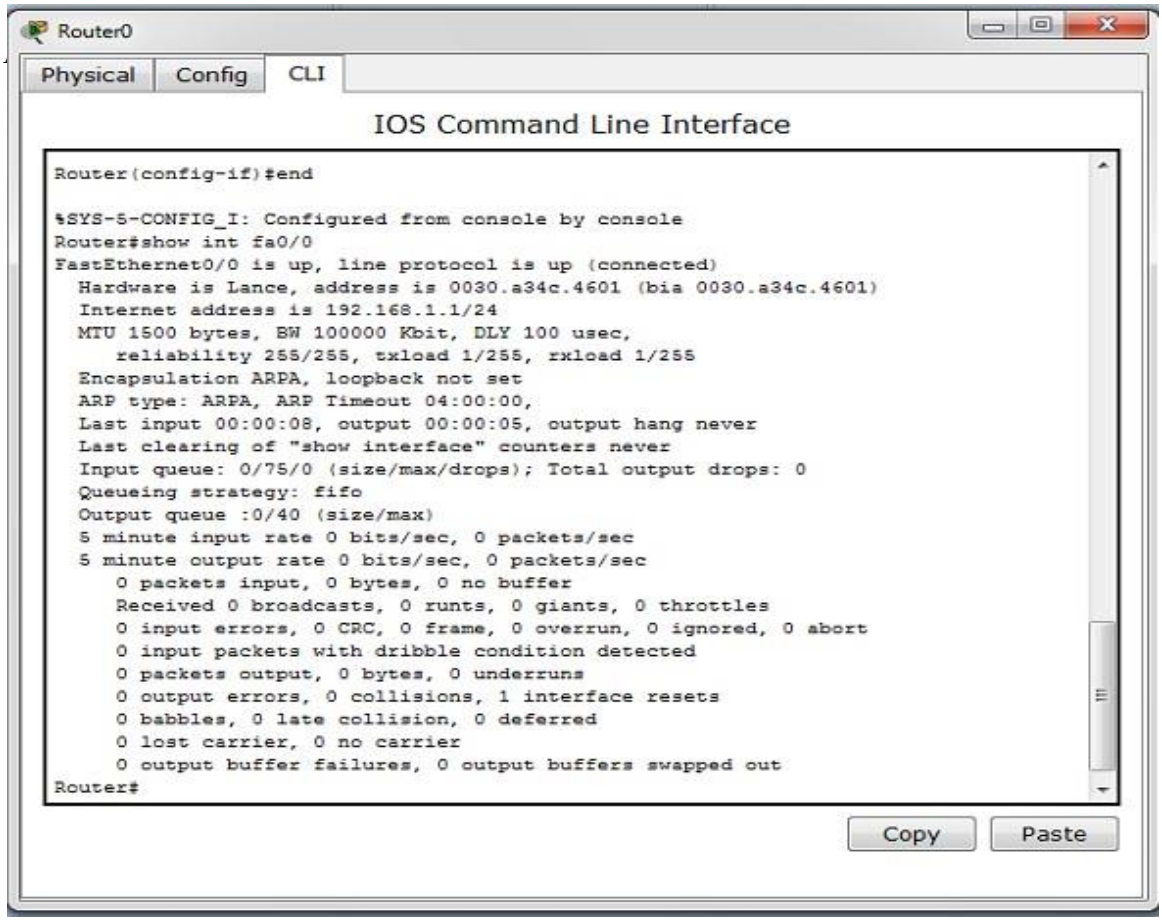
To change the host name of router:





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





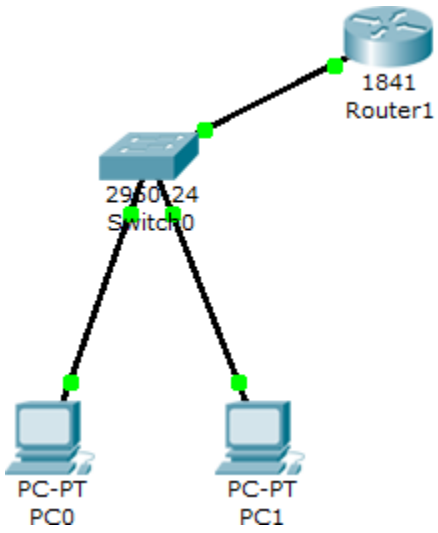
CSE-331 Computer Networks

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 No. 06

CLO3,2,4

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

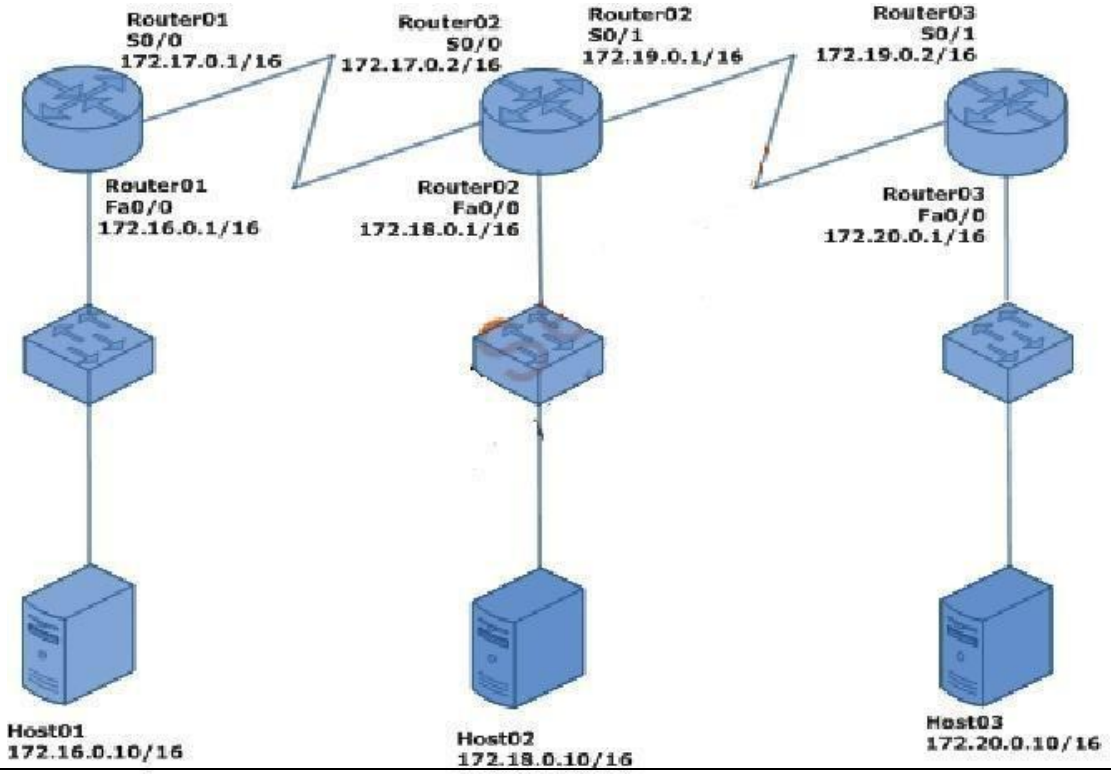
IP naming using host table.

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. 07

CLO2,3,4

Aim:

IP naming-II:

A router can learn about remote networks in one of two ways:

- **Manually:** Remote networks are manually entered into the route table using static routes.
- **Dynamically:** Remote routes are automatically learned using a dynamic routing protocol.

Figure 2-1 provides a sample scenario of static routing. Figure 2-2 provides a sample scenario of dynamic routing using EIGRP.

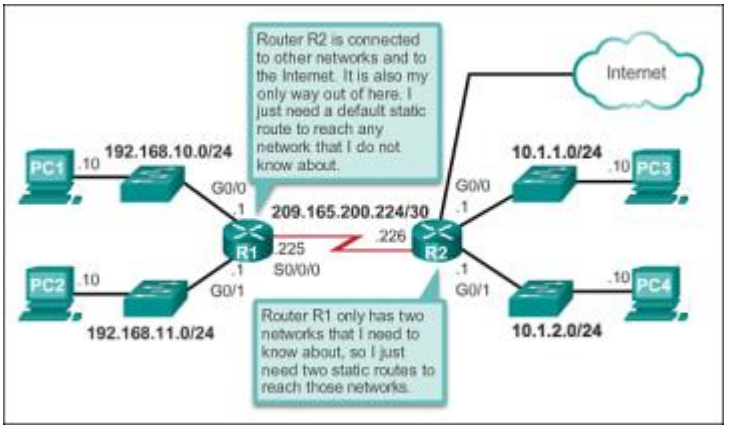


Figure 2-1 Static and Default Route Scenario

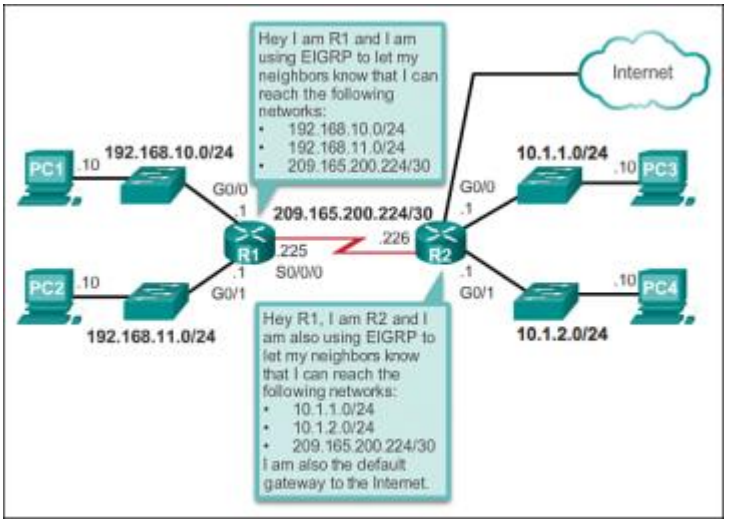


Figure 2-2 Dynamic Routing Scenario

A network administrator can manually configure a static route to reach a specific network. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured any time the network topology changes. A static route does not change until the administrator manually reconfigures it.

Why Use Static Routing?

Static routing provides some advantages over dynamic routing, including:

- Static routes are not advertised over the network, resulting in better security.
- Static routes use less bandwidth than dynamic routing protocols, as routers do not exchange routes.
- No CPU cycles are used to calculate and communicate routes.
- The path a static route uses to send data is known.

Static routing has the following disadvantages:

- Initial configuration and maintenance is time-consuming.
- Configuration can be error-prone, especially in large networks.
- Administrator intervention is required to maintain changing route information.
- Does not scale well with growing networks; maintenance becomes cumbersome.
- Requires complete knowledge of the whole network for proper implementation.

In Table 2-1, dynamic and static routing features are compared. Notice that the advantages of one method are the disadvantages of the other.

Table 2-1 Dynamic Routing Versus Static Routing

	Dynamic Routing	**Static Routing**
Configuration Complexity	Generally independent of the network size	Increases with the network size
Topology Changes	Automatically adapts to topology changes	Administrator intervention required
Scaling	Suitable for simple and complex topologies	Suitable for simple topologies
Security	Less secure	More secure
Resource Usage	Uses CPU, memory, link bandwidth	No extra resources needed
Predictability	Route depends on the current topology	Route to destination is always the same

Lab No. 09

CLO2,3

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)

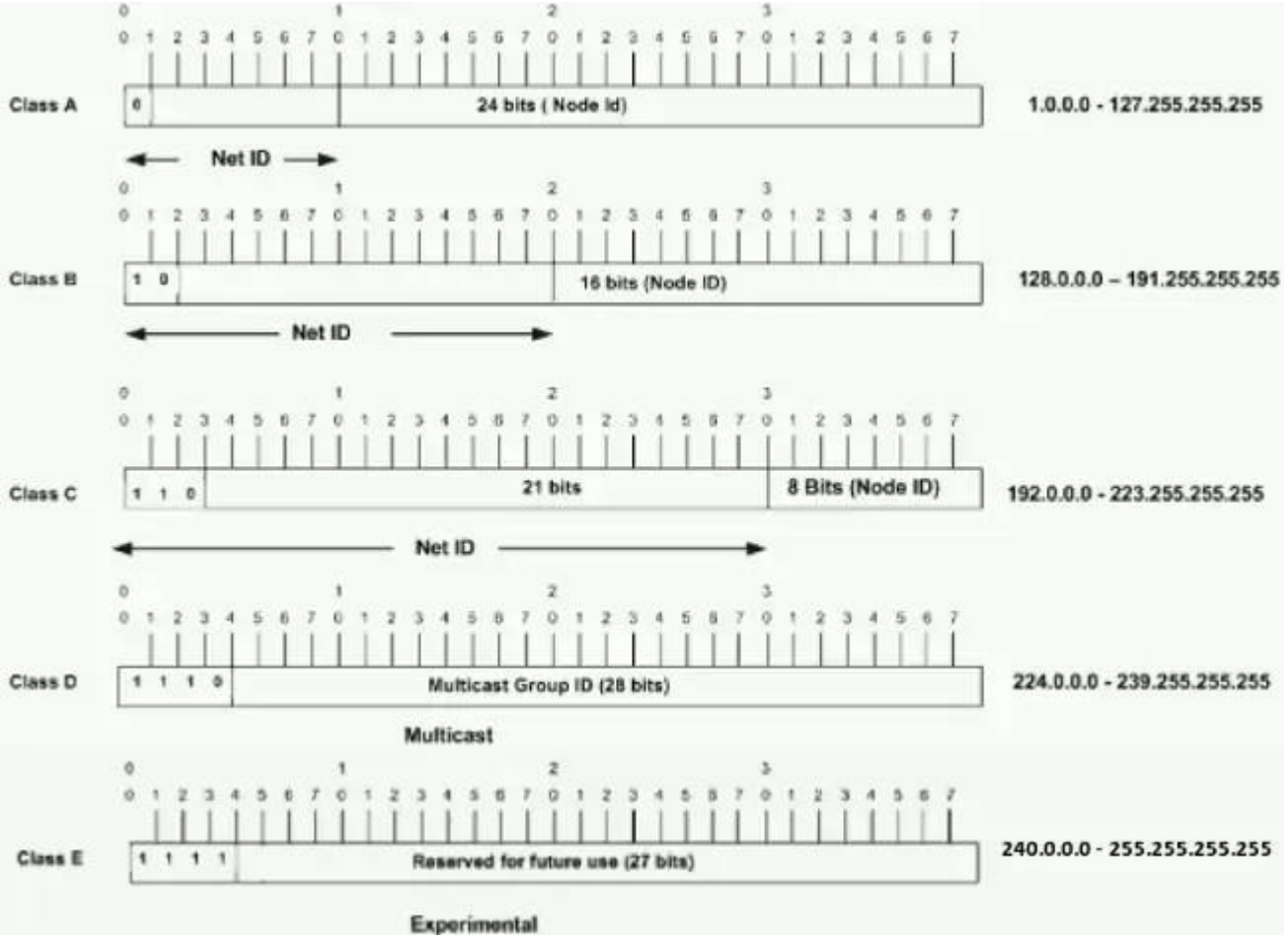
And this sample shows an IP address represented in both binary and decimal.

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 No. 10

CLO1,3,4

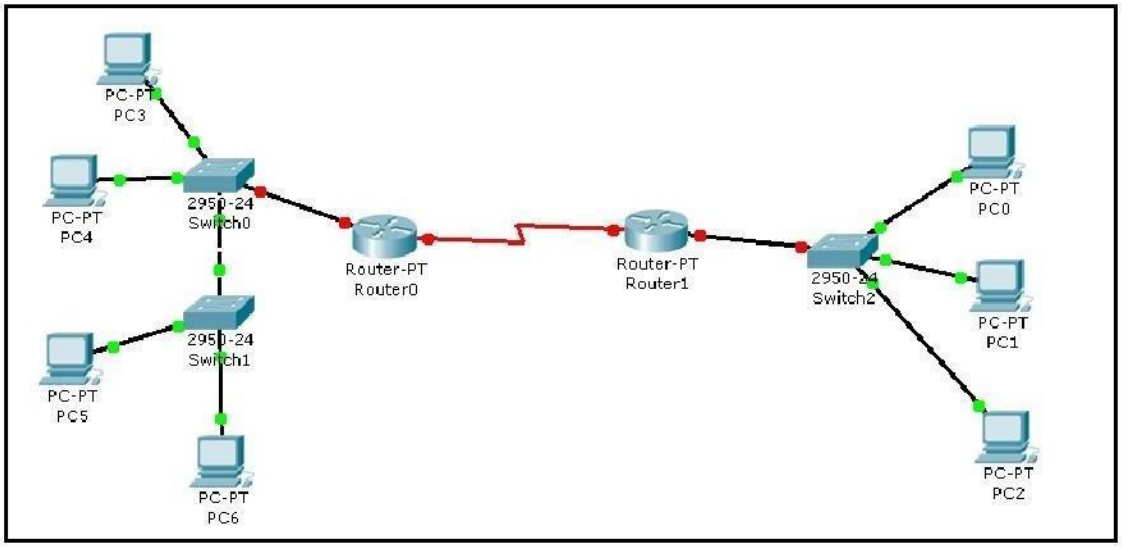
Dynamic Routing (RIP)

Using packet tracer in addressing :

1- We connect the figure .

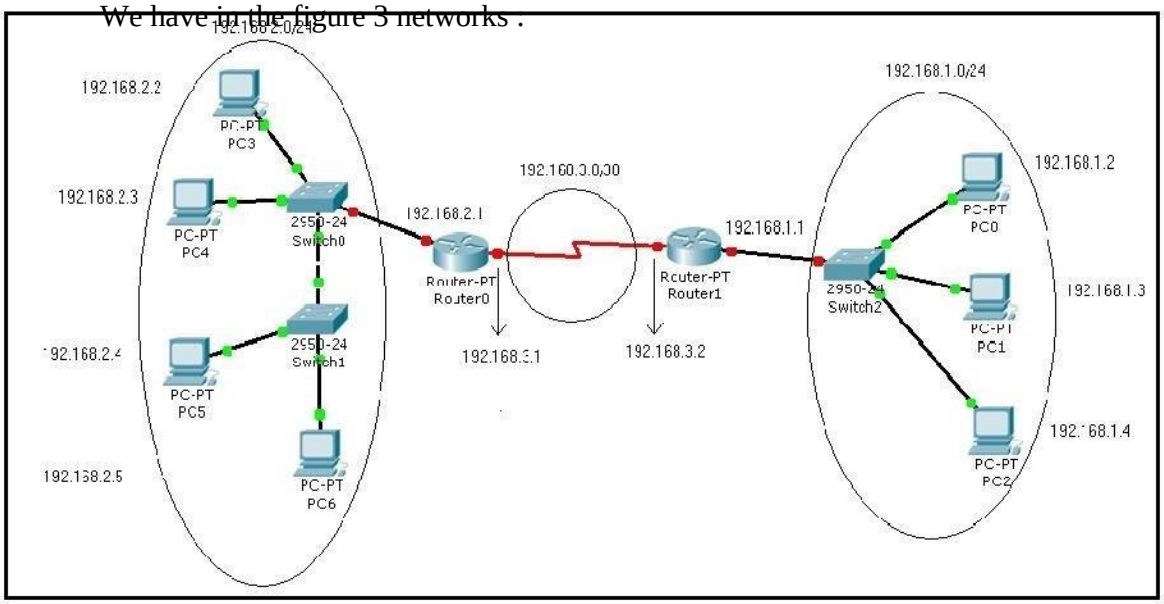
The cables used are :

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

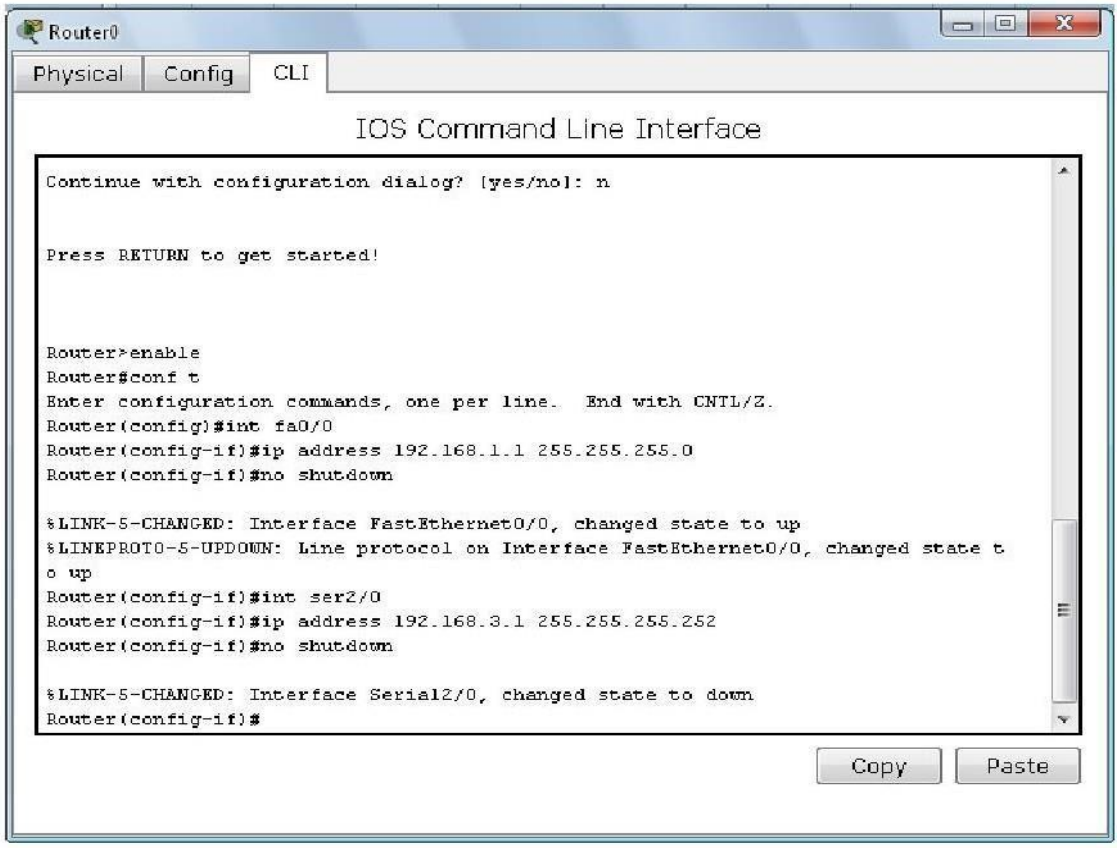


2- We put the IP address for each device as the following :

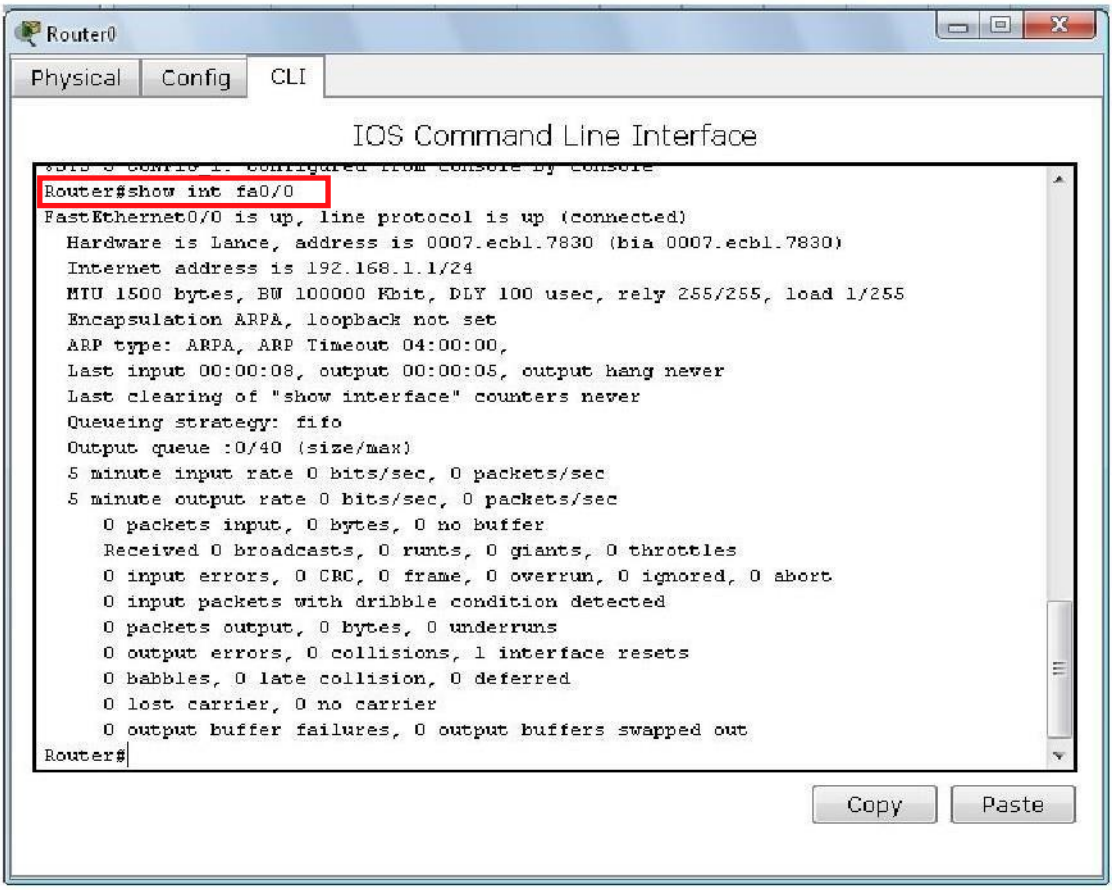
We have in the figure 3 networks .

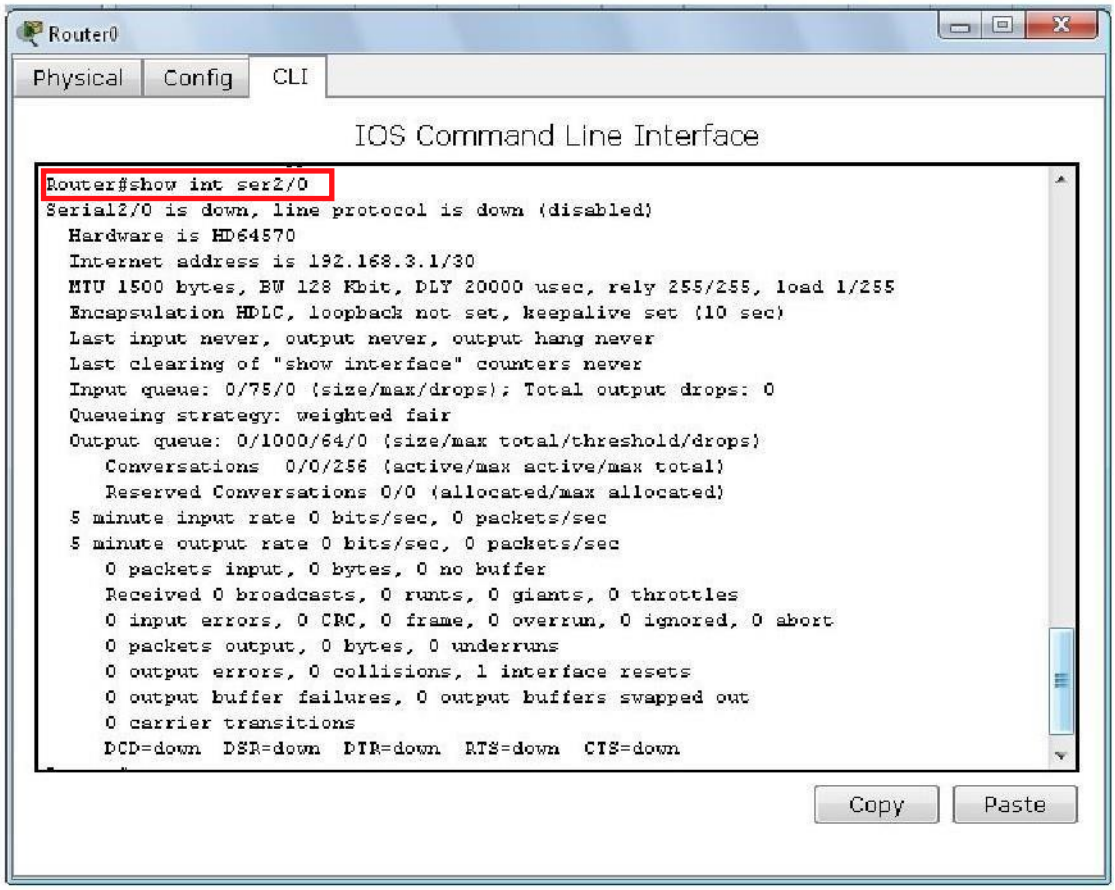


To disrtribute the Ip's on the router interfaces we do the following :

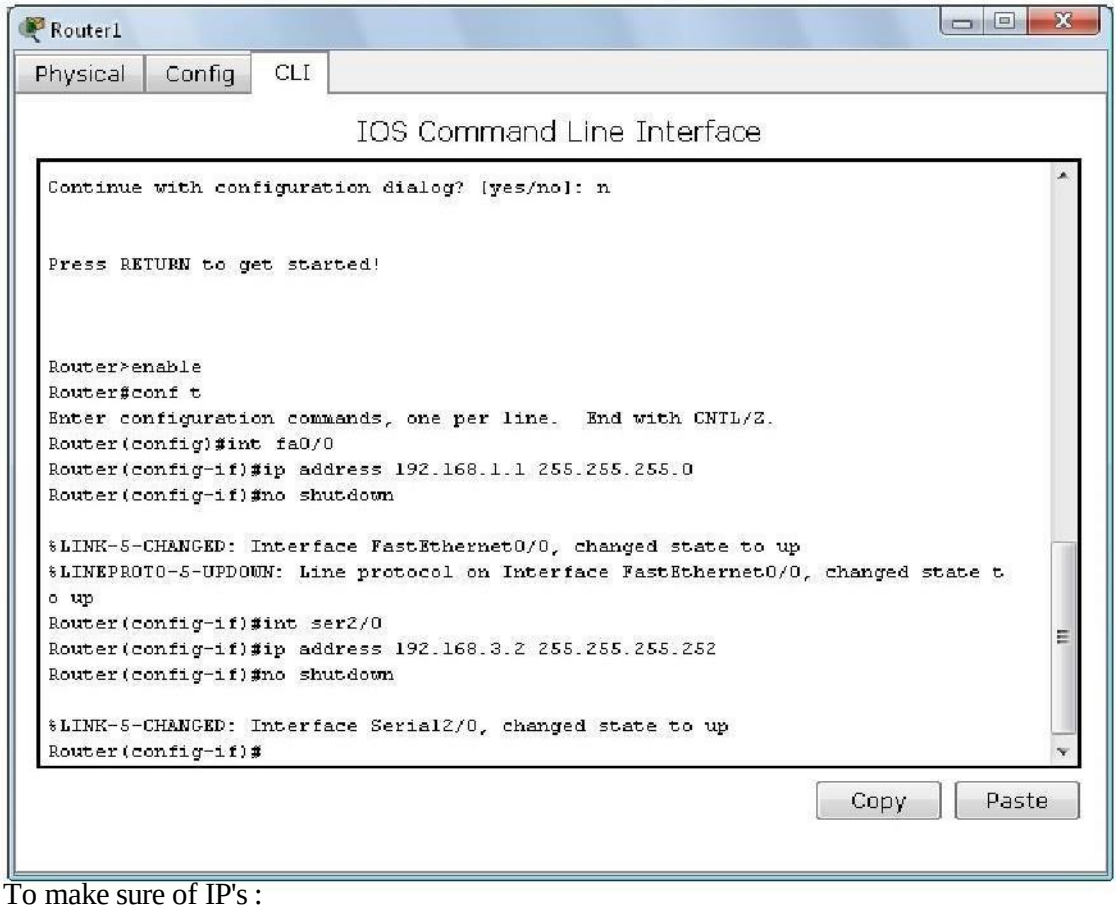


To make sure of the IP :

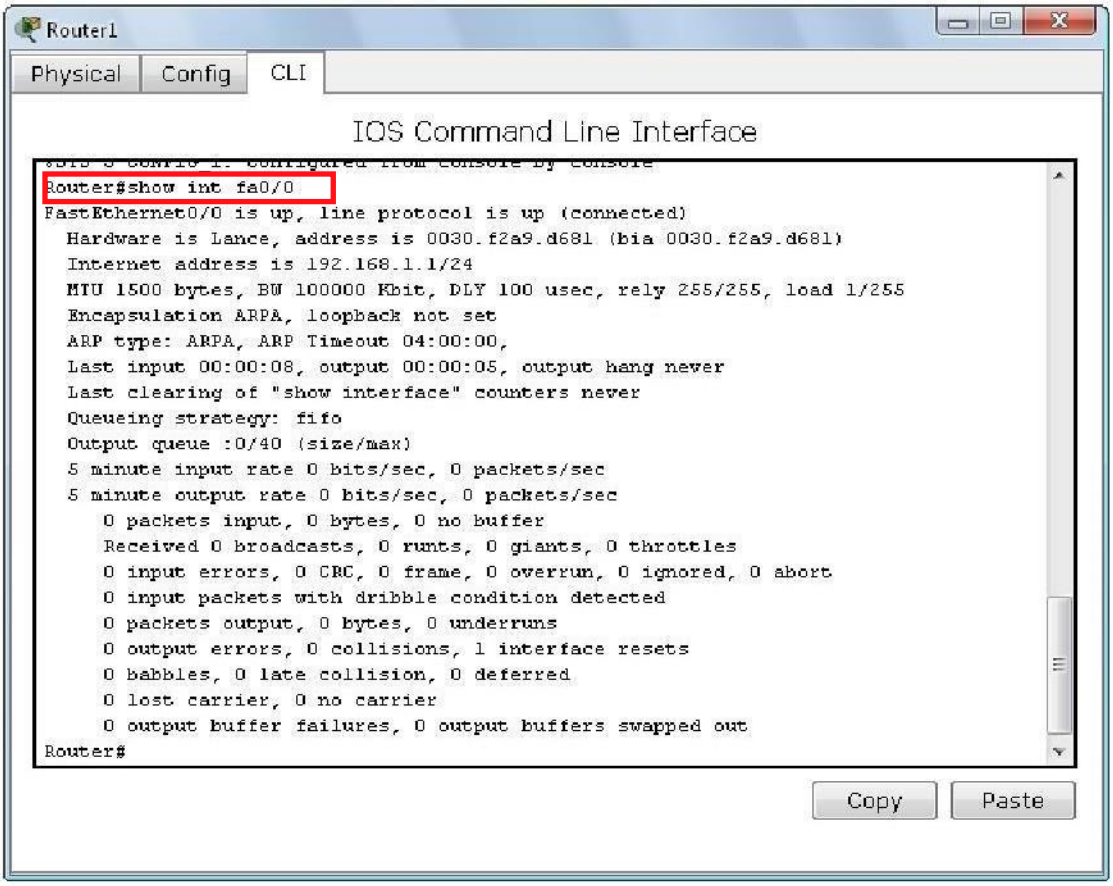

Router0
Physical
Config
CLI
IOS Command Line Interface
Router$show int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0007.ecb1.7830 (bia 0007.ecb1.7830)
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
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

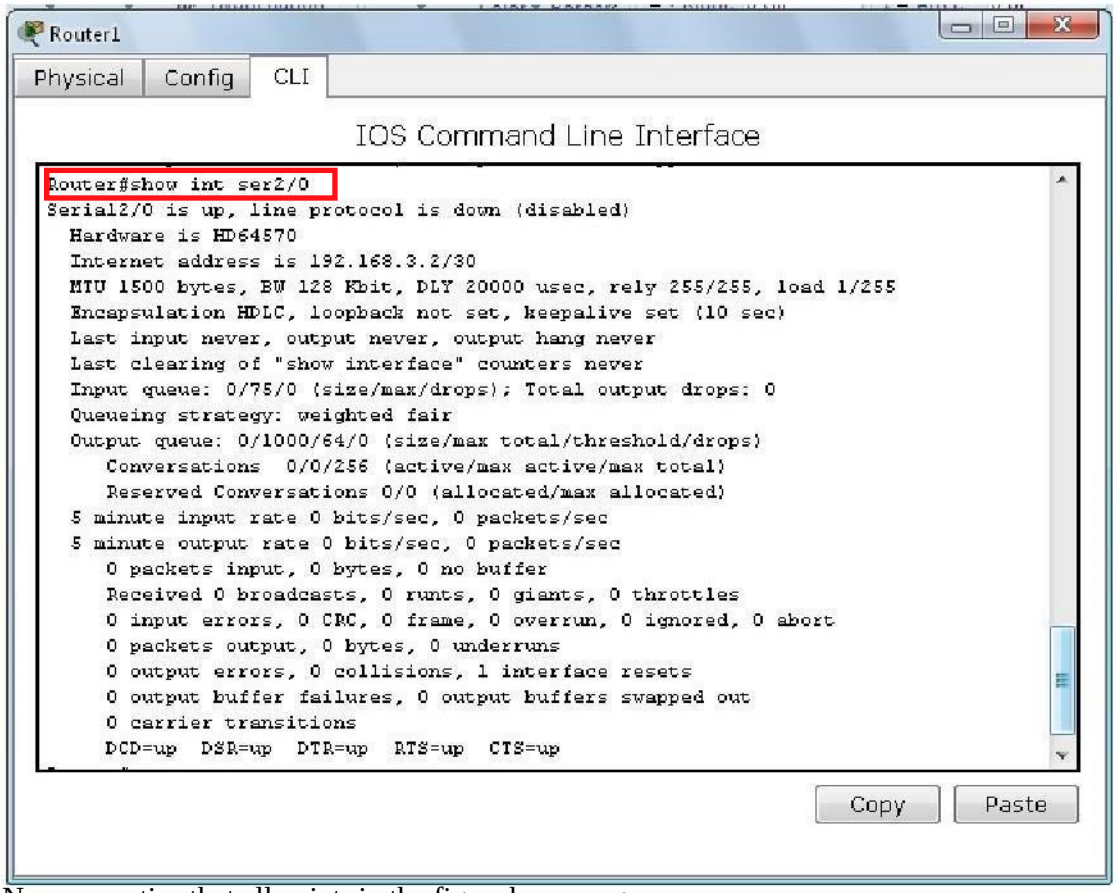


For the second router :

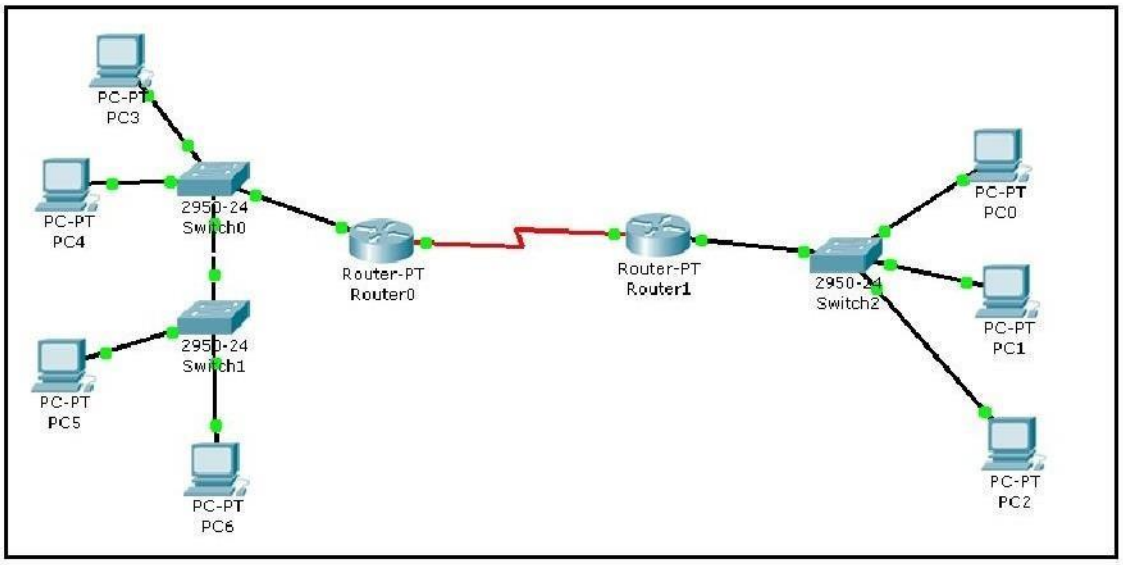


To make sure of IP's :

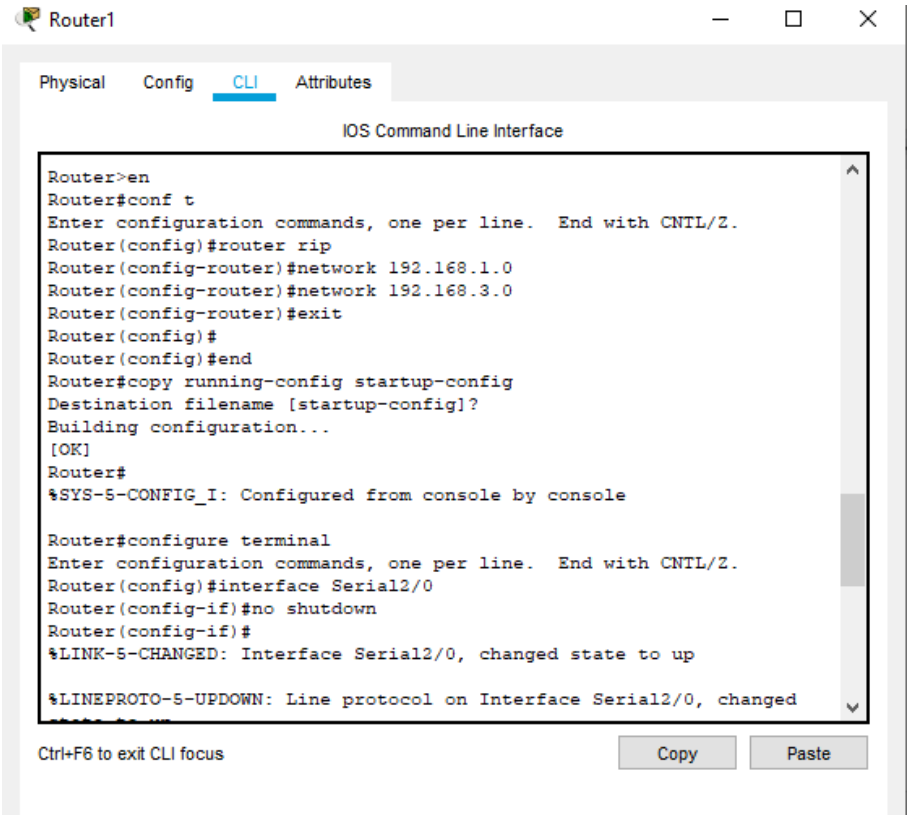

Router1
Physical
Config
CLI
IOS Command Line Interface
Router$show int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
Hardware is Lance, address is 0030.f2a9.d681 (bia 0030.f2a9.d681)
Internet address is 192.168.1.1/24
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
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

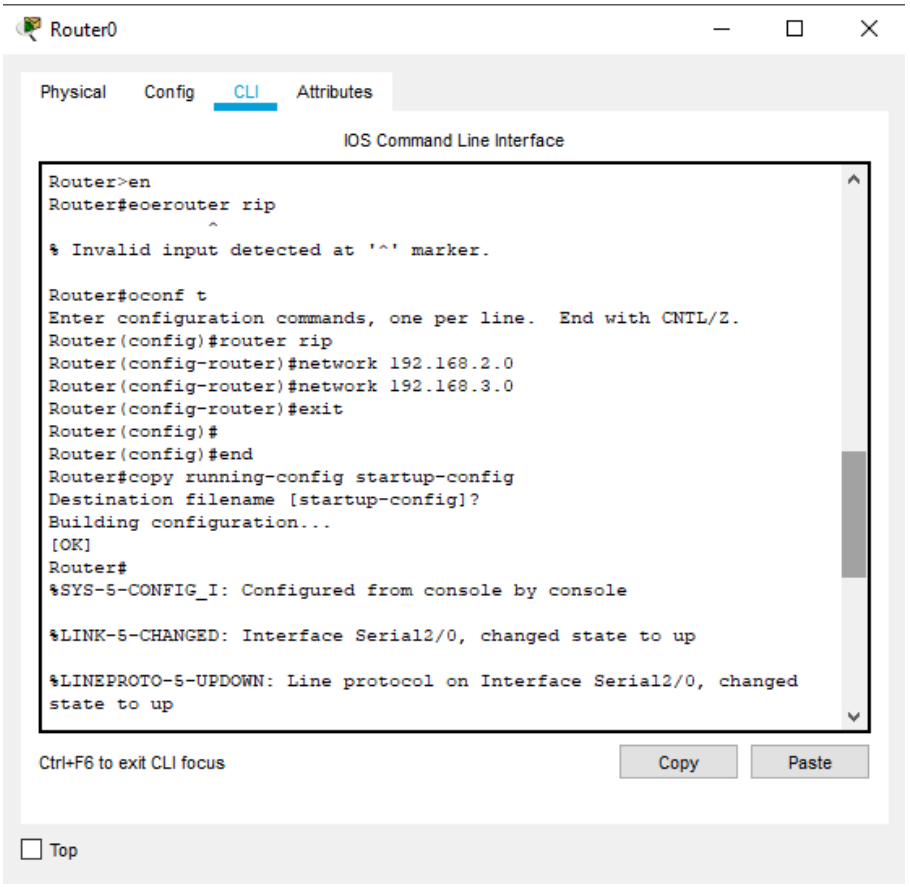


Now we notice that all points in the figure became green :

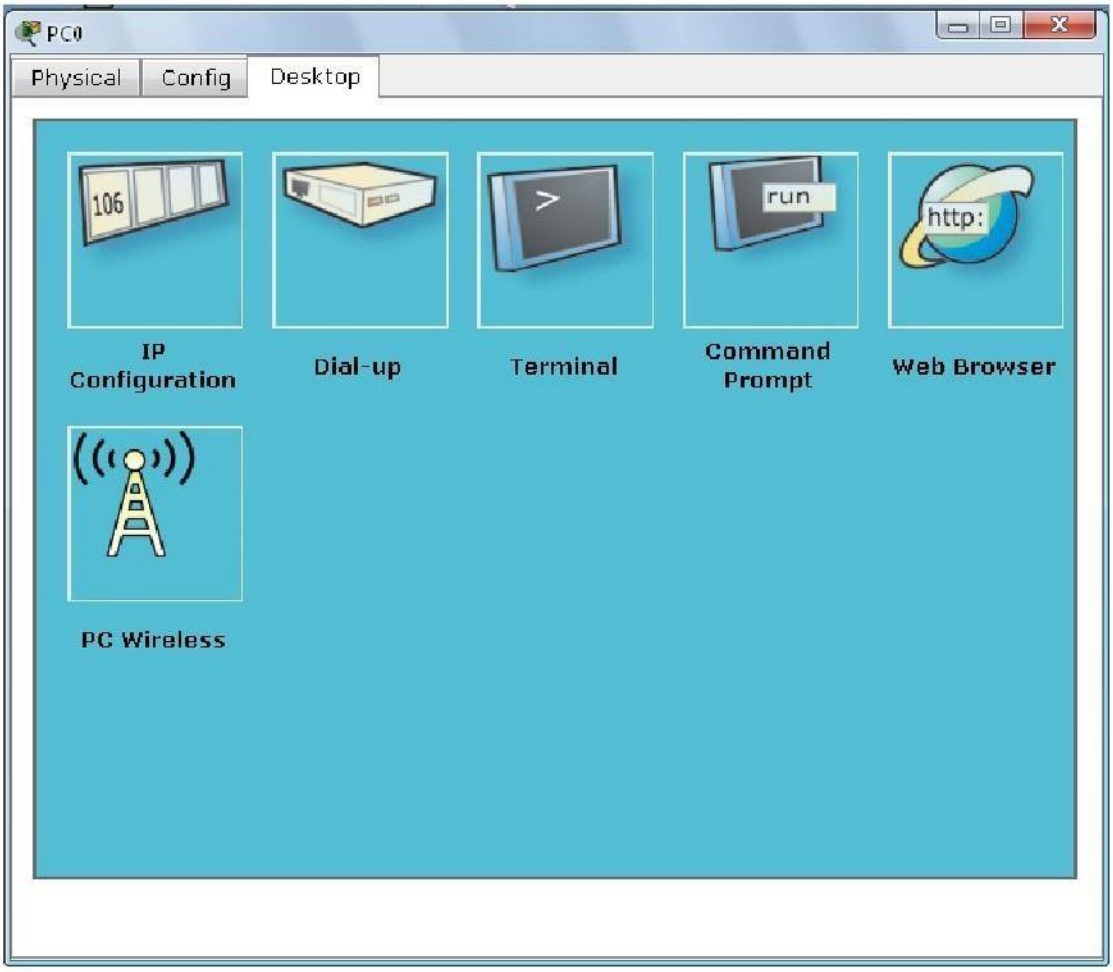


Now, configure RIP on both networks





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**

- OSPF
- EIGRP

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

Lab No. 11

CLO4,1,3

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}

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

- Using Wild Cards
  - 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

- Wild Card Masks
  - 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

- **What are wild card masks**
  - 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

- **Examples**
  - 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

- 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

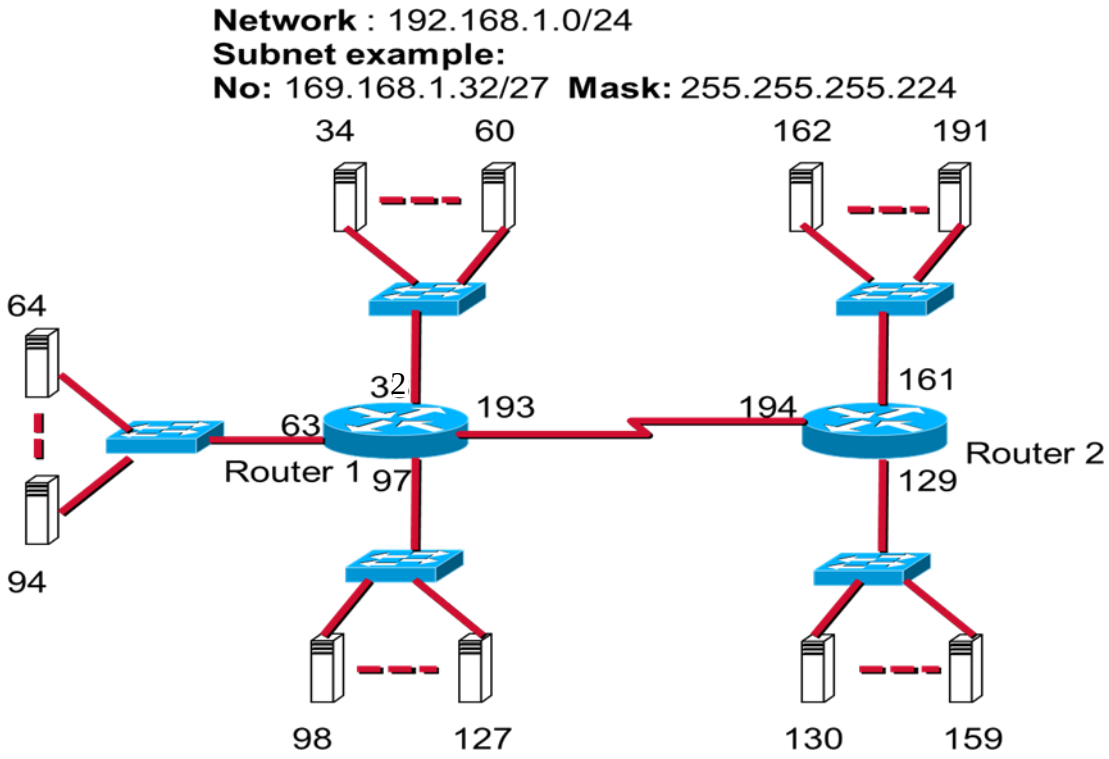
Task 1:

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
```

CSE-331 Computer Networks
Lab No. 12

CLO1,2

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

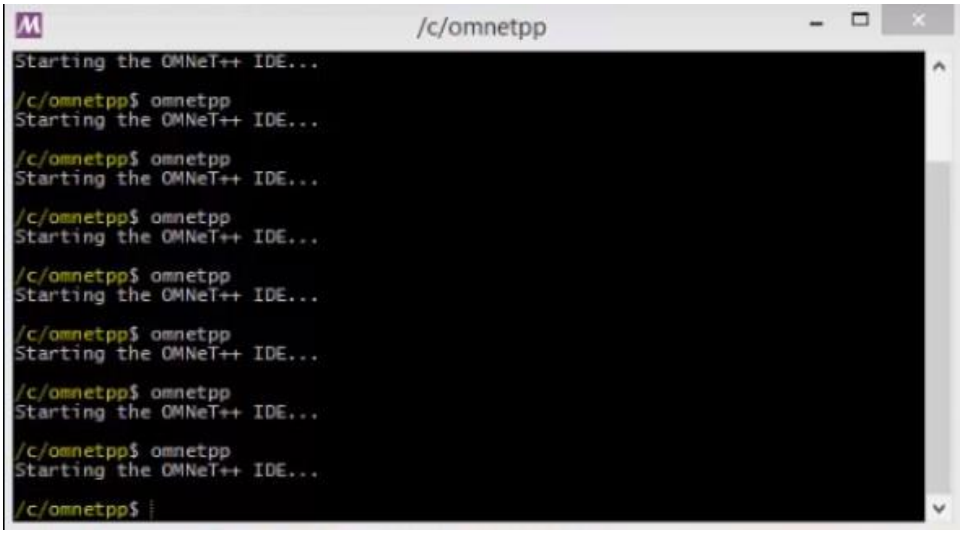
There are several ways to install the INET Framework:

- 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.

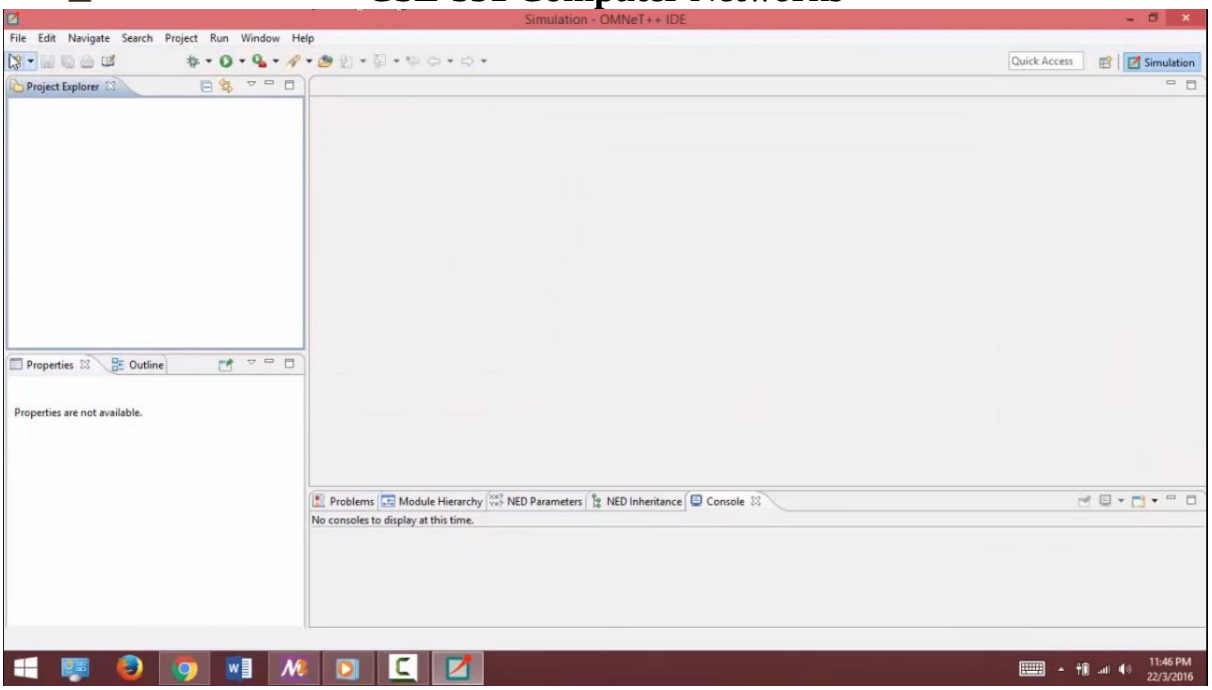
First project on OMNET++

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

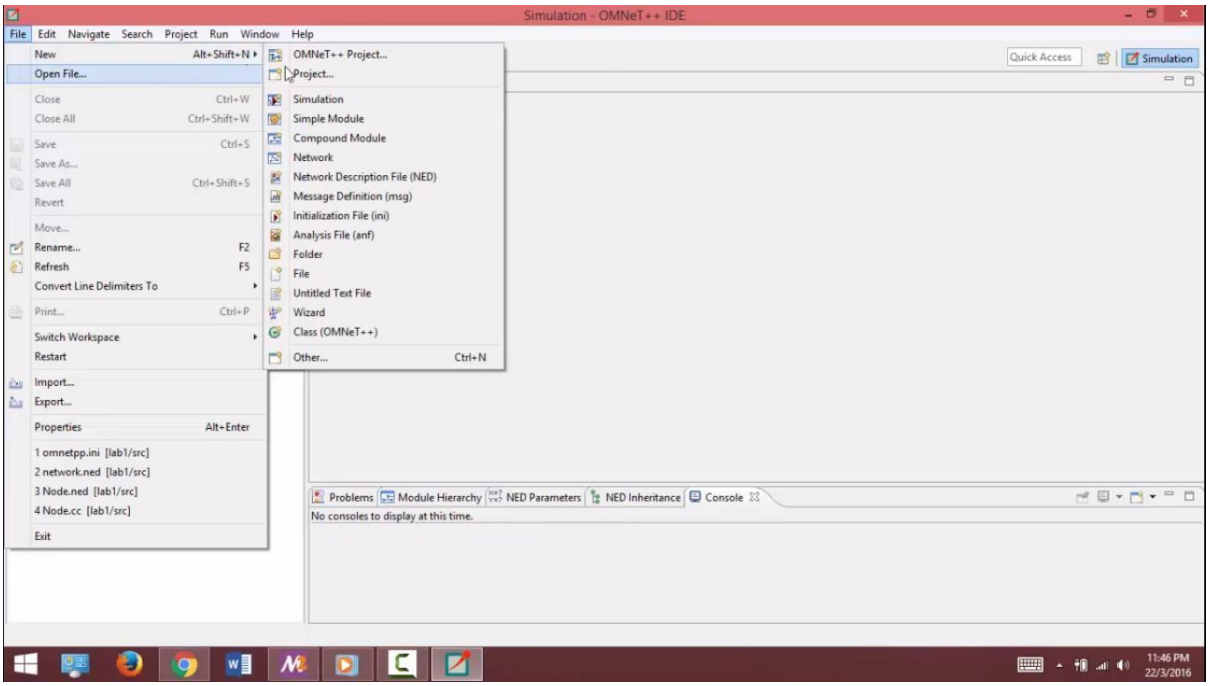
Write omnetpp in the cmd of omnet to launch the interface



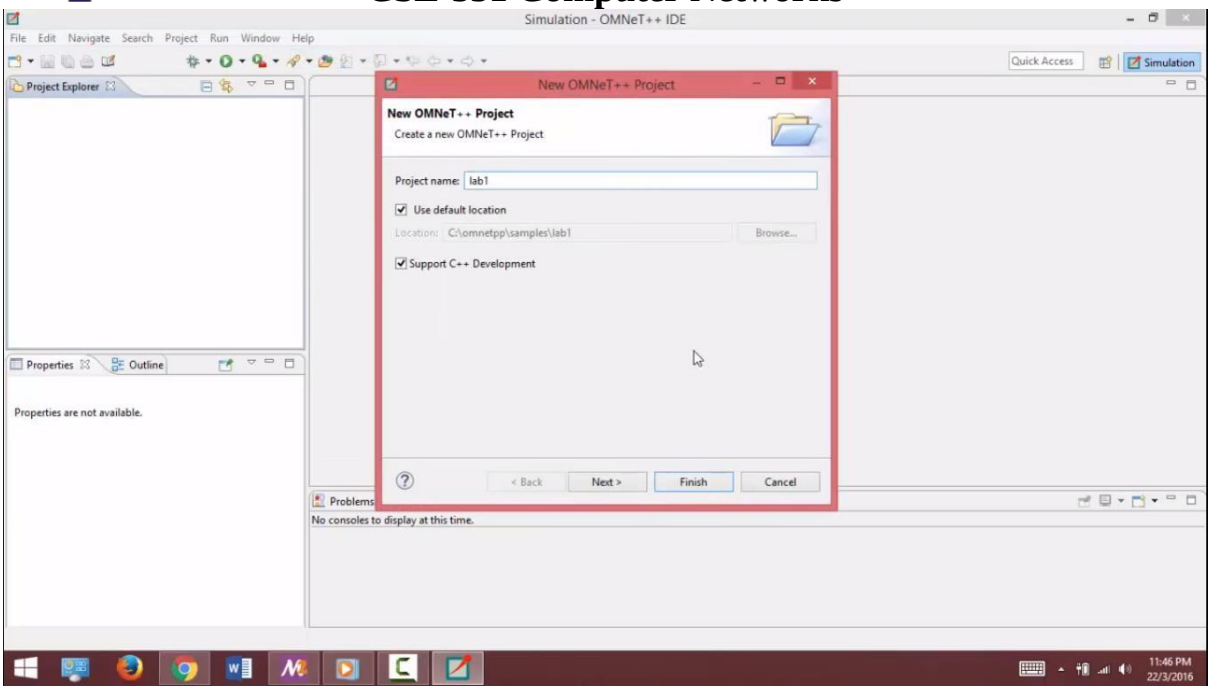
This is the interface of OMNET++



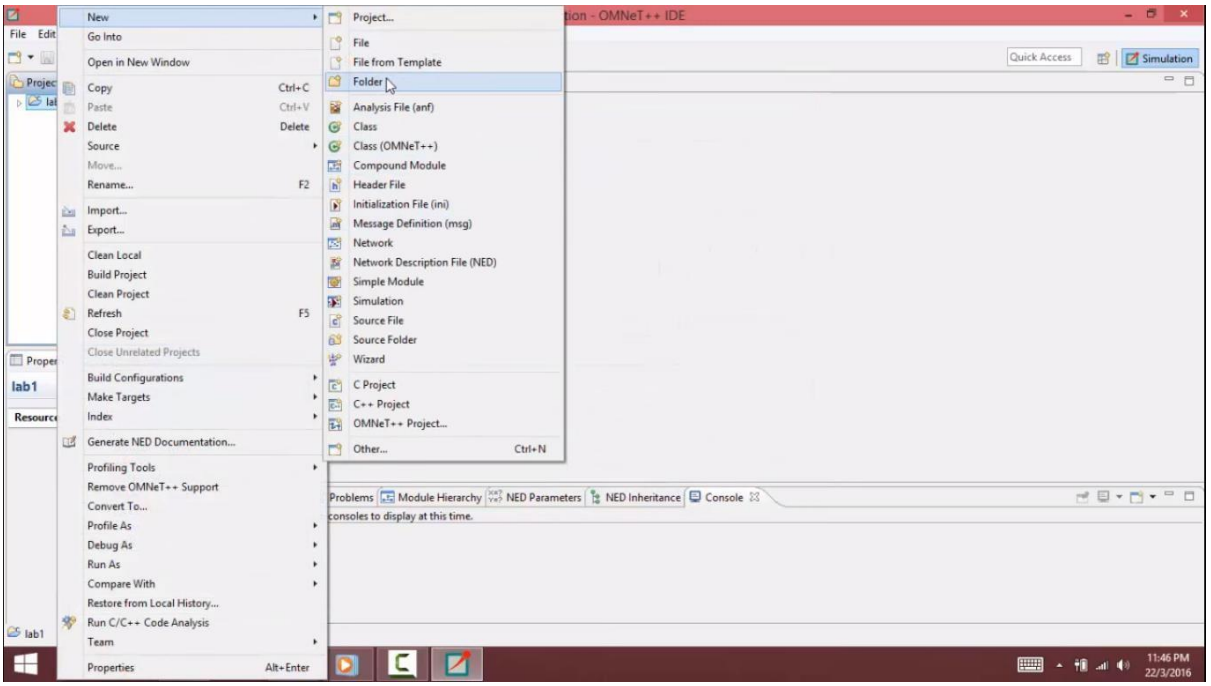
Select file->open file-> OMNet++ Project



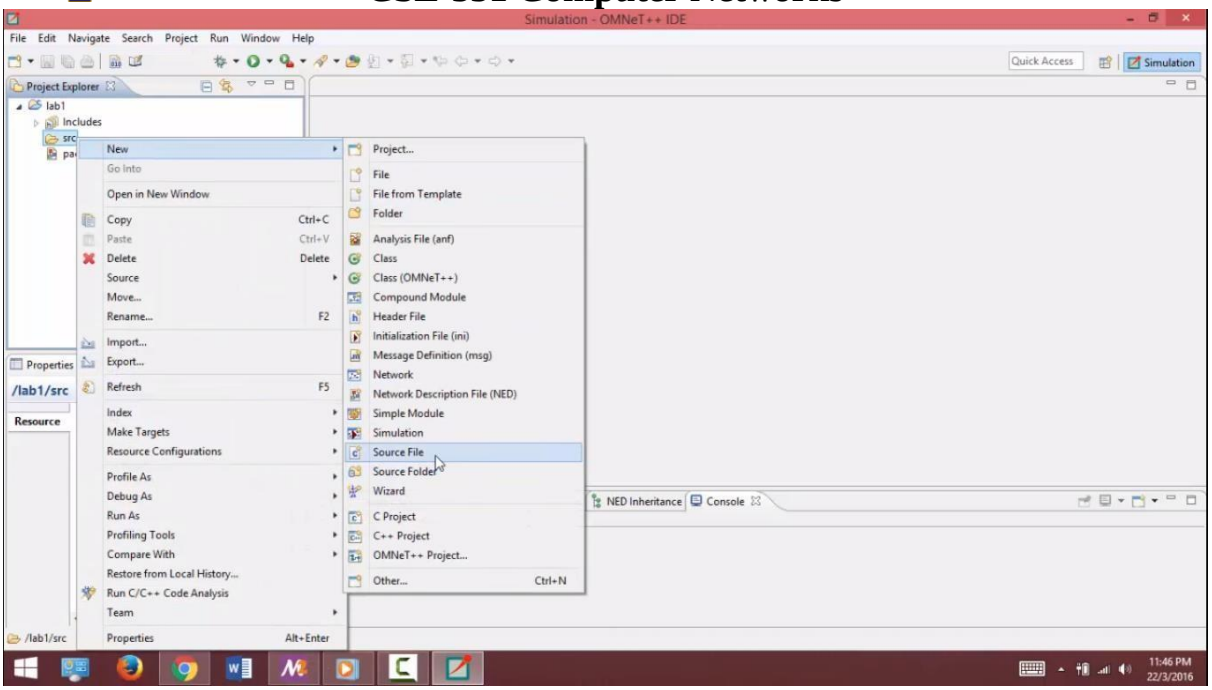
Write name of your project and click Next and then Finish



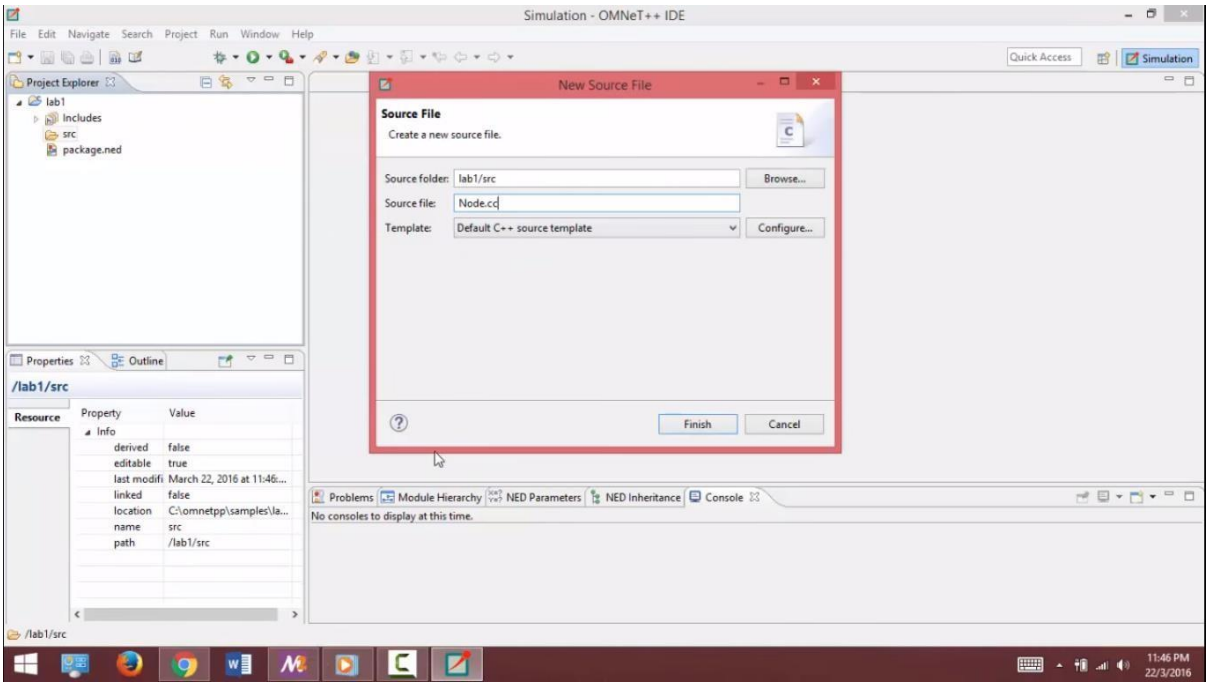
Right click on the name of project and make a new folder, give it a name and click Finish.



Expand the project select the folder and create new source file



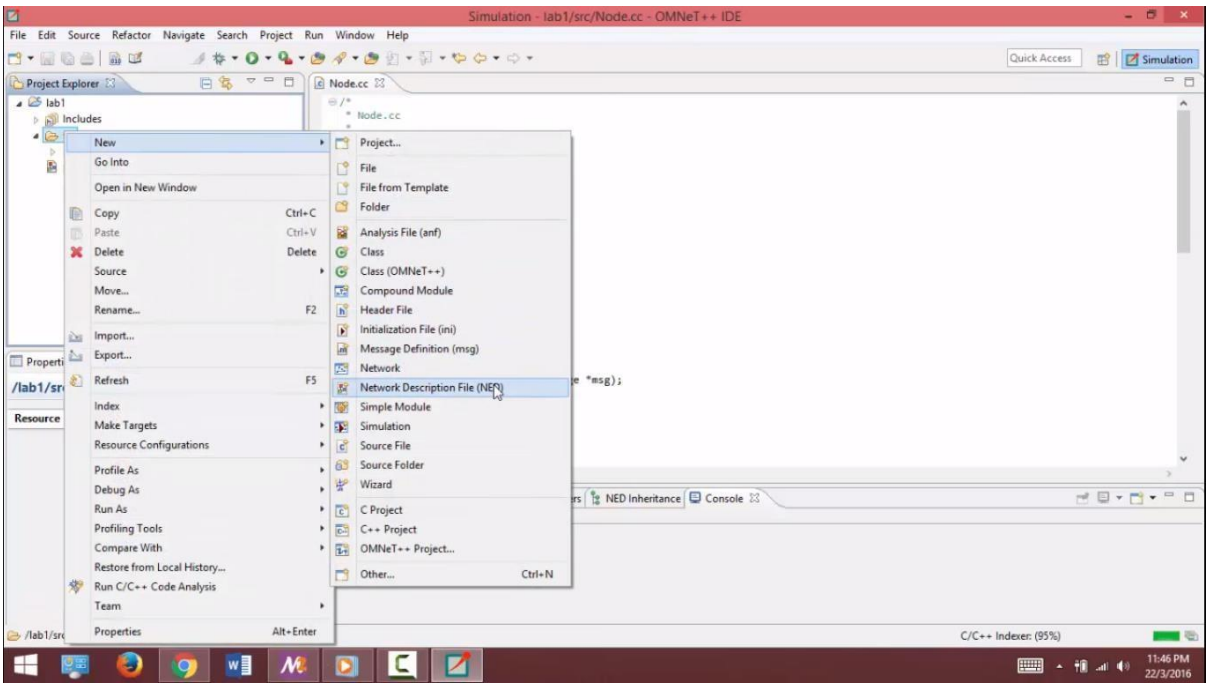
Name the file with extension of *.cc* and click Finish.



Click on package.ned and add following code there

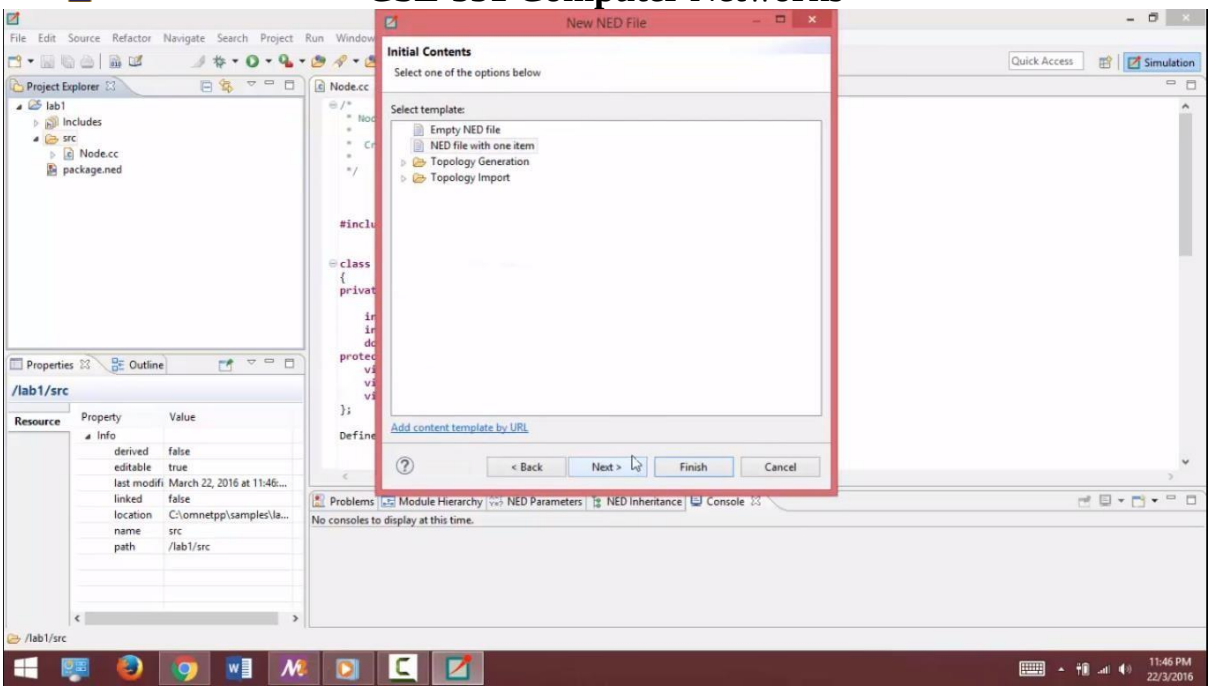
CSE-331 Computer Networks

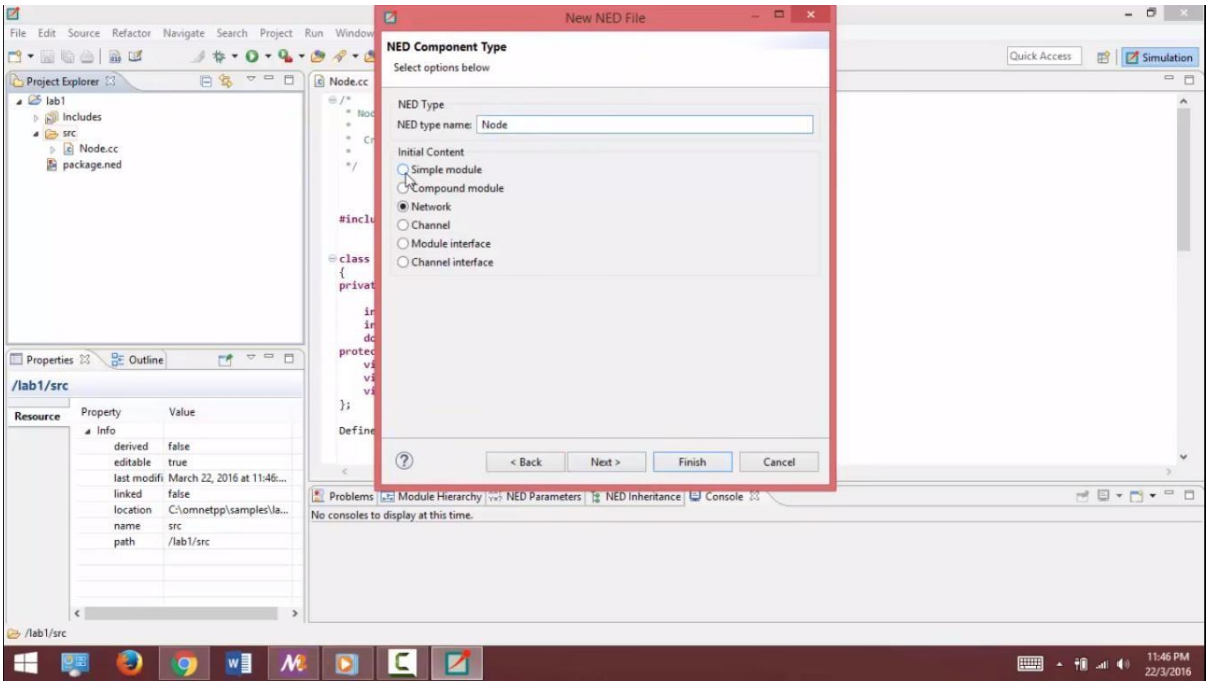
```
#include<omnetpp.h>


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);
}
```



Course Instructor: Mr. Amjad Majeed

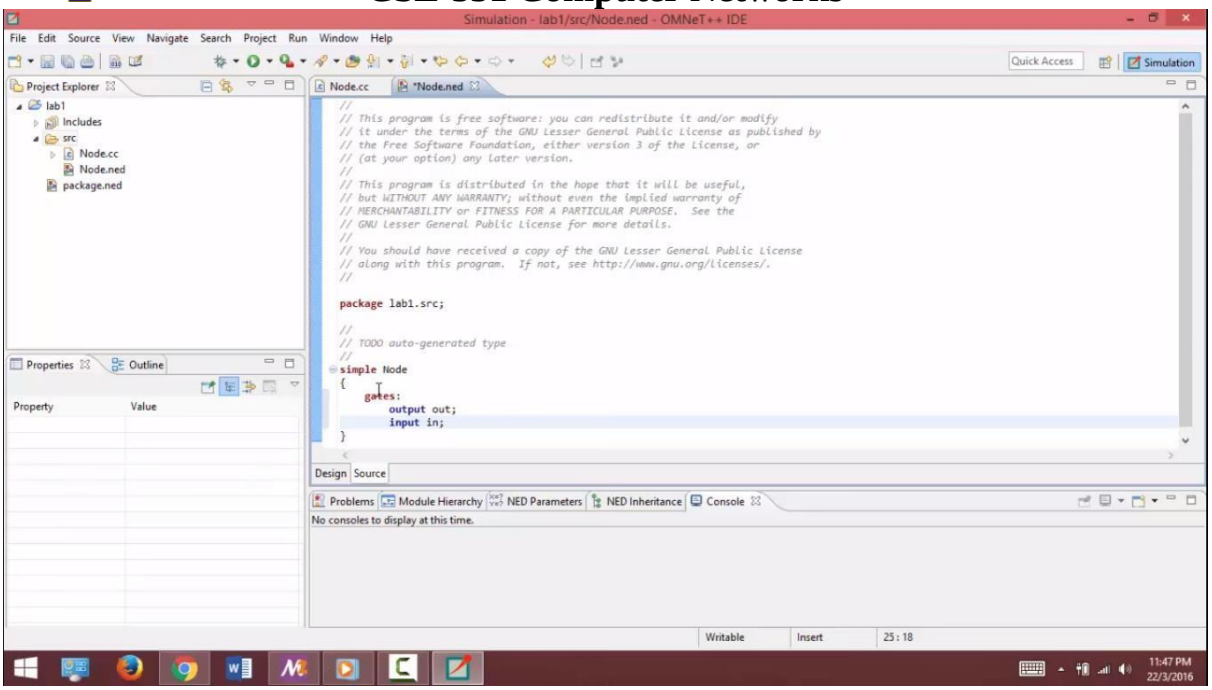
CSE-331 Computer Networks

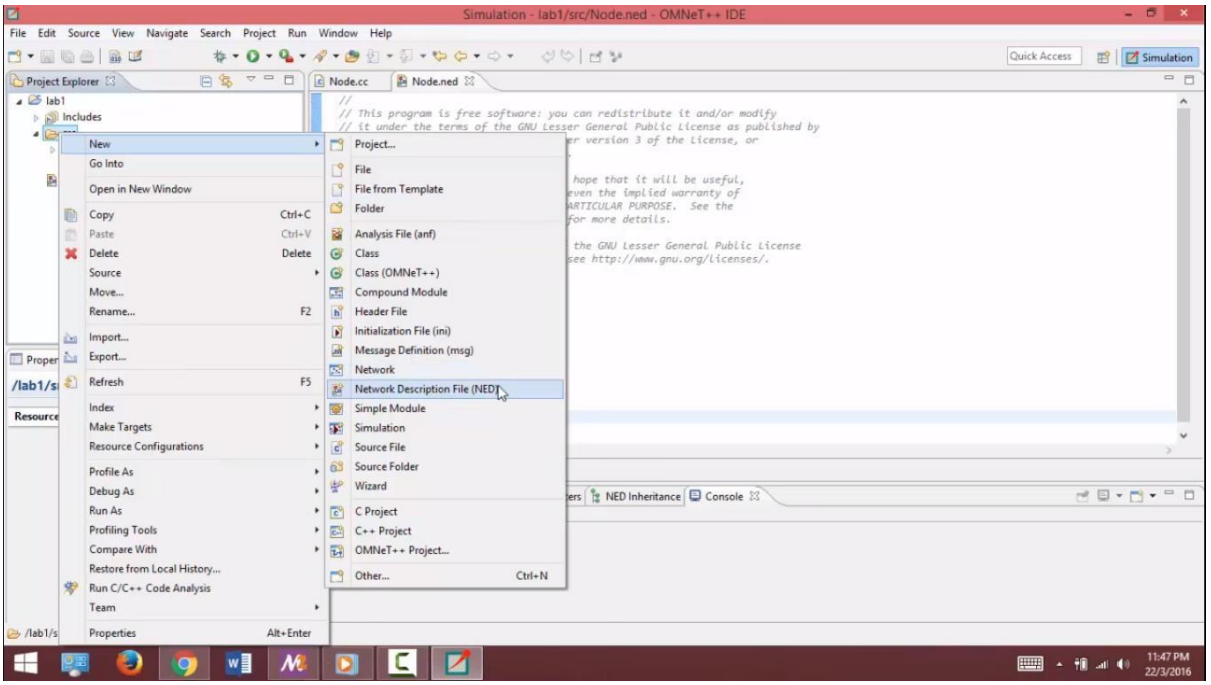




Course Instructor: Mr. Amjad Majeed

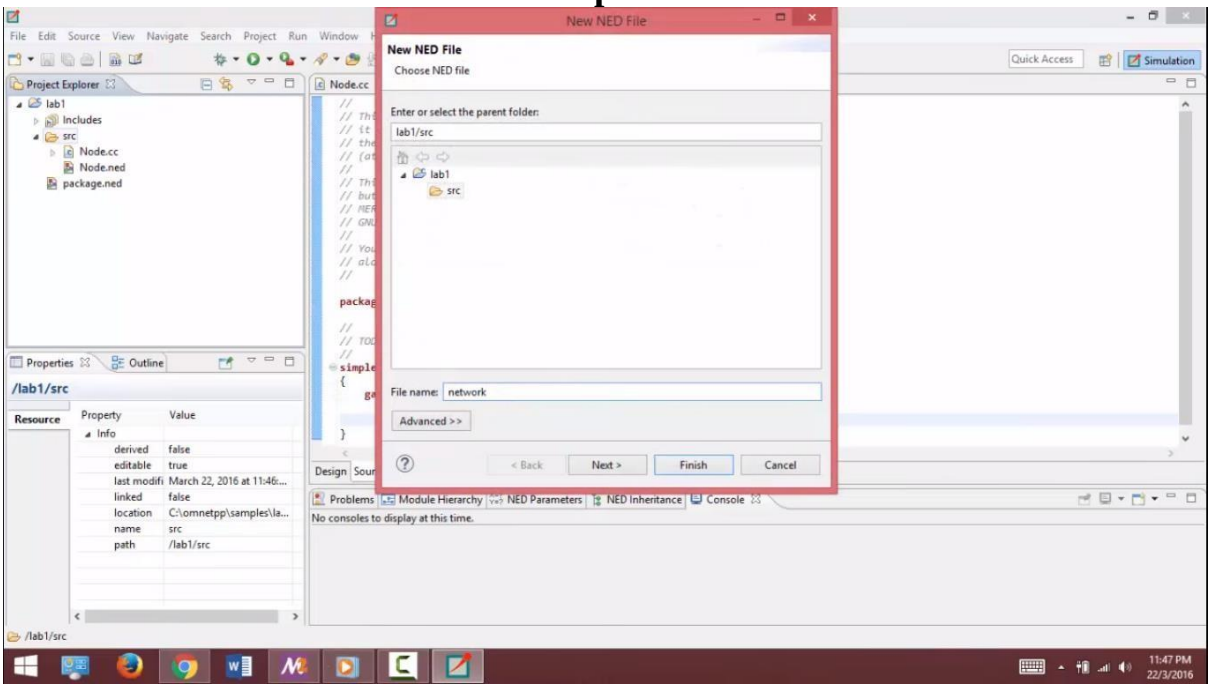
CSE-331 Computer Networks

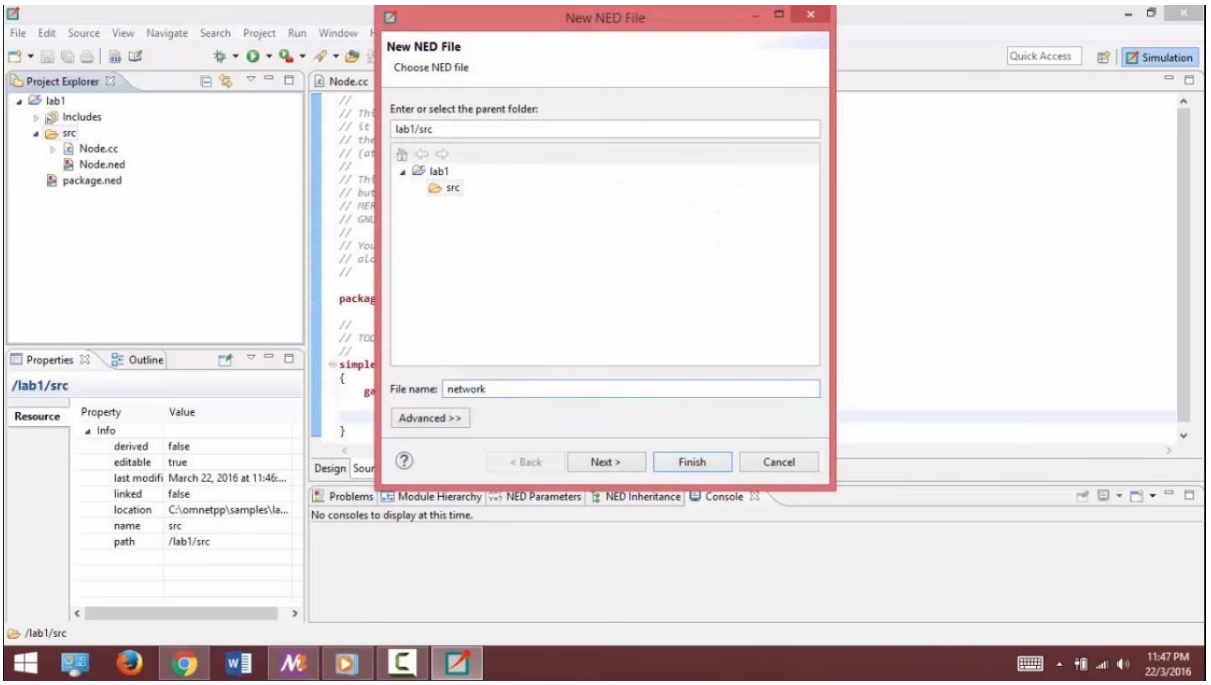




Course Instructor: Mr. Amjad Majeed

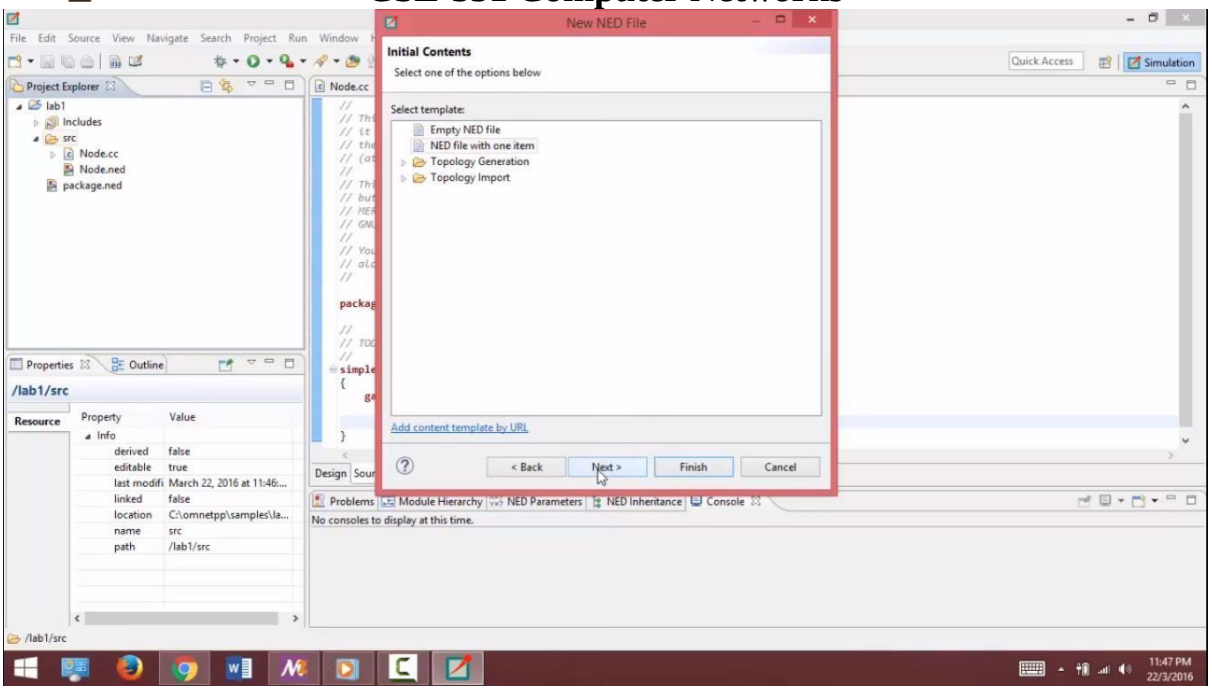
CSE-331 Computer Networks

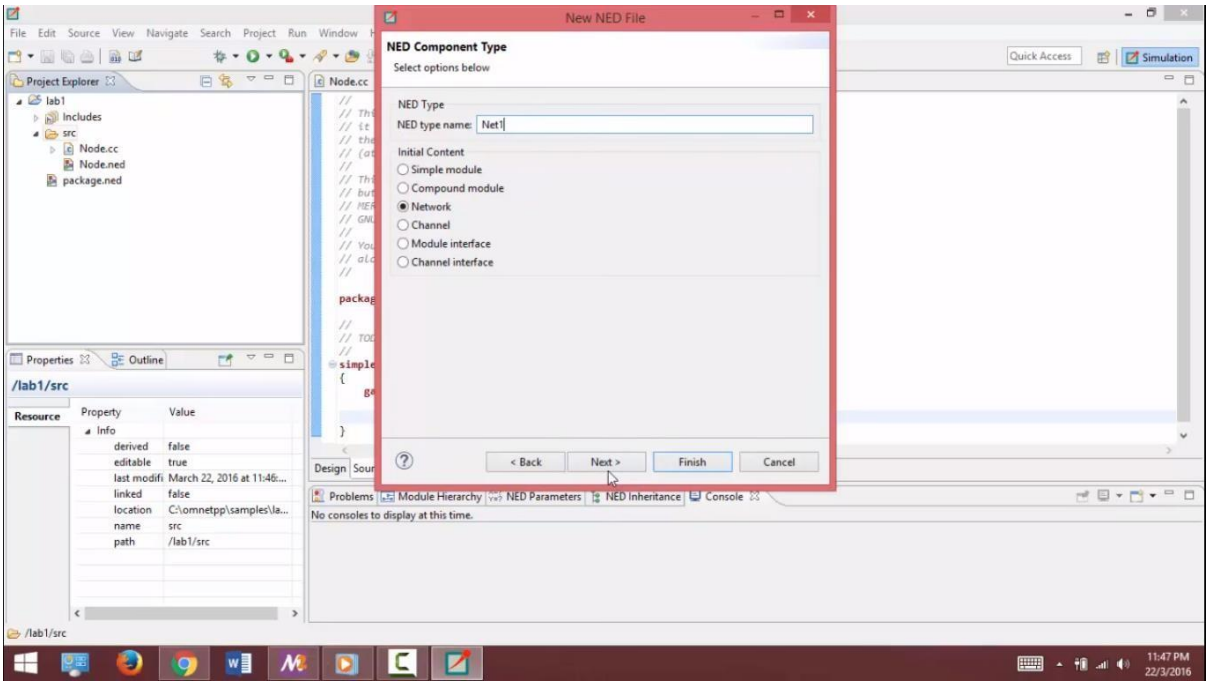




Course Instructor: Mr. Amjad Majeed

CSE-331 Computer Networks

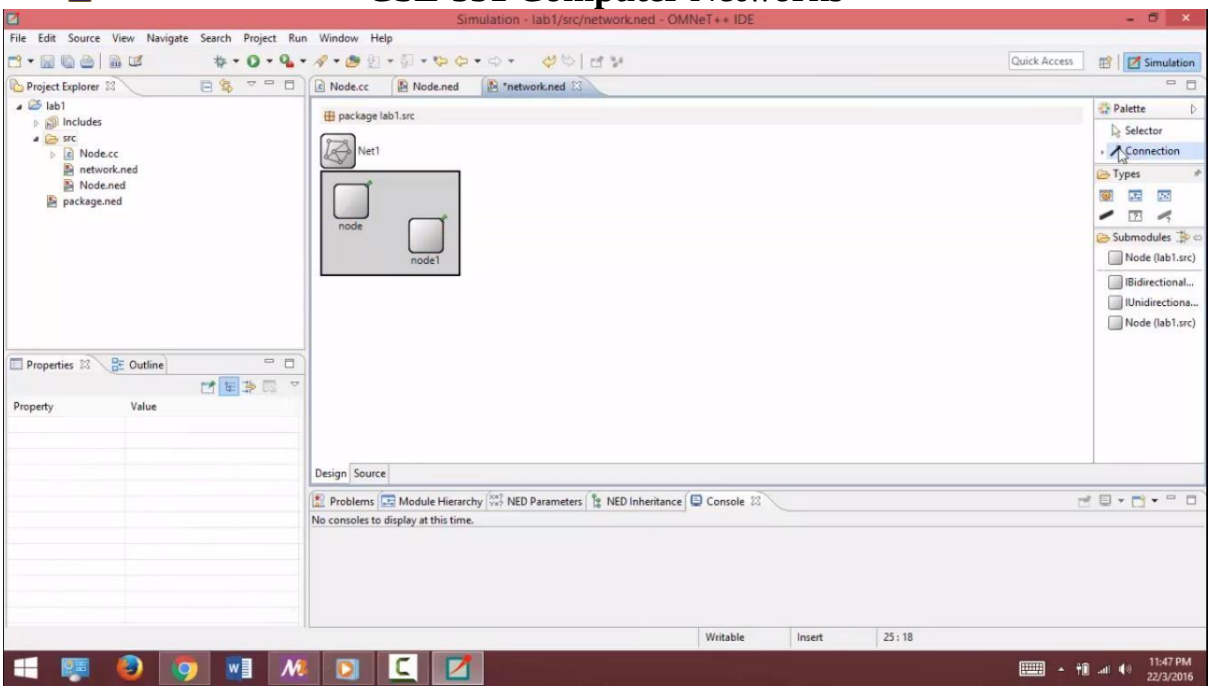


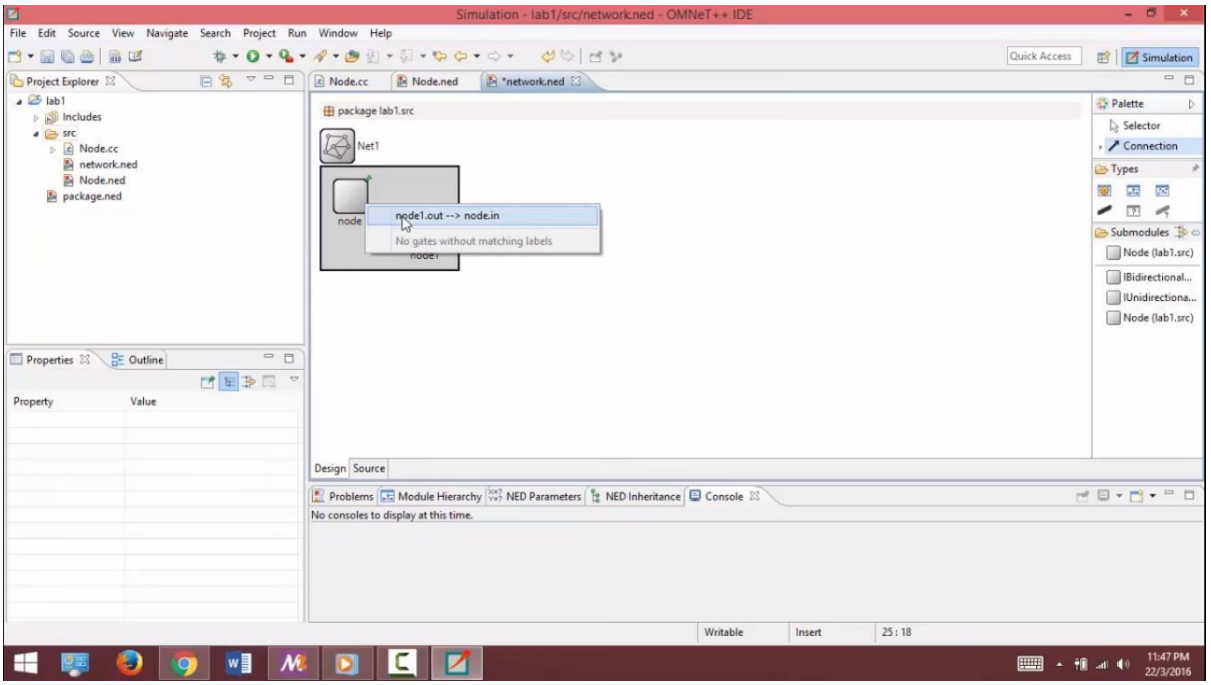


Drag Node(lab1.src) (twice) into the network and connect them both ways with the connection as shown below.

Course Instructor: Mr. Amjad Majeed

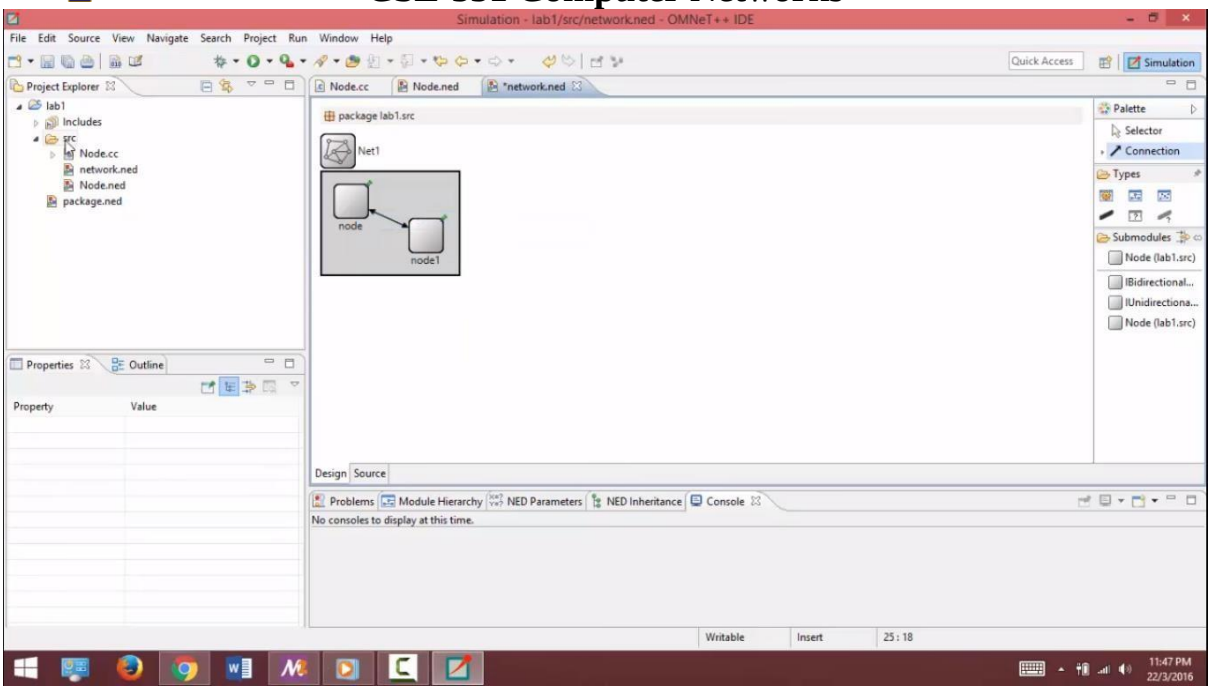
CSE-331 Computer Networks

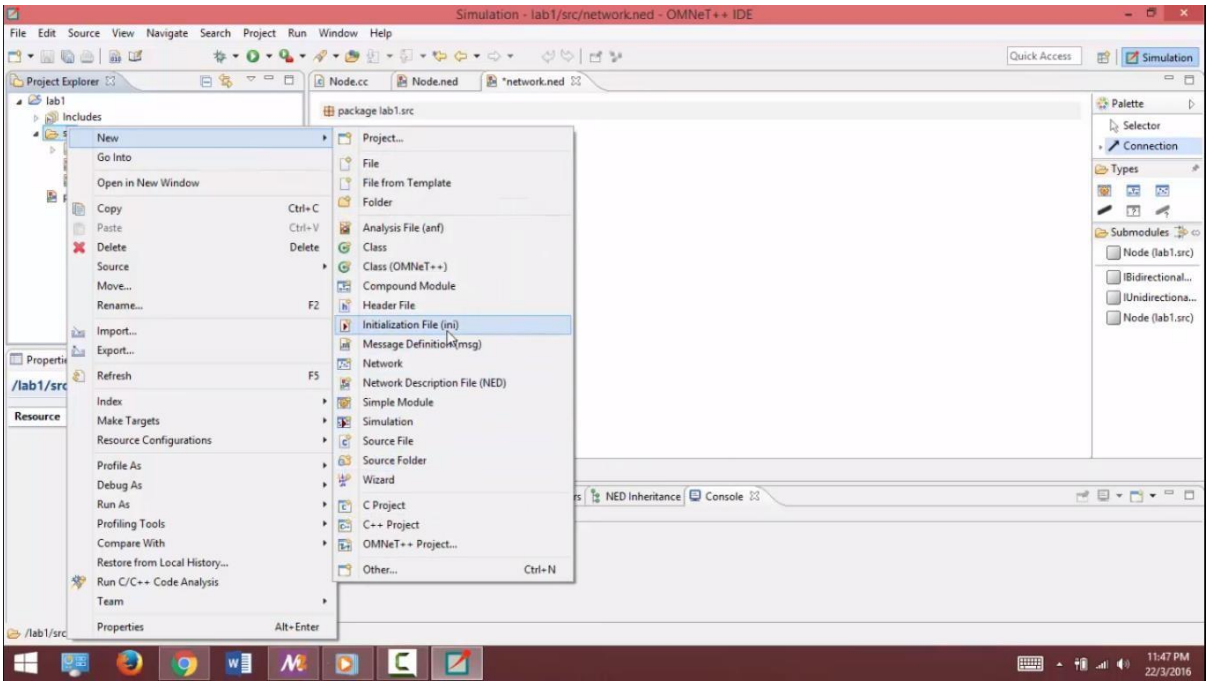




Course Instructor: Mr. Amjad Majeed

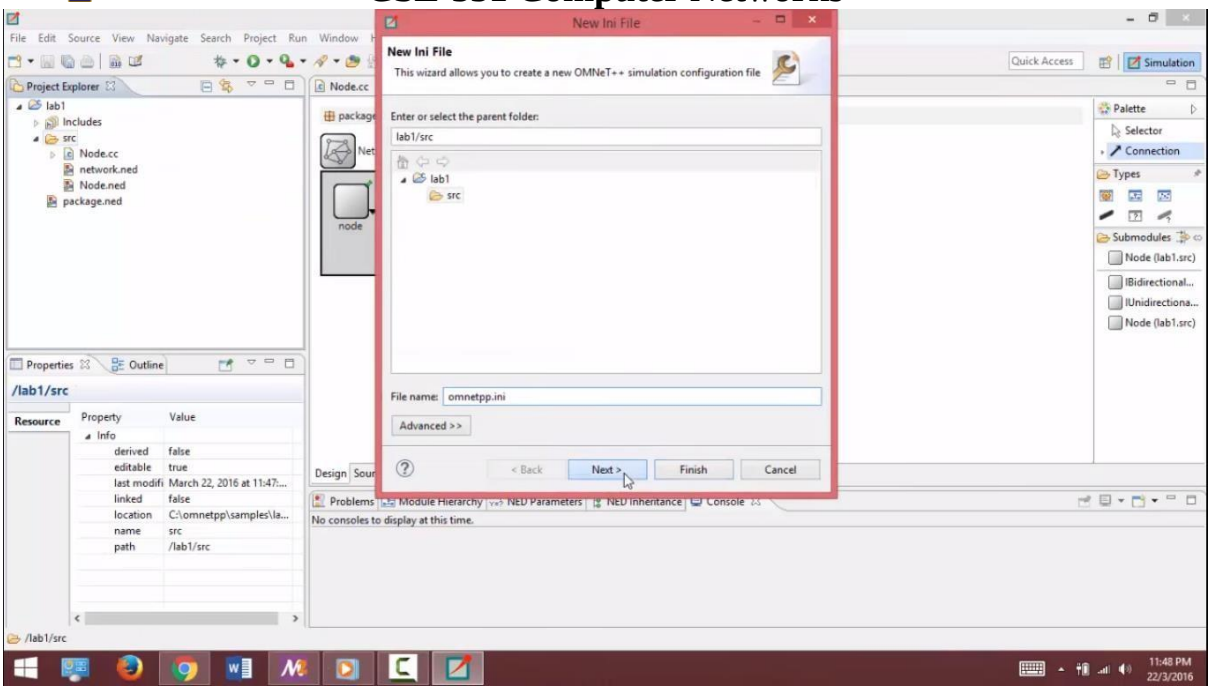
CSE-331 Computer Networks

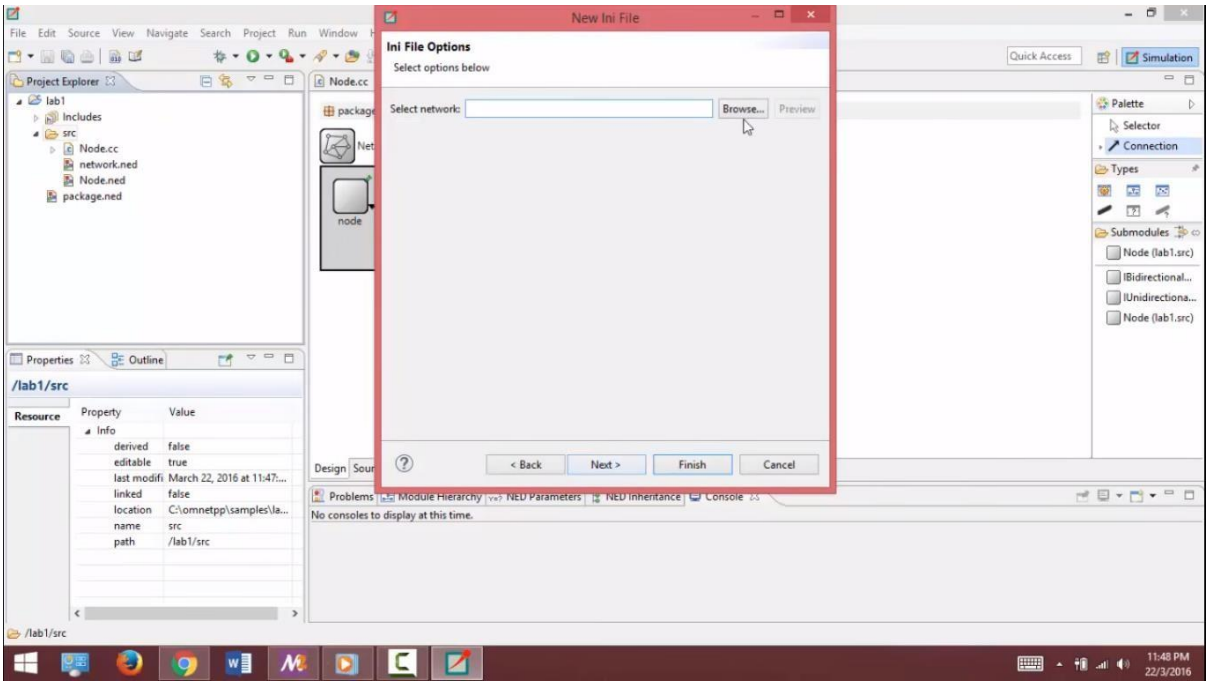




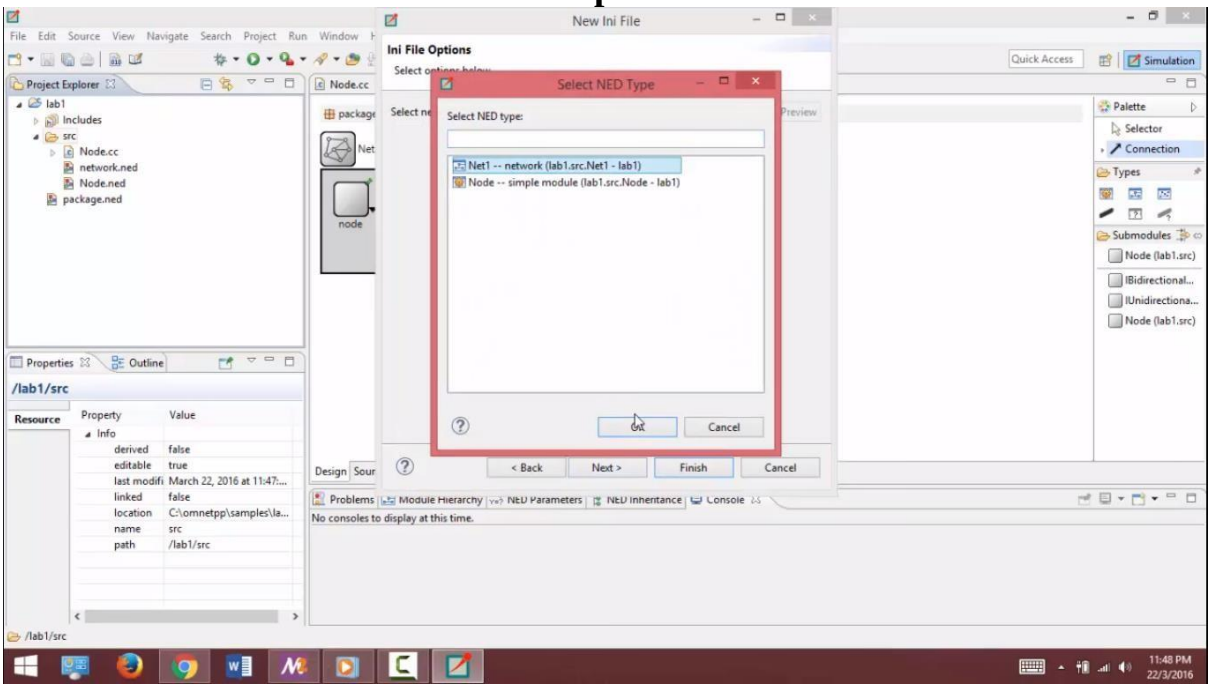
Course Instructor: Mr. Amjad Majeed

CSE-331 Computer Networks

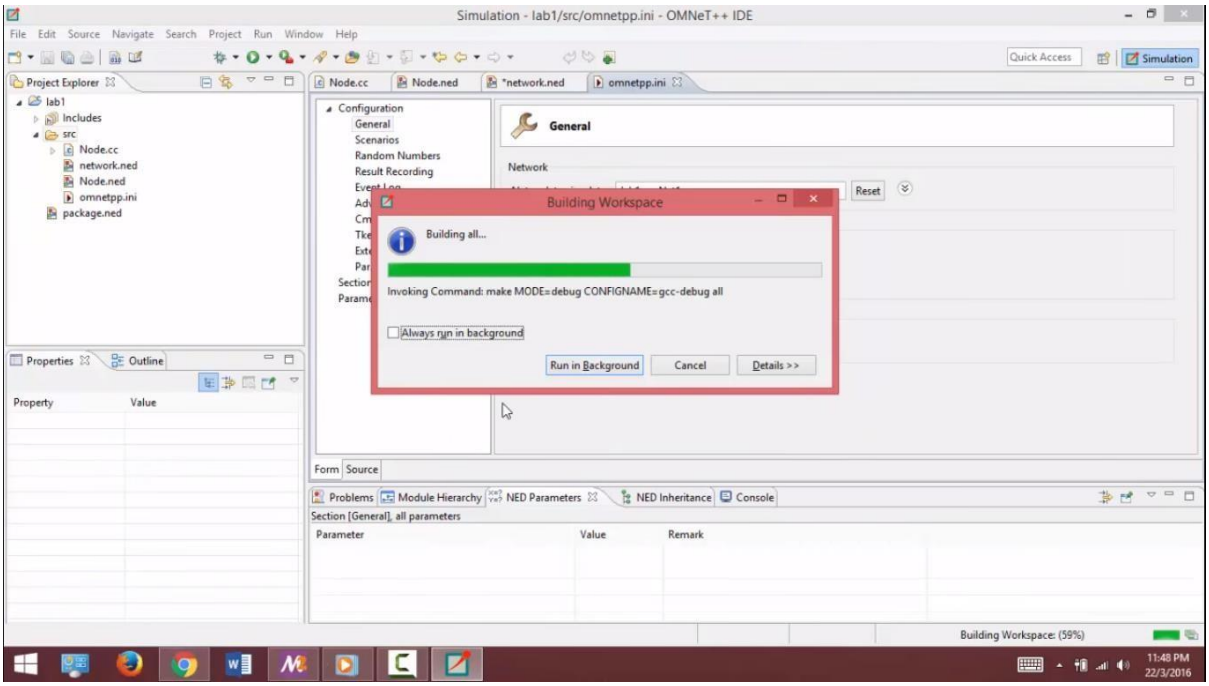




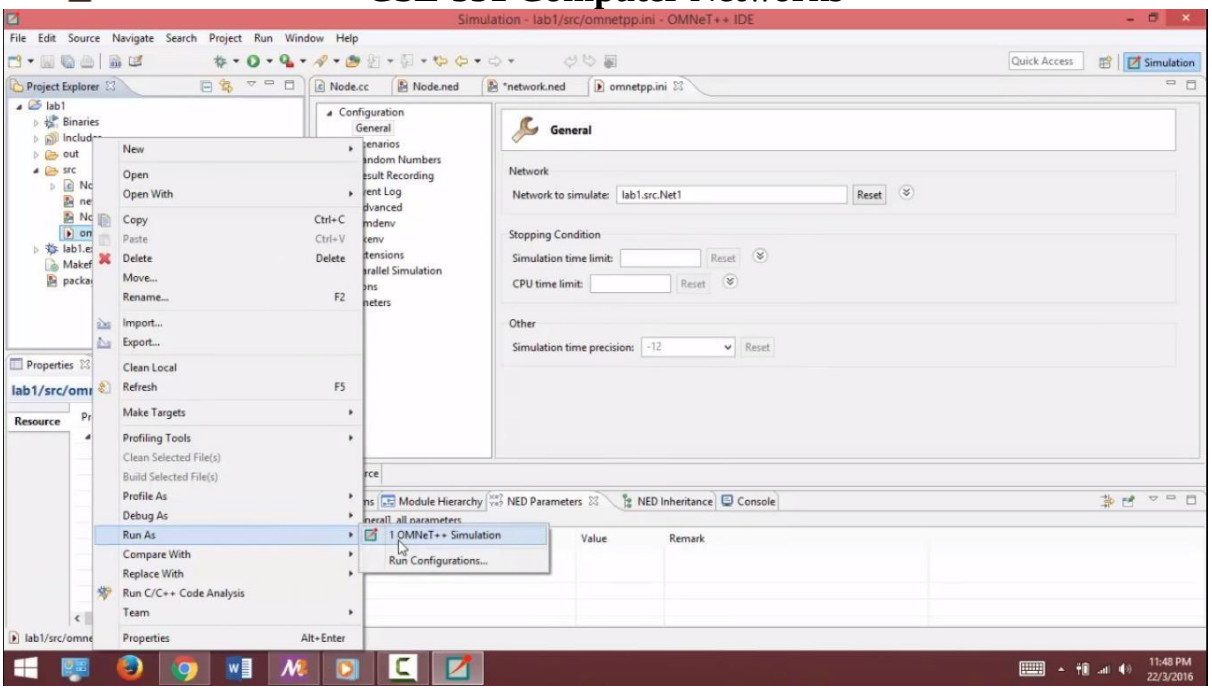
Course Instructor: Mr. Amjad Majeed

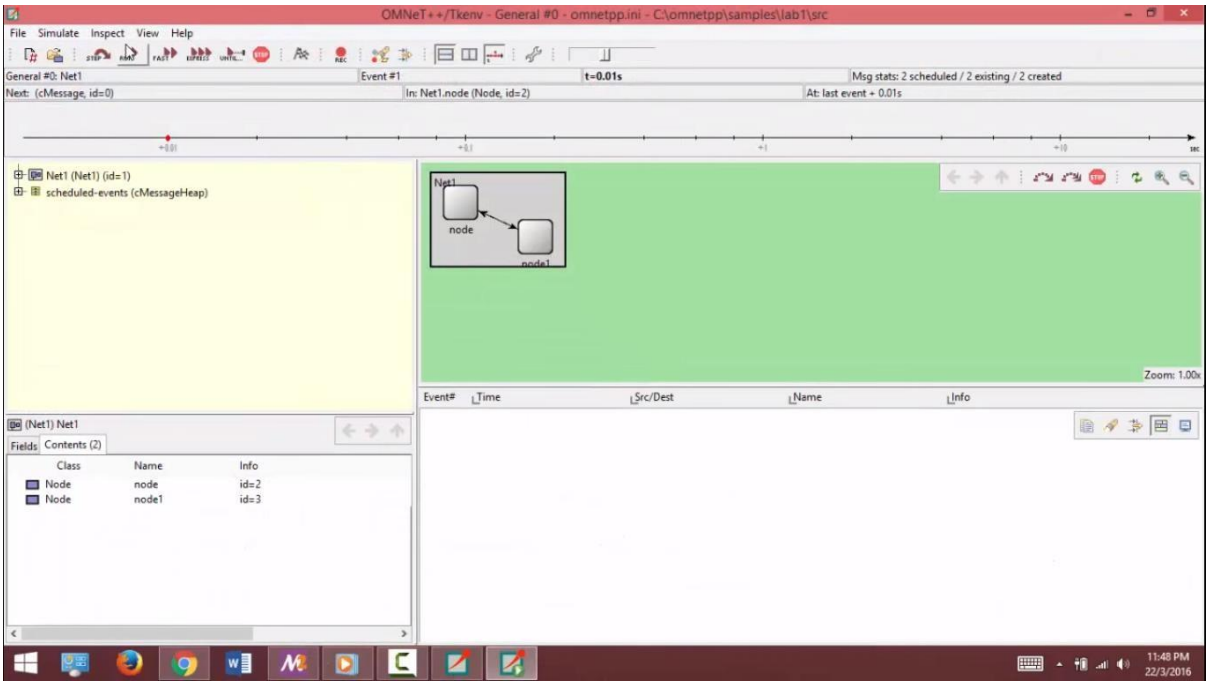


Now click ok, then Next and then Finish.

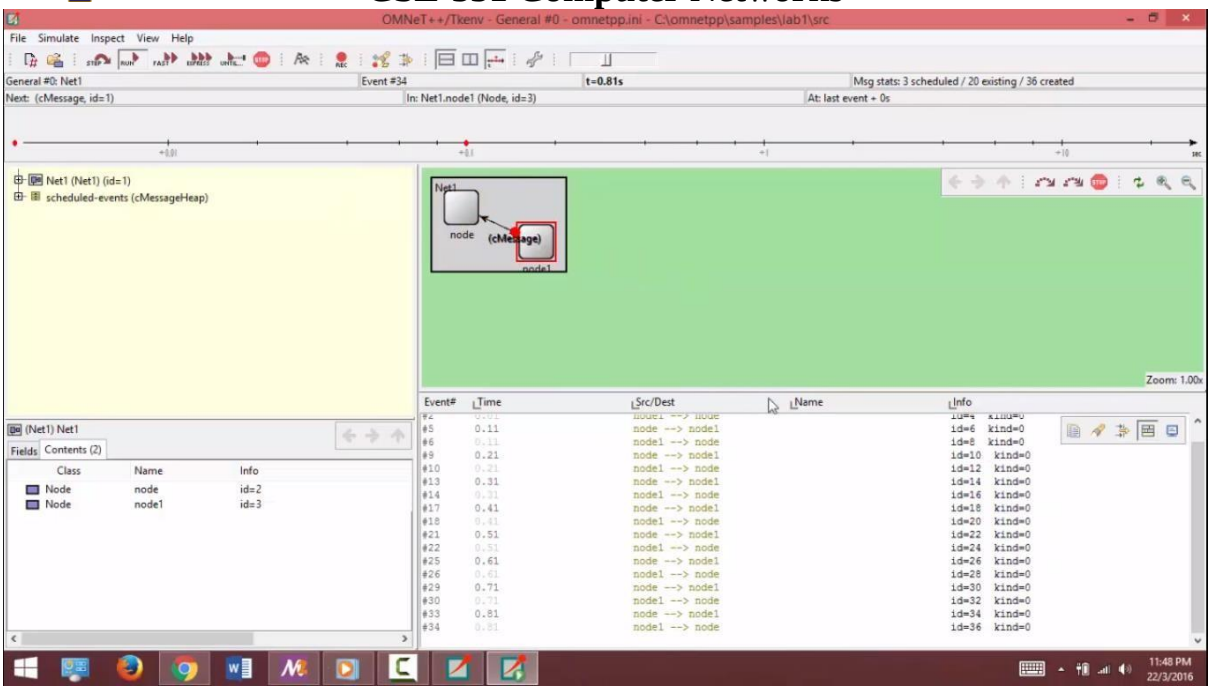


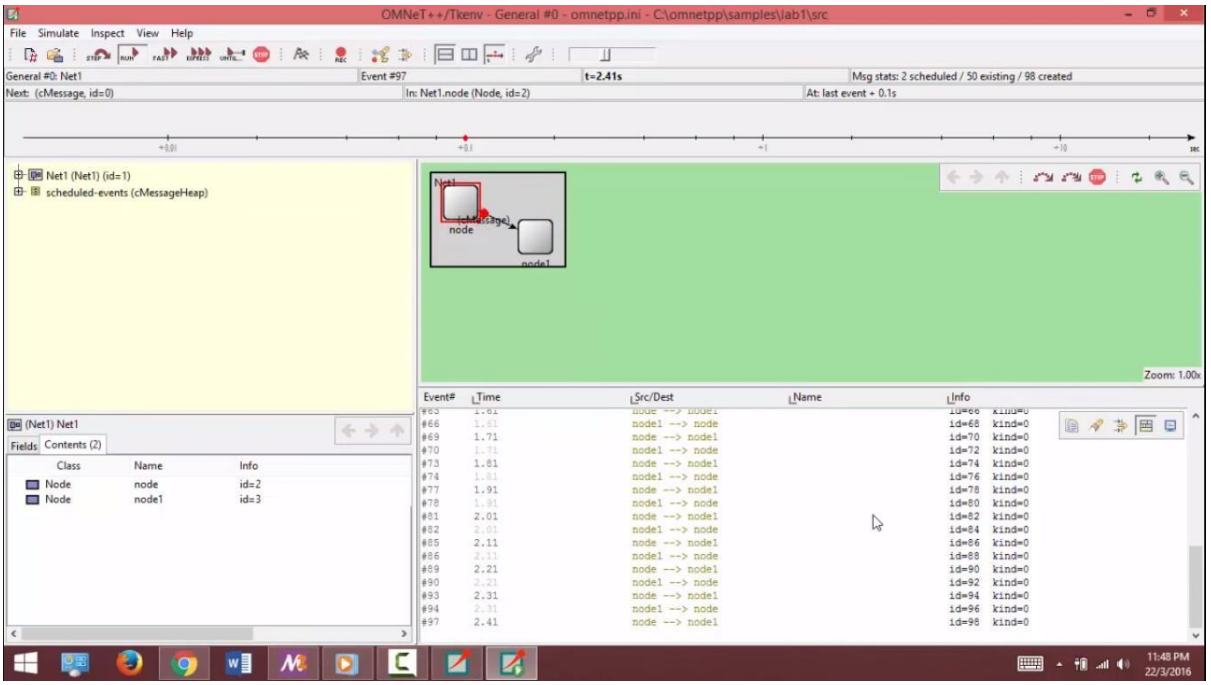
CSE-331 Computer Networks





Course Instructor: Mr. Amjad Majeed





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