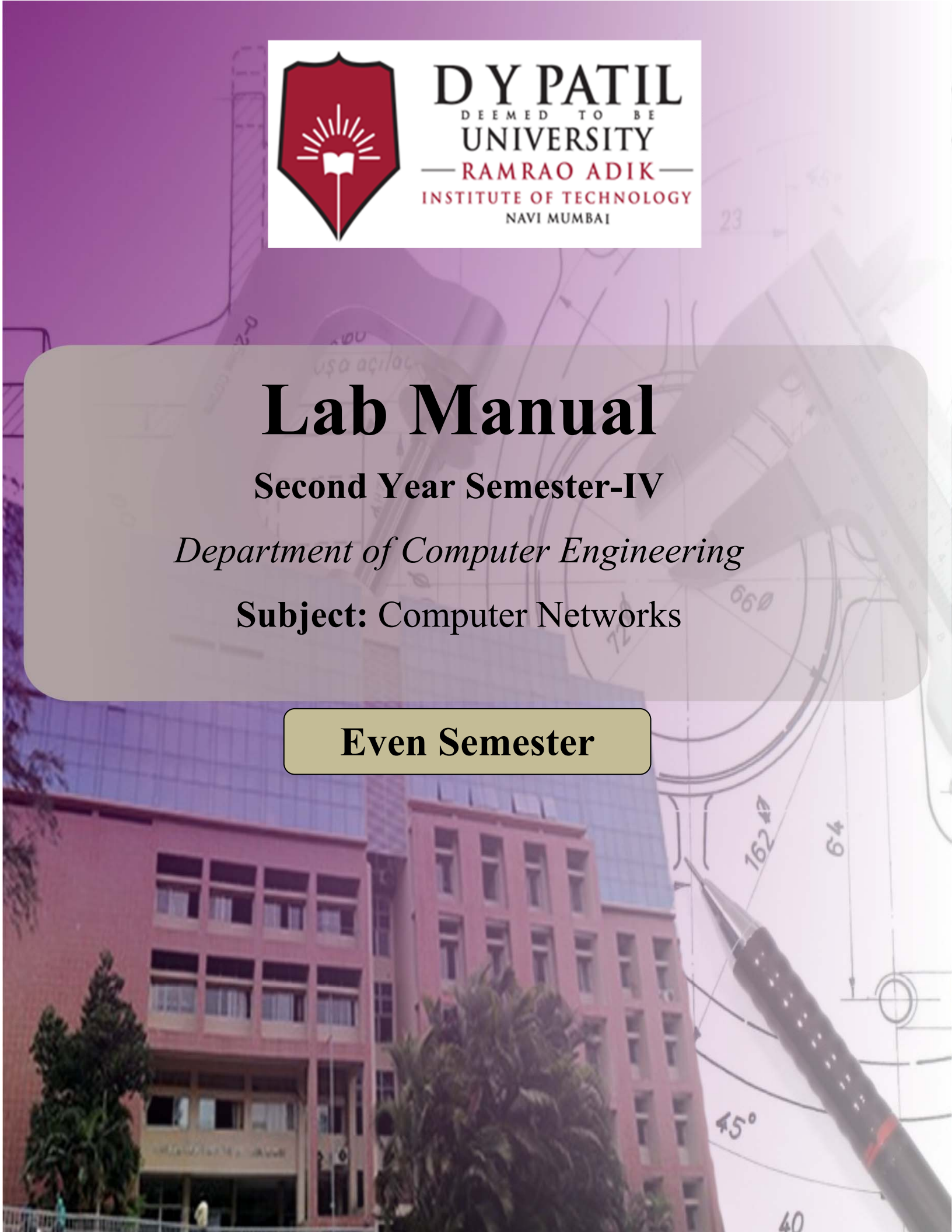


D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Lab Manual

Second Year Semester-IV

Department of Computer Engineering

Subject: Computer Networks

Even Semester

Institute Vision, Mission & Quality Policy

Vision

To foster and permeate higher and quality education with value added engineering, technology programs, providing all facilities in terms of technology and platforms for all round development with societal awareness and nurture the youth with international competencies and exemplary level of employability even under highly competitive environment so that they are innovative adaptable and capable of handling problems faced by our country and world at large.

Mission

The Institution is committed to mobilize the resources and equip itself with men and materials of excellence thereby ensuring that the Institution becomes pivotal center of service to Industry, academia, and society with the latest technology. RAIT engages different platforms such as technology enhancing Student Technical Societies, Cultural platforms, Sports excellence centers, Entrepreneurial Development Center and Societal Interaction Cell. To develop the college to become an autonomous Institution & deemed university at the earliest with facilities for advanced research and development programs on par with international standards. To invite international and reputed national Institutions and Universities to collaborate with our institution on the issues of common interest of teaching and learning sophistication.

Quality Policy

ज्ञानधीनं जगत् सर्वम।

Knowledge is supreme.

Our Quality Policy

It is our earnest endeavour to produce high quality engineering professionals who are innovative and inspiring, thought and action leaders, competent to solve problems faced by society, nation and world at large by striving towards very high standards in learning, teaching and training methodologies.

Our Motto: If it is not of quality, it is NOT RAIT!

Dr. Vijay D. Patil
President, RAES

Department Vision & Mission

Vision

To impart higher and quality education in computer science with value added engineering and technology programs to prepare technically sound, ethically strong engineers with social awareness. To extend the facilities, to meet the fast changing requirements and nurture the youths with international competencies and exemplary level of employability and research under highly competitive environments.

Mission

- To mobilize the resources and equip the institution with men and materials of excellence to provide knowledge and develop technologies in the thrust areas of computer science and Engineering.
- To provide the diverse platforms of sports, technical, co curricular and extracurricular activities for the overall development of student with ethical attitude.
- To prepare the students to sustain the impact of computer education for social needs encompassing industry, educational institutions and public service.
- To collaborate with IITs, reputed universities and industries for the technical and overall uplifment of students for continuing learning and entrepreneurship.

Index

List of Experiments

Sr. No.	Experiments Name
1	Build a simple network topology and configure it for static routing protocol using packet tracer using **Cisco Packet Tracer**.
2	Use basic networking commands (eg: ping, tracert, nslookup, netstat, ARP, RARP, ip, ifconfig, dig, route, etc) using Linux **Command Prompt**.
3	Study on Bluetooth protocol stack.
4	Implement the Hamming code using **C code/ Java.**
5	Implement Stop and wait protocol / sliding window (selective repeat / Go back N) using **Netsim**.
6	Setup a network and configure IP addressing, subnetting, Masking using **Cisco Packet Tracer**
7	Simulate congestion control (leaky bucket / token bucket) using **C code/Netsim**
8	Implement TCP or UDP Sockets using **Java**
9	Perform File Transfer and Access using FTP commands
10	Perform Remote login using Telnet server/ SSH server using Linux Command Prompt
11	1. Set up multiple IP addresses on a single LAN. 2. Using nestat and route commands of Linux, do the following: ▪ View current routing table ▪ Add and delete routes ▪ Change default gateway 3. Perform packet filtering by enabling IP forwarding using IPtables in Linux.
12	Use Wireshark to understand the operation of TCP/IP layers : ▪ Ethernet Layer : Frame header, Frame size etc. ▪ Data Link Layer : MAC address, ARP (IP and MAC address binding) ▪ Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo) ▪ Transport Layer: TCP Ports, TCP handshake segments etc. ▪ Application Layer: DHCP, FTP, HTTP header formats.

Course Outcome & Experiment Plan

Course Objectives:

1.	To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
2.	To explore the fundamental components of Application layer.
3.	To understand the issues and challenges of Transport layer.
4.	To gain core knowledge of network layer protocols and IP addressing.
5.	To acquire knowledge of Data Link layer services and protocols.
6.	To know and discover applications of various application layer protocols.

Lab Outcomes:

CO1	Demonstrate the concept of data communication with the help of networking commands and topology
CO2	Conceptualize data communication at application layer.
CO3	Demonstrate the transport layer with the help of socket programming Techniques and protocols.
CO4	Design the network using IP addressing and sub netting / super netting schemes using Network tools and simulators such as Wireshark ,NS2 etc.
CO5	Demonstrate the Data link layer with the help of error detection and correction.
CO6	Explore the protocols at physical layer.

Experiment Plan:

Module No.	Week No.	Experiments Name	Course Outcome
1.	W1	Build a simple network topology and configure it for static routing protocol using packet tracer using **Cisco Packet Tracer**.	CO1
2.	W2	Use basic networking commands (eg: ping, tracert, nslookup, netstat, ARP, RARP, ip, ifconfig, dig, route, etc) using Linux **Command Prompt**.	CO1
3.	W3	Study on Bluetooth protocol stack.	CO6
4.	W4	Implement the Hamming code using **Java.**	CO5
5.	W5	Implement Stop and wait protocol / sliding window (selective repeat / Go back N) using **Netsim**.	CO5
6.	W6	Setup a network and configure IP addressing, subnetting, Masking using **Cisco Packet Tracer.**	CO4
7.	W7	Simulate congestion control (leaky bucket / token bucket) using **C code/Netsim.**	CO4
8.	W8	Implement TCP or UDP Sockets using **Java.**	CO3
9.	W9	Perform File Transfer and Access using FTP commands	CO2
10.	W10	Perform Remote login using Telnet server/ SSH server using Linux Command Prompt	CO2
11.	W11	1. Set up multiple IP addresses on a single LAN. 2. Using nestat and route commands of Linux, do the following: ▪ View current routing table ▪ Add and delete routes ▪ Change default gateway 3. Perform packet filtering by enabling IP forwarding using IPtables in Linux.	CO4
12.	W12	Use Wireshark to understand the operation of TCP/IP layers : ▪ Ethernet Layer : Frame header, Frame size etc. ▪ Data Link Layer : MAC address, ARP (IP and MAC address binding) ▪ Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo) ▪ Transport Layer: TCP Ports, TCP handshake segments etc. ▪ Application Layer: DHCP, FTP, HTTP header formats.	CO3,CO4, CO5
13.	W13	Case study on different network discovery using discovery tools (eg. Mrtg/Prtg)	CO4, CO5

Mapping Course Outcomes (CO) - Program Outcomes (PO)

Subject Weight	Course Outcomes	Contribution to Program outcomes											
		P_a	P_b	P_c	P_d	P_e	P_f	P_g	P_h	P_i	P_j	P_k	P_l
PR 80%	**CO1.** Demonstrate the concept of data communication with the help of networking commands and topology.	3		2			1		1	1	1		1
	CO2. Conceptualize data communication at application layer	1		2	3				1	1	1		1
	CO3. Demonstrate the transport layer with the help of socket programming Techniques and protocols.	1	1	1	3				1	1	1		1
	CO4. Design the network using IP addressing and subnetting / supernetting schemes using network tools and simulators such as wireshark ,NS2 etc	1			2	3			1	1	1		1
	CO5. Demonstrate the Data link layer with the help of error detection and correction.	1	2			3			1	1	1		1
	CO6. Explore the protocols at physical layer.	2				1		3	1	1	1		1

Study and Evaluation Scheme

Course Code	Course Name	Teaching Scheme			Credits Assigned			
CEL402	Computer Networks	Theory Hrs.	Practical Hrs.	Tutorial	Theory	Practical	Tutorial	Total
		--	02	--	--	01	--	01

Course Code	Course Name	Examination Scheme		
CEL402	Computer Networks Lab	Term Work	Oral/Practical	Total
		25 (Experiments: 15-marks, Attendance Theory& Practical: 05-marks, Assignments: 05-marks)	25	50

Term Work:
The Term work Marks are based on the weekly experimental performance of the students, Oral performance and regularity in the lab.
Students are expected to be prepared for the lab ahead of time by referring the manual and perform the experiment under the guidance and discussion. Next week the experiment write-up to be corrected along with oral examination. Practical**:**

End Semester Examination:
End of the semester, there will be Practical examination along with oral evaluation based on the laboratory work and the corresponding theory course Syllabus.

Experiment No.: 1

Build a simple network topology and configure it for static routing protocol

- **Aim:** Build a simple network topology and configure it for static routing protocol using packet tracer.

- **Objectives:** To introduce concepts and fundamentals of network topologies and their configurations.

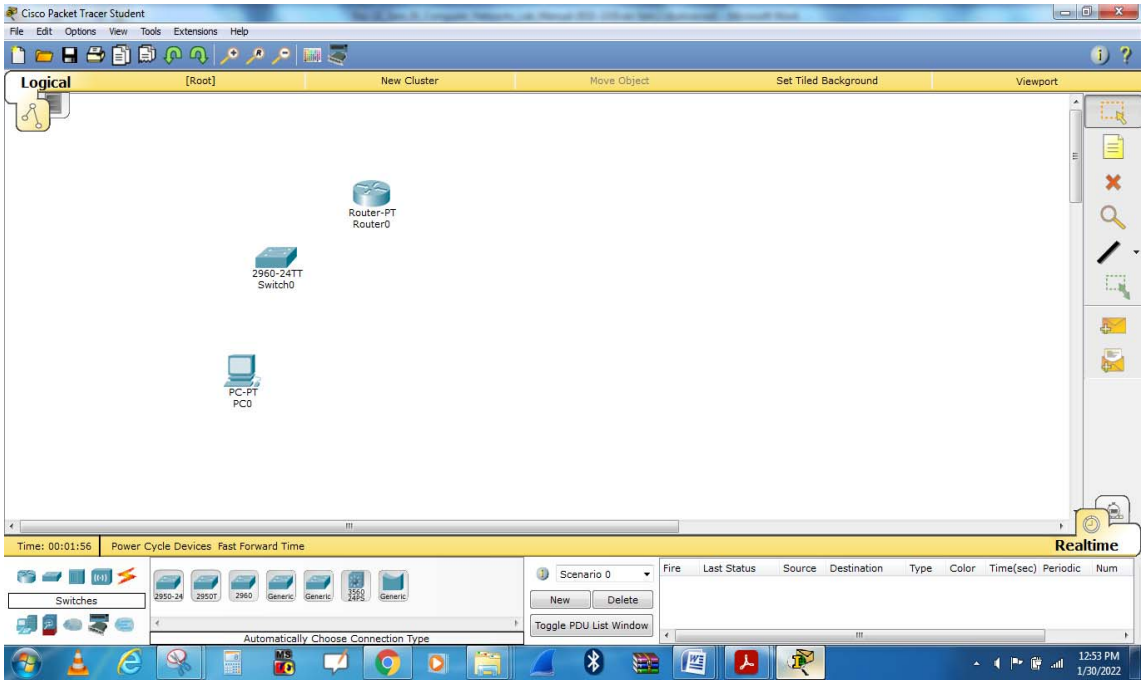
- **Outcomes:** The learner will be able to
  - Analyze the functioning of various networking devices.
  - Use the simulation tool Cisco Packet Tracer for building networks.
  - Recognize the need for topology.

- **Hardware/Software required:** Cisco Packet Tracer

- **Theory**

Setting the topology

1. Drag and drop the router/switch/computer from the bottom of the screen



2. Select **end devices** from the bottom left-hand corner and drag it to the sandbox screen.
3. Select **connections** from the same bottom left-hand corner. When you connect like-devices (Such as a router and computer) you use a crossover cable, so you should select **Automatically choose connection** from the second menu to the immediate right. Click on PC0 to Switch0 and *Router0*.

4. Repeat step number 3 to connect **PC1,PC2,PC3, and Switch0 to Router0.**

Configuring the router in packet tracer

5. Next we have to open the Ethernet ports to allow communication. Although they are physically connected, they are in a state that is known as being in **administrative shut down**. Now click on the **CLI** tab to access the configuration menu.

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n

Press RETURN to get started!

Router>**enable**
Router#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#**interface FastEthernet0/0**
Router(config-if)#**no shutdown**

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

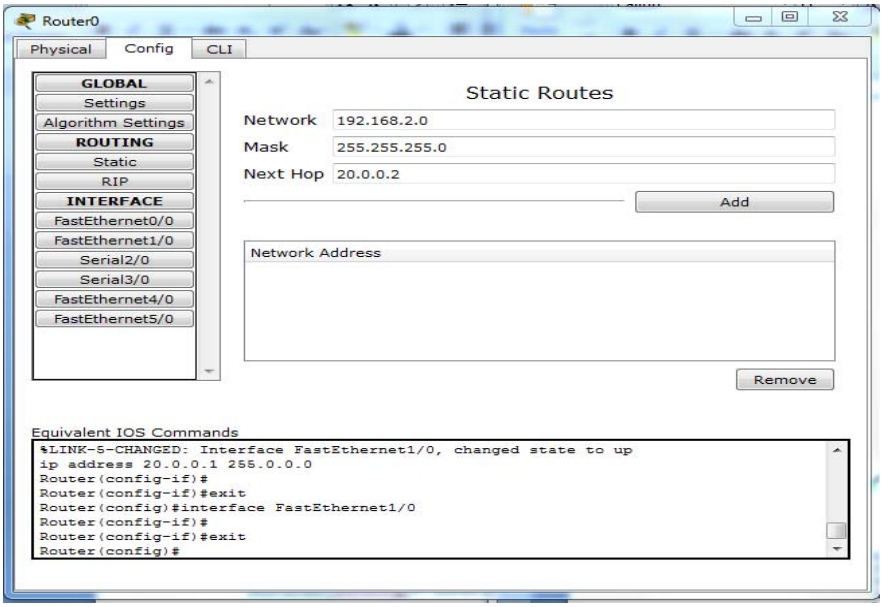
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#**ip address 192.168.1.1 255.255.255.0**
Router(config-if)#**exit**
Router(config)#**interface Serial2/0**
Router(config-if)#**no shutdown**

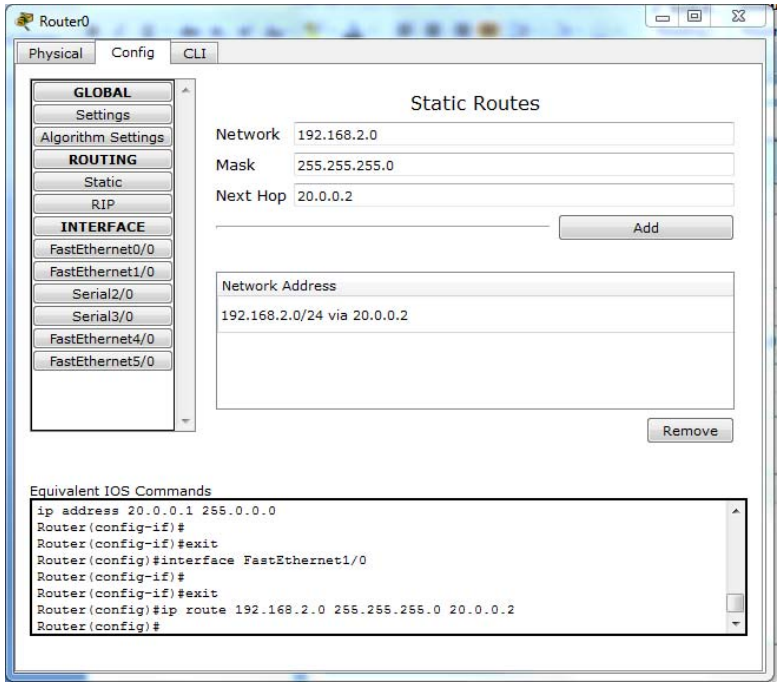
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

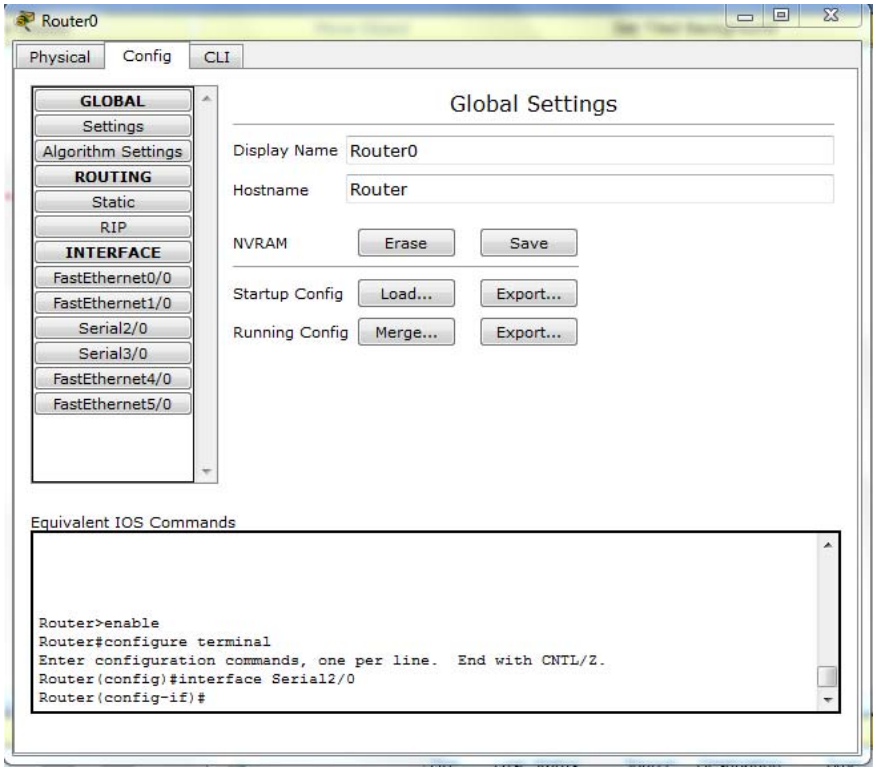
Router(config-if)#**ip address 20.0.0.1 255.0.0.0**
Router(config-if)#**exit**
Router(config)#**ip route 192.168.2.0 255.255.255.0 20.0.0.2**
Router(config)#**end**
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
Static Routes
Network 192.168.2.0
Mask 255.255.255.0
Next Hop 20.0.0.2
Add
Network Address
Remove
Equivalent IOS Commands
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up
ip address 20.0.0.1 255.0.0.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#

Add network

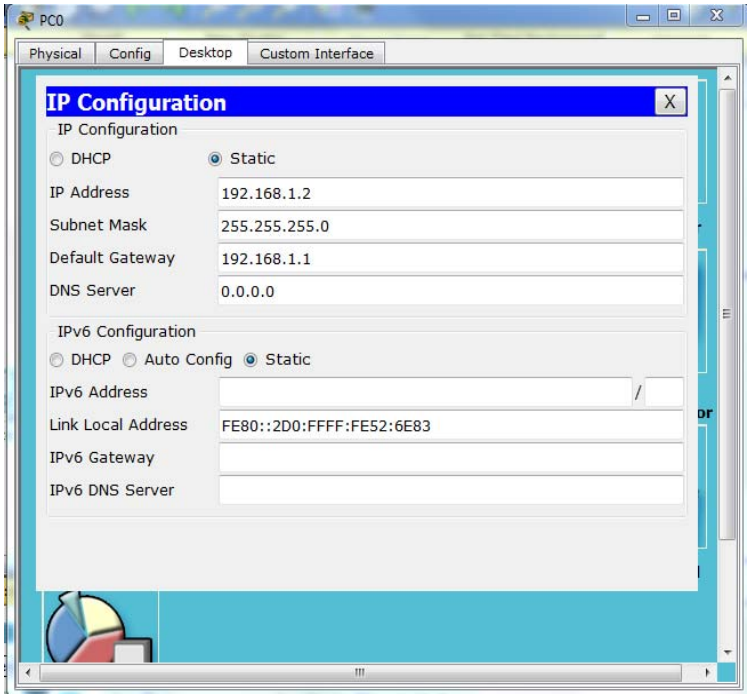

Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
Static Routes
Network 192.168.2.0
Mask 255.255.255.0
Next Hop 20.0.0.2
Add
Network Address
192.168.2.0/24 via 20.0.0.2
Remove
Equivalent IOS Commands
ip address 20.0.0.1 255.0.0.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#ip route 192.168.2.0 255.255.255.0 20.0.0.2
Router(config)#

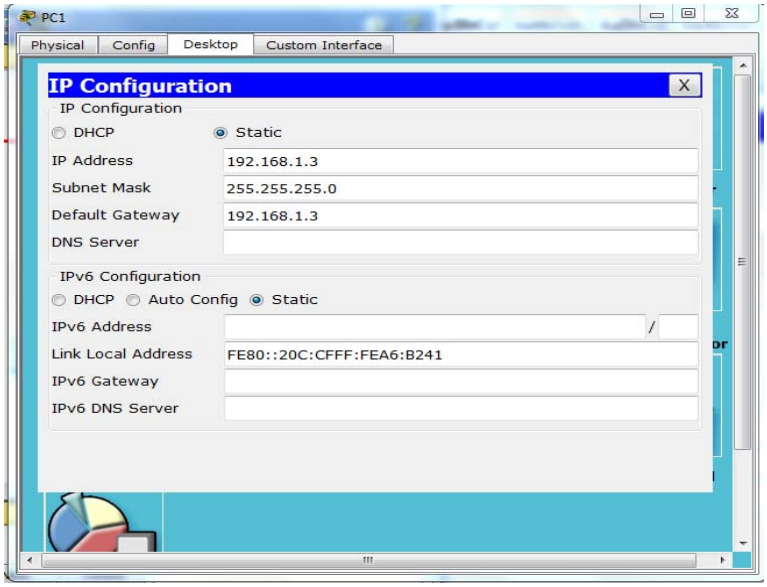
Go to **Config >settings** save configuration of static routing.

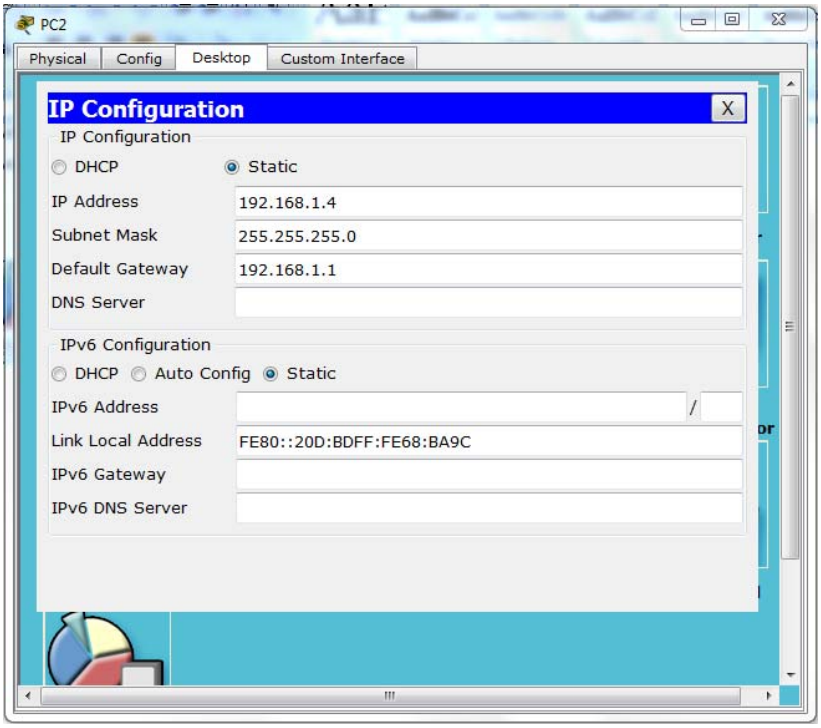


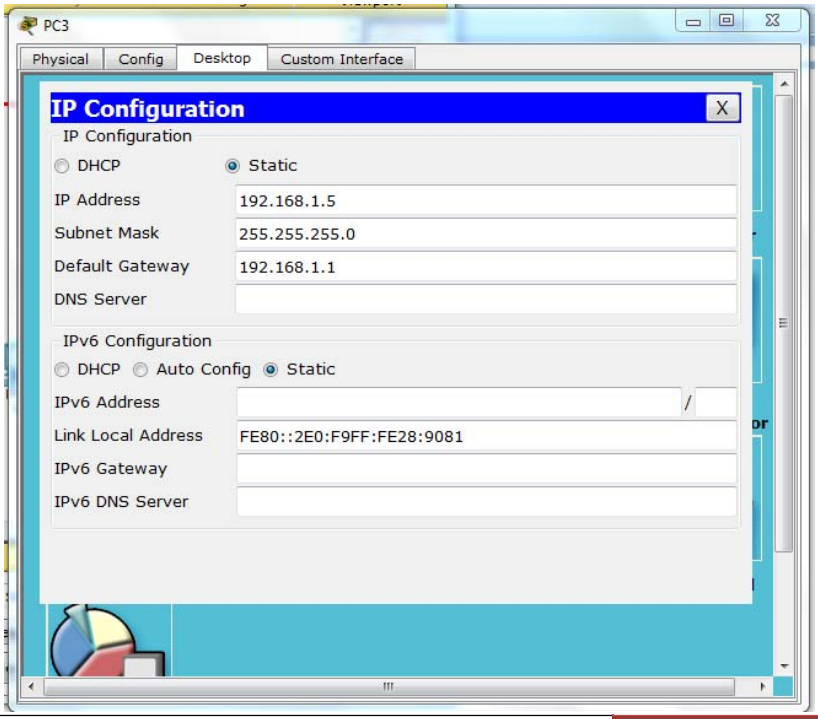
Computer configuration

1. Click on PC0 and go to Desktop
2. Choose IP configuration and configure IP address as 192.168.1.2 , click on subnet mask, subnet mask address will come automatically.
3. Default Gate way is Rourt0 IP address, configure this address as 192.168.1.1.
4. Close configuration.
5. Repeat step 1 to Step 4 to configure Ip addresses to PC1 to PC3.


PC0
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
192.168.1.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2D0:FFFF:FE52:6E83
IPv6 Gateway
IPv6 DNS Server


PC1
Physical
Config
Desktop
Custom Interface
IP Configuration
IP Configuration
DHCP
Static
IP Address
192.168.1.3
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.3
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::20C:CFFF:FEA6:B241
IPv6 Gateway
IPv6 DNS Server


PC2
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.1.4
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::20D:BDFF:FE68:BA9C
IPv6 Gateway
IPv6 DNS Server


PC3
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.1.5
Subnet Mask
255.255.255.0
Default Gateway
192.168.1.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2E0:F9FF:FE28:9081
IPv6 Gateway
IPv6 DNS Server

Router1 Configuration

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n

Press RETURN to get started!

Router>**enable**
Router#**configure terminal**
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#**interface FastEthernet0/0**
Router(config-if)#**no shutdown**

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

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

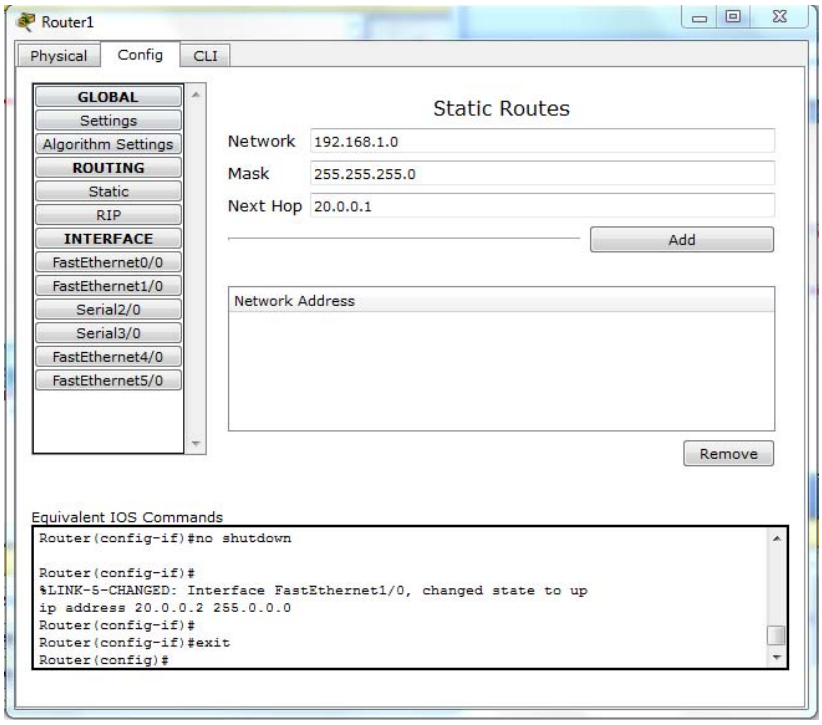
Router(config-if)#**ip address 192.168.2.1 255.255.255.0**
Router(config-if)#**exit**
Router(config)#**interface Serial2/0**
Router(config-if)#**no shutdown**

%LINK-5-CHANGED: Interface Serial2/0, changed state to up

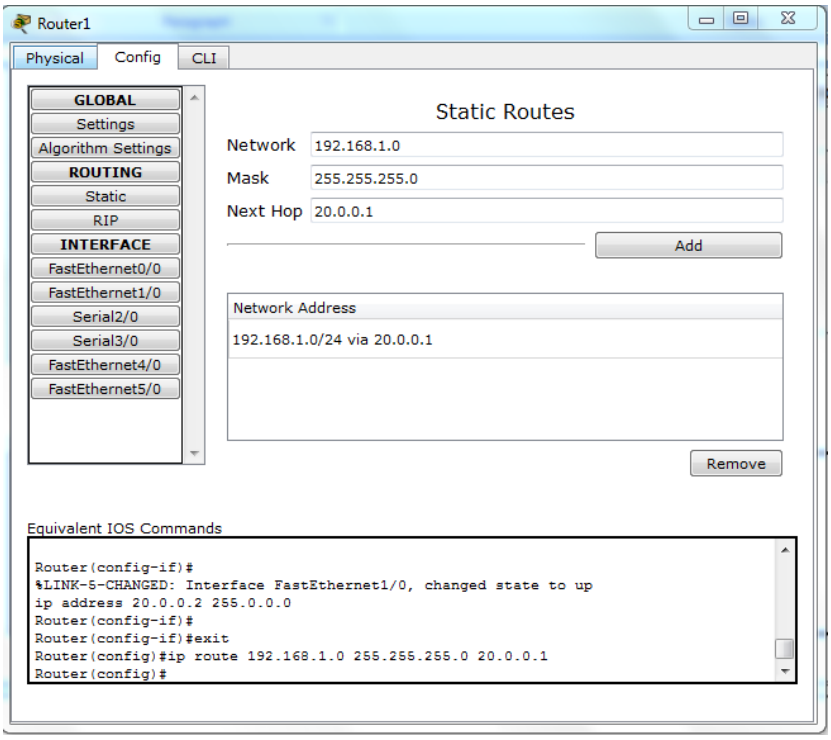
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#**ip address 20.0.0.2 255.0.0.0**
Router(config-if)#**exit**
Router(config)#**ip route 192.168.1.0 255.255.255.0 20.0.0.1**
Router(config)#**end**
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console

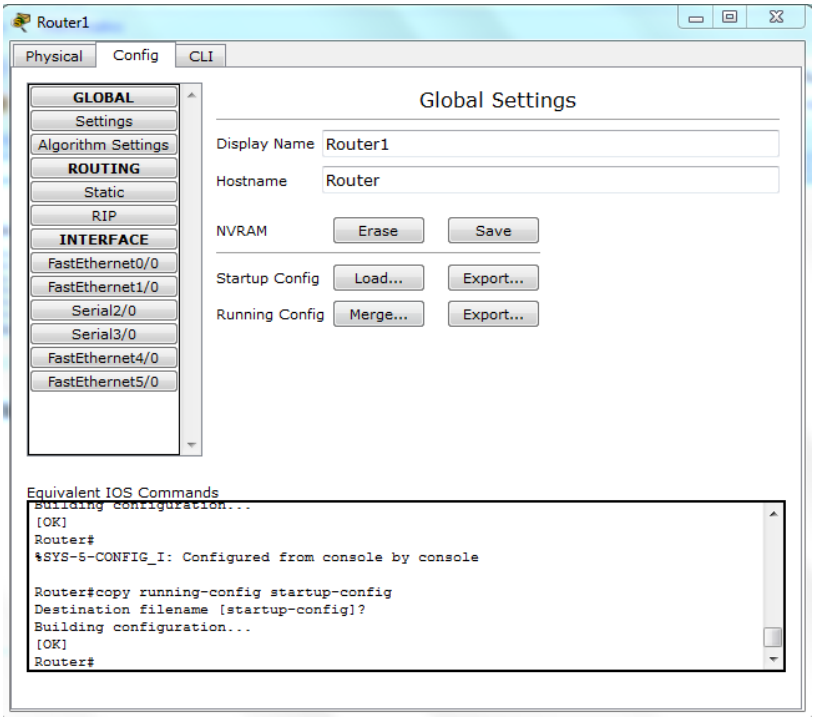
Static Routing



Add network

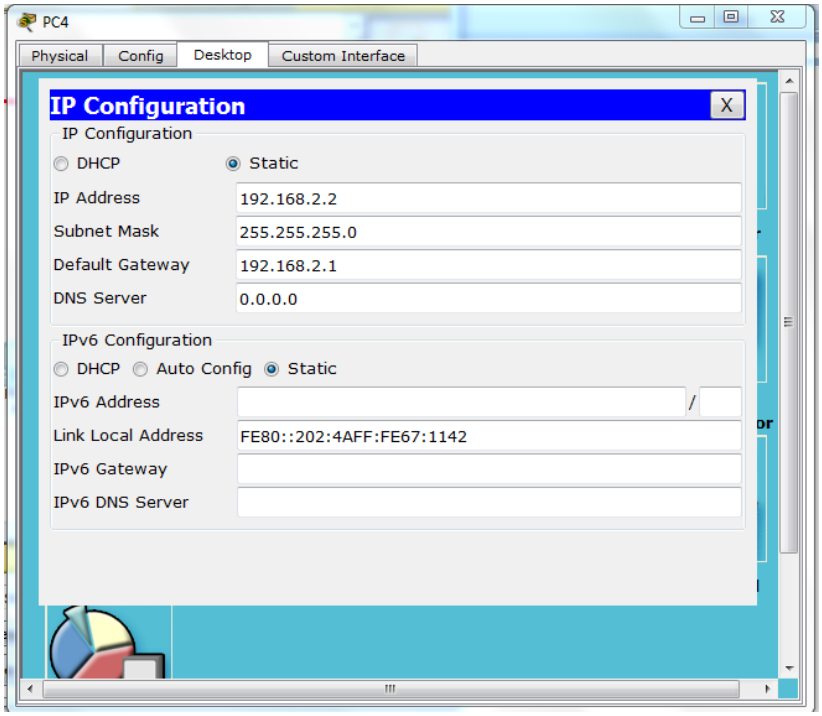


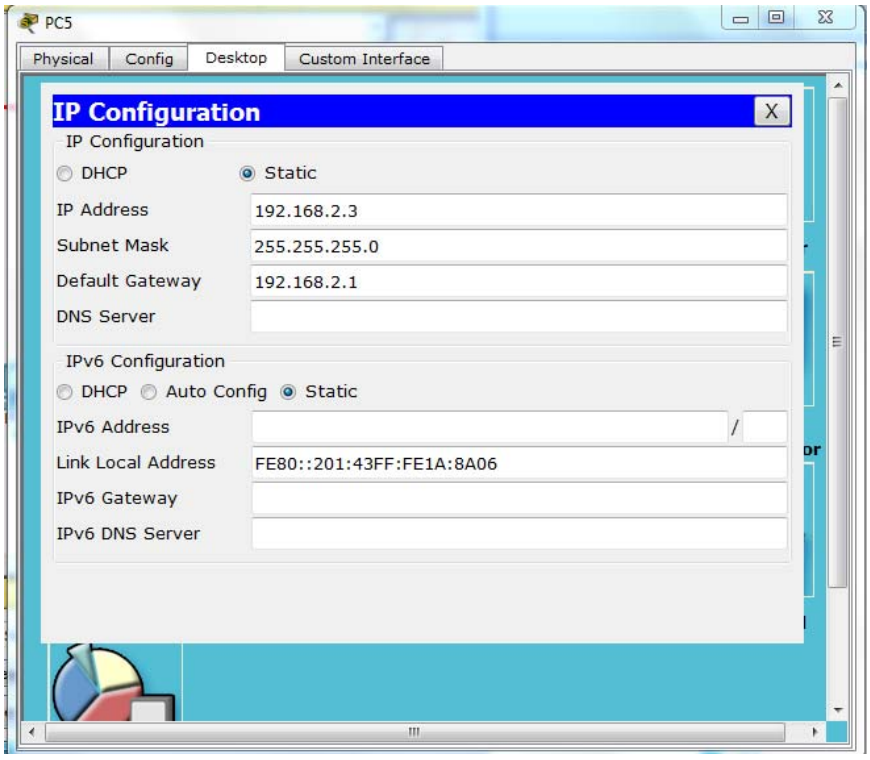
Go to **Config >settings** save configuration of static routing.

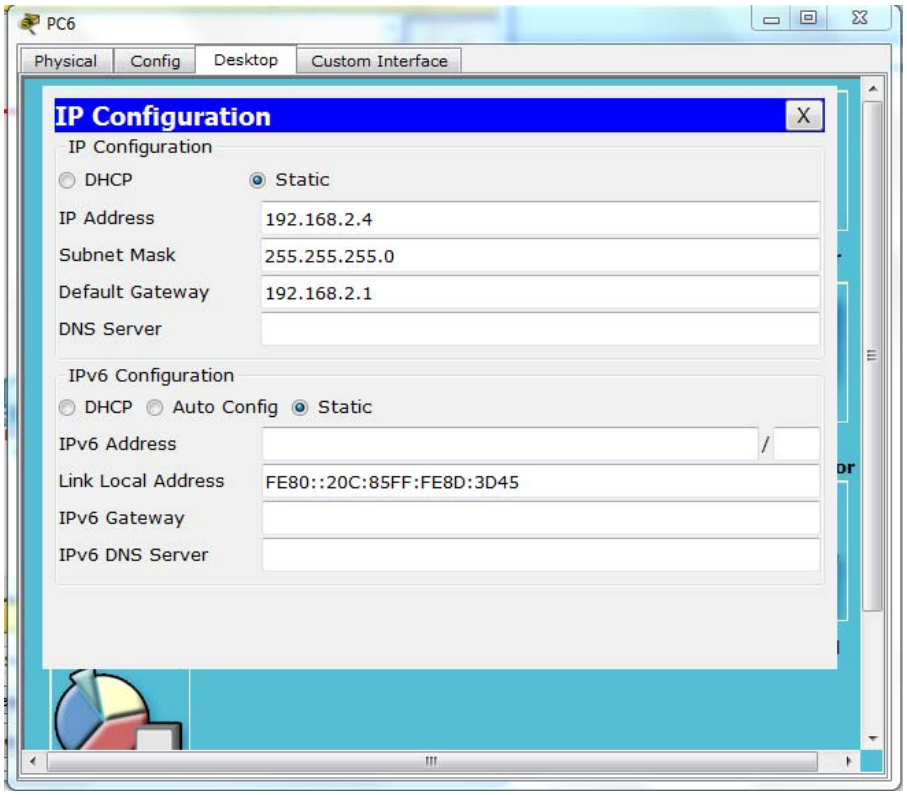


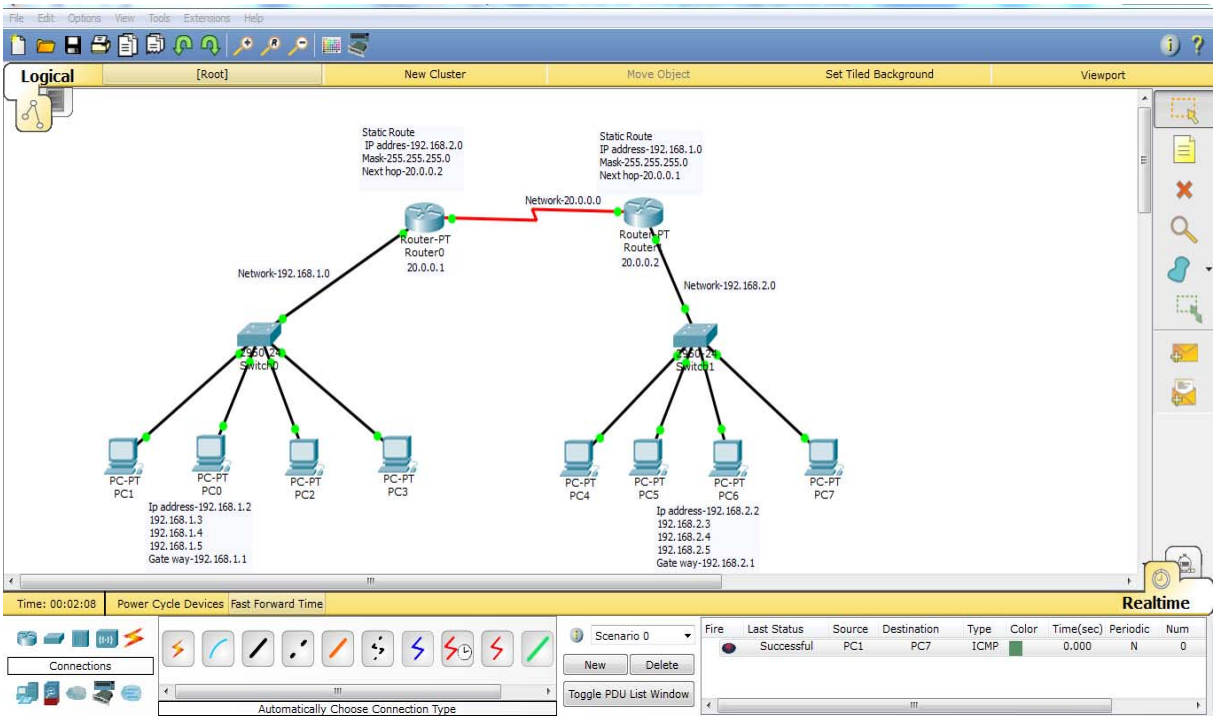
Computer configuration

1. Click on PC4 and go to Desktop
2. Choose IP configuration and configure IP address as 192.168.2.2 , click on subnet mask, subnet mask address will come automatically.
3. Default Gate Way is Rourt1 IP address, configure this address as 192.168.2.1.
4. Close configuration.
   Repeat step 1 to Step 4 to configure Ip addresses to PC5 to PC7.


PC4
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.2
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::202:4AFF:FE67:1142
IPv6 Gateway
IPv6 DNS Server


PC5
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.3
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::201:43FF:FE1A:8A06
IPv6 Gateway
IPv6 DNS Server


PC6
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.4
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::20C:85FF:FE8D:3D45
IPv6 Gateway
IPv6 DNS Server



Click on PC0 and then click PC7. On the lower right of the screen you will see a message box that says “Successful.”

- **Output Analysis:**
  (Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

- **Additional Learning:**
  (Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

- **Conclusion:** (Students should write conclusion on their own)

- **Viva Questions:**
  - What is the difference between star topology and bus topology?
  - State advantages and disadvantages of each of star, bus, ring, mesh.
  - Perform the experiment for ring topology and draw your own conclusion.
  - State the working difference between hub and switch.

- **References:**
  - A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e)
  - B.A. Forouzan, "Data Communications and Networking", TMH (5e)
  - James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

  - https://nptel.ac.in/content/storage2/courses/106105080/pdf/M5L1.pdf
  - https://nptel.ac.in/content/storage2/courses/117105076/pdf/5.1%20Lesson%2015.pdf
  - https://www.netacad.com/courses/packet-tracer
  - https://www.youtube.com/watch?v=A7kOCHdfYtw
  - https://www.youtube.com/watch?v=PmDmRbz7EHM

Experiment No.: 2

Networking Command

1. **Aim:** Use basic networking commands in Linux (ping, tracert, nslookup, netstat, ip, ifconfig, dig, route)

2. **Objectives:** To introduce concepts and fundamentals networking commands.

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of various networking commands.
   - Use the commands for building networks.
   - Recognize the need for networking commands in life-long learning.

4. **Hardware/Software required:** Command Prompt

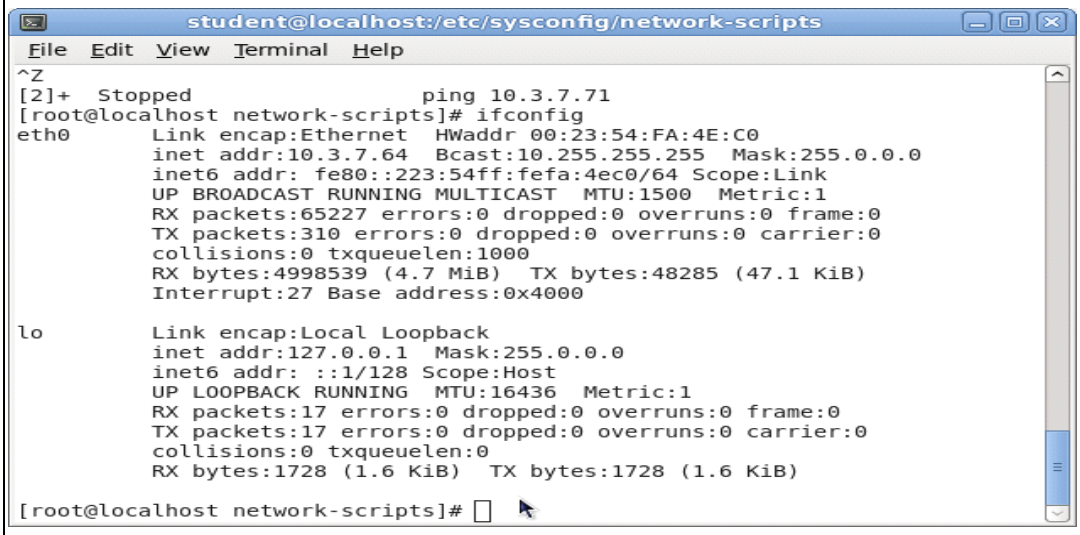
5. **Theory:**

1. ifconfig

ifconfig is used to configure the system's kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the system's active interfaces. If a single interface argument is given, it displays the status of the given interface only.

Eg: ifconfig

Running ifconfig with no options will display the configuration of all active interfaces.



1: ifconfig Command

2. ping

ping is a simple way to send network data to, and receive network data from, another computer on a network. It is frequently used to test, at the most basic level, whether another system is reachable over a network, and if so, how much time it takes for that data to be exchanged.

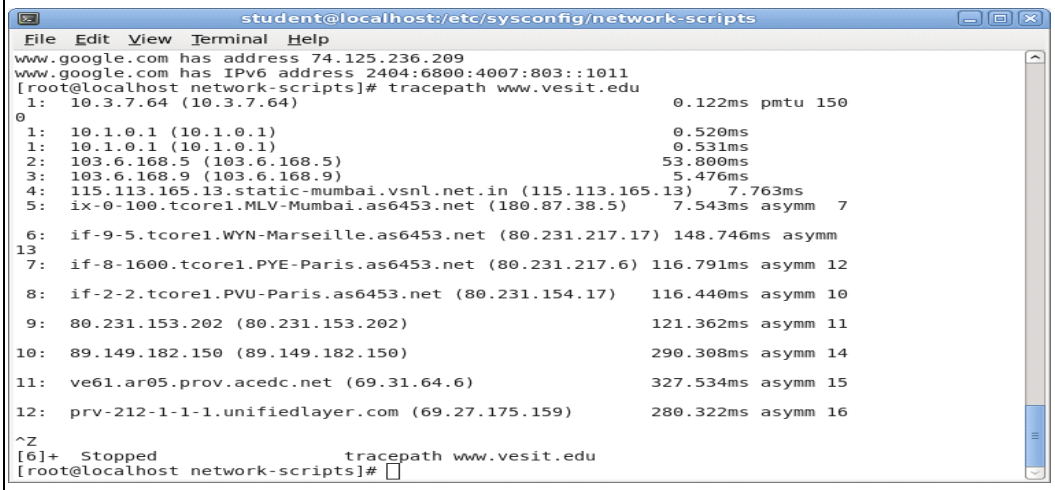
Eg: ping google.com

Ping the host google.com to see if it is alive.

2: ping Command

3. tracepath

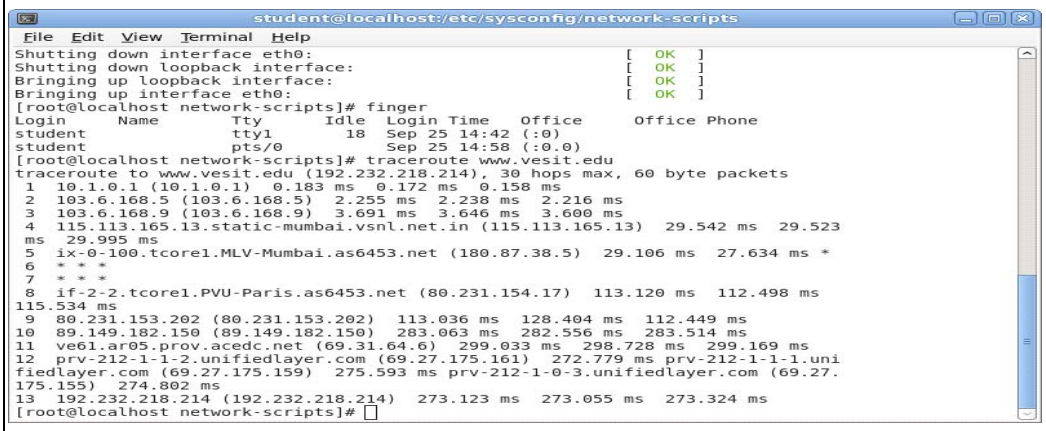
www.vesit.edu : It traces the complete path to a networking host discovering the MTU along the path.
It uses UDP port or some random port. It is similar to traceroute, only it does not require superuser privileges and has no fancy options.
Syntax: tracepath destination [port]



3: tracepath Command

4. traceroute

www.vesit.edu: traceroute prints the route that packets take to a network host. It is used to find network path from machine to server.
The server name above is destination name or IP address.
Syntax: traceroute <server name>

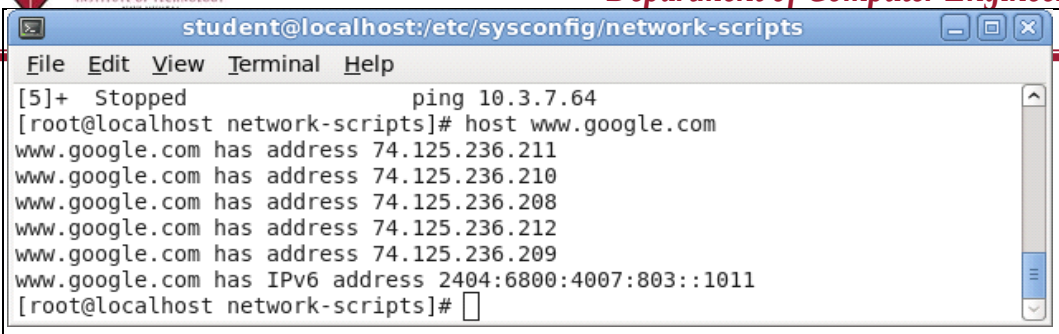


4: traceroute Command

5. host

www.google.com host is a simple utility for performing DNS lookups.

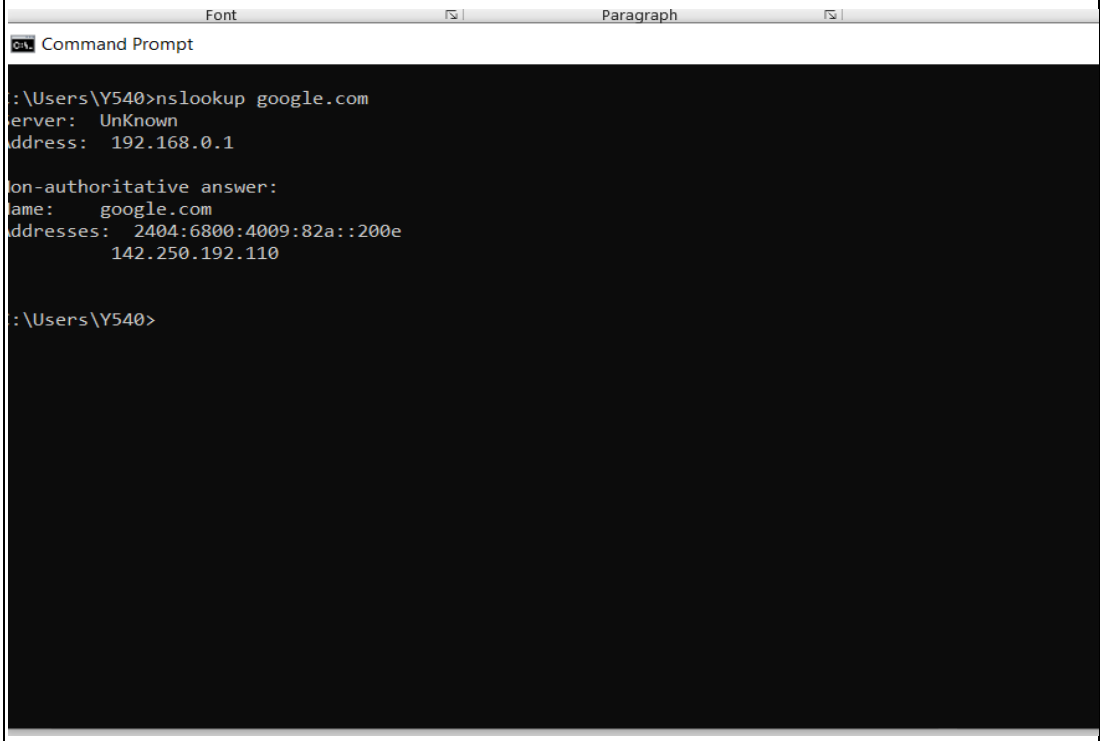
It is normally used to convert names to IP addresses and vice versa. When no arguments or options are given, host prints a short summary of its command line arguments and options.



5: host Command

6. NSLOOKUP

Nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems



7. netstat

The netstat command is used to print network connections, routing tables, interfacestatistics, masquerade connections, and multicast memberships. It is used for finding problems in the network and to determine the amount of traffic on the network as a performance measurement.

Eg: netstat –an

Shows information about all active connections to the server, including the source and destination IP addresses and ports, if you have proper permissions.

8. **ip** command in Linux is present in the net-tools which is used for performing several network administration tasks. IP stands for Internet Protocol. This command is used to show or manipulate routing, devices, and tunnels. It is similar to *ifconfig* command but it is much more powerful with more functions and facilities attached to it. *ifconfig* is one of the deprecated commands in the net-tools of Linux that has not been maintained for many years. ip command is used to perform several tasks like assigning an address to a network interface or configuring network interface parameters.
   It can perform several other tasks like configuring and modifying the default and static routing, setting up tunnel over IP, listing IP addresses and property information, modifying the status of the interface, assigning, deleting and setting up IP addresses and routes.

```
student@localhost:/etc/sysconfig/network-scripts
File  Edit  View  Terminal  Help
11  ve61.ar05.prov.acedc.net (69.31.64.6)  299.033 ms  298.728 ms  299.169 ms
```

```
student@lenovo804-ThinkCentre-M70e: ~
student@lenovo804-ThinkCentre-M70e:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
    link/ether 44:37:e6:4d:df:1b brd ff:ff:ff:ff:ff:ff
    inet 10.1.8.4/8 brd 10.255.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::4637:e6ff:fe4d:df1b/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:cf:c7:15:71 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
       valid_lft forever preferred_lft forever
student@lenovo804-ThinkCentre-M70e:~$
```

9. Dig

dig command stands for ***Domain Information Groper***. It is used for retrieving information about DNS name servers. It is basically used by network administrators. It is used for verifying and troubleshooting DNS problems and to perform DNS lookups. Dig command replaces older tools such as nslookup and the host.

```
$ dig example.com any
; <<>> DiG 9.6.1 <<>> example.com any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4016
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                   IN      ANY

;; ANSWER SECTION:
example.com.            172719  IN      NS      a.iana-servers.net.
example.com.            172719  IN      NS      b.iana-servers.net.
example.com.            172719  IN      A       208.77.188.166
example.com.            172719  IN      SOA     dns1.icann.org. hostmaster.icann.org. 2007051703 7200 3600 1209600 86400

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed Aug 12 11:40:43 2009
;; MSG SIZE  rcvd: 154
```

6. Output Analysis:

(Students should write output analysis based on the working of different networking networking commands used. Specify each scenario explicitly with output analysis)

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. Conclusion :

(Students should write conclusion on their own)

9. Viva Questions:

- What is the difference between traceroute and ifconfig
- How to find IP address of a website.
- State the working of nslookup.
- How to determine whether a we are able to connect to a system or not?

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e)
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE)
d. https://www.youtube.com/watch?v=OJ1gScTXqRc (NPTEL)

Experiment No.: 3

Bluetooth protocol stack

1. **Aim: Study on Bluetooth protocol stack**

2. **Objectives:** To introduce concepts and fundamentals of data communication and computer networks.

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of Bluetooth for communication.

4. **Hardware/Software required:** Device with Bluetooth feature

5. **Theory:**

Bluetooth wireless technology is a short-range radio technology, which is developed for Personal Area Network (PAN). Bluetooth is a standard developed by a group of electronics manufacturers that allows any sort of electronic equipment -- from computers and cell phones to keyboards and headphones -- to make its own connections, without wires, cables or any direct action from a user.
Bluetooth wireless technology makes it possible to transmit signals over short distances between telephones, computers and other devices and thereby simplify communication and synchronization between devices.
It is a global standard that:
• Eliminates wires and cables between both stationary and mobile devices
• Facilitates both data and voice communication
• Offers the possibility of ad hoc networks and delivers the ultimate synchronicity between all your personal devices
Bluetooth is intended to be a standard that works at two levels:
• It provides agreement at the physical level -- Bluetooth is a radio-frequency standard.
• It also provides agreement at the next level up, where products have to agree on when bits are sent, how many will be sent at a time and how the parties in a conversation can be sure that the message received is the same as the message sent.
Bluetooth Architecture
There are two types of topology for Bluetooth – Piconet, Scatternet.

1. Piconet :- The Piconet is a small ad hoc network of devices (normally 8 stations).

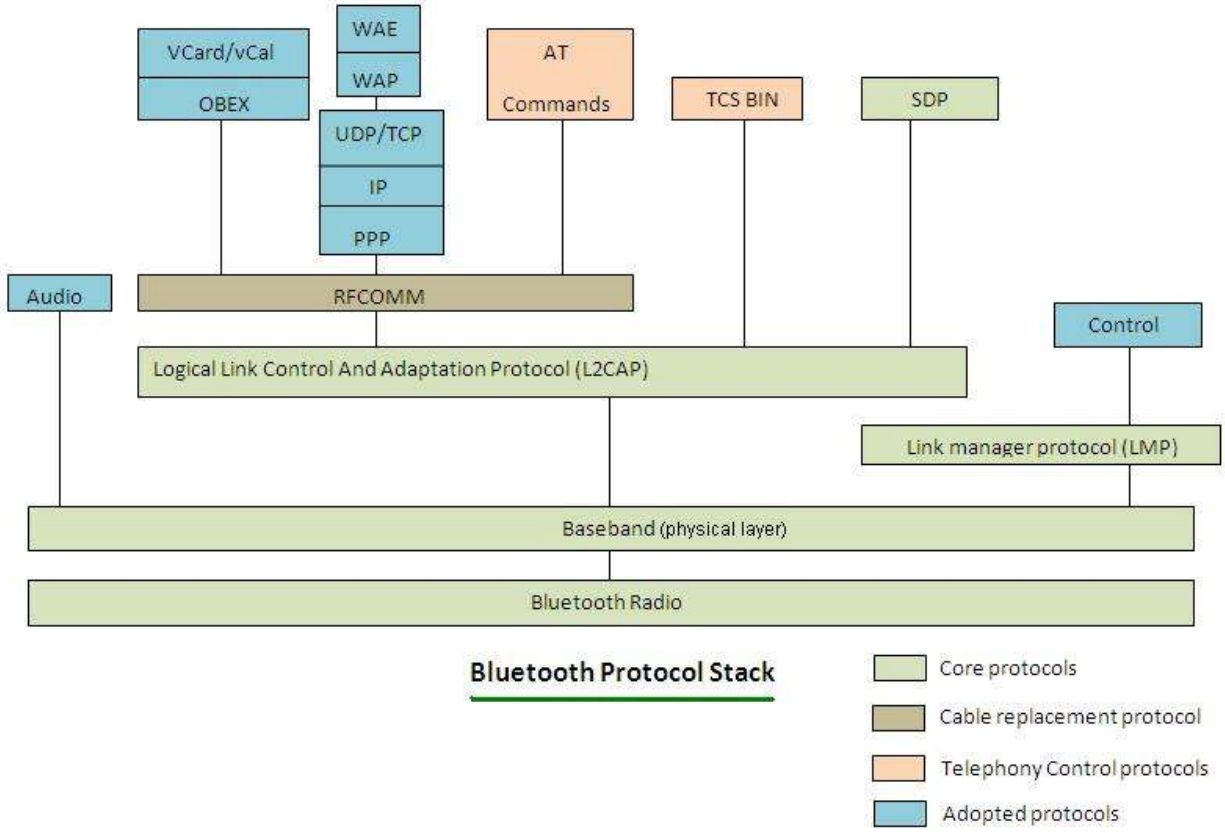
It has the following features:

- One is called Master and the others are called Slaves
- All slave stations synchronizes their clocks with the master
- Possible communication - One-to-one or one-to-many

2. Scatternet :- It is formed by combining several Piconets.

Key features of the scatternet topology are mentioned below:
• A Scatternet is the linking of multiple co-located piconets through the sharing of common master or slave devices.
• A device can be both a master and a slave.

Bluetooth Protocol Stack and Architecture



The different layers of Bluetooth Architecture are –

ISO	
	Application
7	RFCOMM / SDP
6	L2CAP
5	Host controller interface
4	Link manager
3	Link controller
2	Baseband (ACL, SCO or eSCO Link)
1	Radio layer

• Radio Layer: The Radio layer defines the requirements for a Bluetooth transceiver operating in the 2.4 GHz ISM band.

• Baseband: The Baseband layer describes the specification of the Bluetooth Link Controller (LC),

which carries out the baseband protocols and other low-level link routines. It specifies Piconet/Channel definition, "Low-level" packet definition, Channel sharing.

• LMP: The Link Manager Protocol (LMP) is used by the Link Managers (on either side) for link set-up and control.

• HCI: The Host Controller Interface (HCI) provides a command interface to the Baseband Link Controller and Link Manager, and access to hardware status and control registers.

• L2CAP: Logical Link Control and Adaptation Protocol (L2CAP) supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information.

• RFCOMM: The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10.

• SDP: The Service Discovery Protocol (SDP) provides a means for applications to discover, which services are provided by or available through a Bluetooth device.

• Application : These are software modules which connect the host application program to the Bluetooth communications system. As such they reside and execute on the same processing resource as the host system application.

6. Conclusion :

Thus we have studied the different layers in Bluetooth protocol stack.

7. Output Analysis:

(Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

8. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

9. Conclusion :

(Students should write conclusion on their own)

10. Viva Questions:

- State advantages and disadvantages of each of bluetooth
- Name different layers of Bluetooth protocol stack.
- What is piconet and scatternet.

11. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e).
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE)

Experiment No.: 4
Implement the Hamming code

1. **Aim:** Implement the Hamming code using C code.
2. **Objectives:** To introduce concepts of Hamming code.
3. **Outcomes:** The learner will be able to
   - Analyze the concept of error detection and.correction.
   - Recognize the need for Hamming code.
4. **Hardware/Software required:** Turbo C, C++ Integrated Development Environment & compile.
5. **Theory:**

Hamming code is a popular error detection and error correction method in data communication. Hamming code can only detect 2 bit error and correct a single bit error which means it is unable to correct burst errors if may occur while transmission of data.

Hamming code uses redundant bits (extra bits) which are calculated according to the below formula:-

$$2^r \geq m+r+1$$

Where **r** is the number of redundant bits required and **m** is the number of data bits.

R is calculated by putting **r = 1, 2, 3 …** until the above equation becomes true.

R1 bit is appended at position **20**

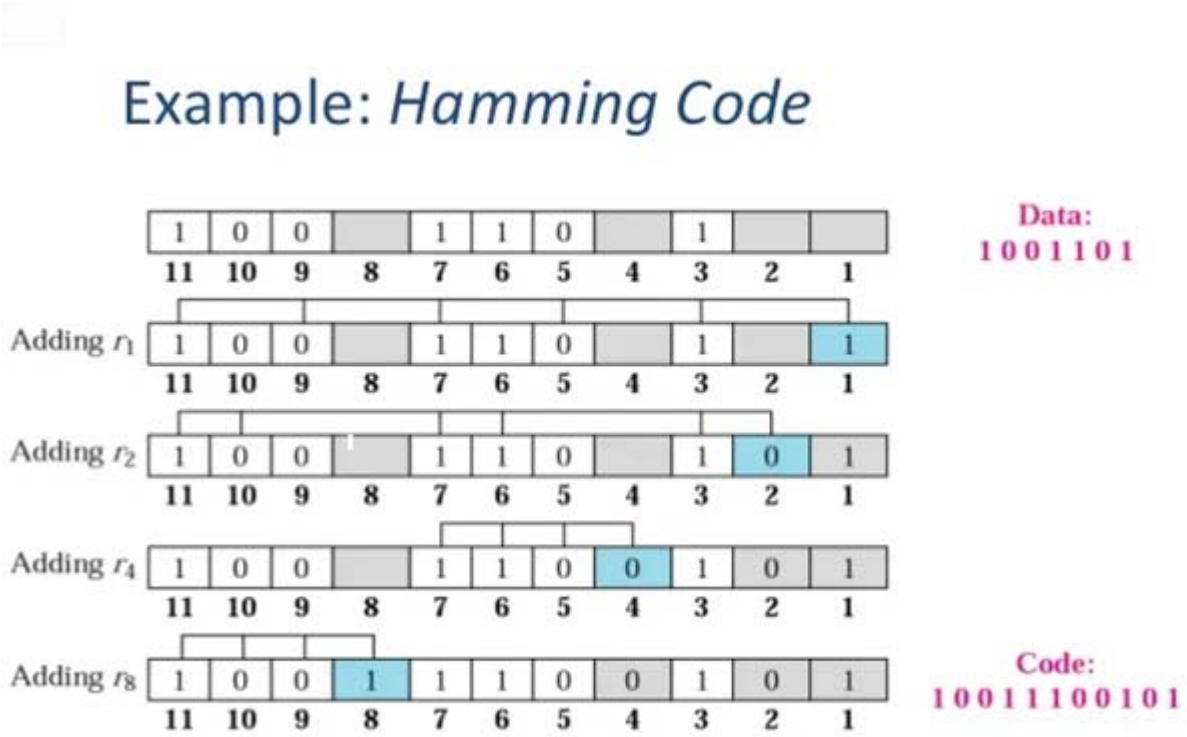
R2 bit is appended at position **21**

R3 bit is appended at position **22** and so on.

These redundant bits are then added to the original data for the calculation of error at receiver's end.

At receiver's end with the help of even parity (generally) the erroneous bit position is identified and since data is in binary we take complement of the erroneous bit position to correct received data.

Respective index parity is calculated for **r1, r2, r3, r4** and so on.



Program for Hamming Code in C++

```
#include<iostream>

using namespace std;

int main() {
   int data[10];
   int dataatrec[10],c,c1,c2,c3,i;

   cout<<"Enter 4 bits of data one by one\n";
   cin>>data[0];
   cin>>data[1];
   cin>>data[2];
   cin>>data[4];
    //Calculation of even parity
   data[6]=data[0]^data[2]^data[4];
data[5]=data[0]^data[1]^data[4];
data[3]=data[0]^data[1]^data[2];

cout<<"\nEncoded data is\n";
```

```
for(i=0;i<7;i++)
        cout<<data[i];

cout<<"\n\nEnter received data bits one by one\n";
    for(i=0;i<7;i++)
        cin>>dataatrec[i];

    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
c=c3*4+c2*2+c1 ;

    if(c==0) {
cout<<"\nNo error while transmission of data\n";
    }
else {
cout<<"\nError on position "<<c;
cout<<"\nData sent : ";
for(i=0;i<7;i++)
         cout<<data[i];

cout<<"\nData received : ";
        for(i=0;i<7;i++)
         cout<<dataatrec[i];

cout<<"\nCorrect message is\n";

//if errorneous bit is 0 we complement it else vice versa
if(dataatrec[7-c]==0)
dataatrec[7-c]=1;
        else
dataatrec[7-c]=0;
for (i=0;i<7;i++) {
cout<<dataatrec[i];
}
}
return 0;
}
```

Output

Enter 4 bits of data one by one
1
0
1
0
Encoded data is
1010010
Enter received data bits one by one
1
0

1 0
0
1
0
No error while transmission of data

6. Conclusion :

(Students should write conclusion on their own)

7. Viva Questions:

- What is Hamming code? How does it works?
- Explain Hamming code generation and correction with example?
- State advantages and disadvantages of Hamming codes.

8. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e).
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 5

Implement Sliding window Protocol

1. **Aim:** Implement Sliding window protocol.

2. **Objectives:** To introduce concepts of implement Sliding window protocol.

3. **Outcomes:** The learner will be able to
   - To study sliding window mechanism.
   - To understand how to implement sliding window mechanism.
   - To discuss the application of sliding window protocol to control traffic flow in networks.

4. **Hardware/Software required:** Turbo C, C++ Integrated Development Environment & compile.

5. **Theory:**

Sliding window is a flow control technique which belongs to the Data Link layer of the OSI model. With a stop-and-wait protocol, the sender waits for an acknowledgment after transmitting every frame. As a result, there is at most a single outstanding frame on the channel at any given time, which may be far less than the channel's capacity. For maximum throughput, the amount of data in transit at any given time should be (channel bandwidth) * (channel delay).

The key feature of the sliding-window protocol is that it permits pipelined communication to better utilize the channel capacity. The sender can send a maximum W frames without acknowledgement. W is called the window size of the sliding window.

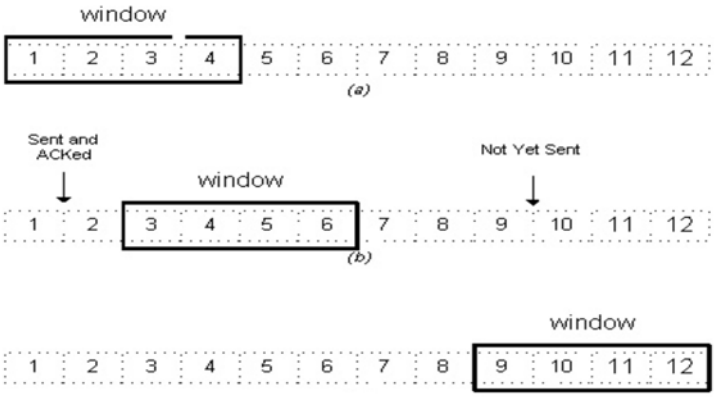
The sliding window maps to the frames in sender's buffer that are to be sent, or have been sent and now are just waiting for acknowledgement.

In computer networks sliding window protocol is a method to transmit data on a network. Sliding window is a flow control technique is applied on the Data Link Layer of OSI model. It solves the problem of missing frames during data transmission between two upper layers, so that they can send and receive frames in order. At data link layer data is in the form of frames.

In networking, window simply means a buffer which has data frames that needs to be transmitted.

In sliding window mechanism, receiver sends out acknowledgement to sender to notify of receiving or missing of frames. Both sender and receiver agree on some window size. If window size=w then after sending w frames sender waits for the acknowledgement (ack) of the first frame.

As soon as sender receives the acknowledgement of a frame it is replaced by the next frames to be transmitted by the sender. If receiver sends a collective or cumulative acknowledgement to 79 sender then it understands that more than one frames are properly received, for example if ack of frame 3 is received it understands that frame 1 and frame 2 are received properly.



In sliding window protocol the receiver has to have some memory to compensate any loss in transmission or if the frames are received unordered.

Efficiency of Sliding Window Protocol

$\eta = (W*t_x)/(t_x+2t_p)$

W = Window Size

t_x = Transmission time

t_p = Propagation delay

9. Sample program

```
#include<stdio.h>
#include<conio.h>
int main()
{int w,i,f,frames[50];
printf("Enter the window size: ");
scanf("%d",&w);
printf("\nEnter the number of frames to transmit: ");
scanf("%d",&f);
printf("\nEnter %d frames:",f);
for(i=1;i<=f;i++)
{
scanf("%d",&frames[i]);
}
printf("With sliding window protocol the frames will be sentin the following
manner(assuming no corruption of frames)\n\n");
printf("After sending%d frames at each stage sender waits for acknowledgement sent by
the receiver\n\n",w);
for(i=1;i<=f;i++)
{if(i%w==0)
```

```
{printf("%d\n",frames[i]);
printf("Acknowledgment of above frames sent is received by sender\n\n");
}
else
printf("%d ",frames[i]);
}
if(f%w!=0)
printf("\n Acknowledgment of above frames sent is received by sender\n");
getch();
}
```

Output:

Enter window size: 3
Enter number of frames to transmit: 5
Enter 5 frames: 12 5 89 4 6
With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)
After sending 3 frames at each stage sender waits for acknowledgement sent by the receiver
12 5 89
Acknowledgement of above frames sent is received by sender
4 6
Acknowledgement of above frames sent is received by sender

7. Conclusions:

8. Viva Questions:

- Why is flow control needed?
- Explain the concept of sliding window flow control.
- How the inefficiency of stop and wait protocol is overcome in sliding window protocol?

9. References:

a. A.S. Tanenbaum, “Computer Networks”, Pearson Education, (4e)
b. B.A. Forouzan, “Data Communications and Networking”, TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 5

Implement Stop and Wait Protocol and Sliding Window Protocol

1. **Aim:** Implement Stop and Wait Protocol and Sliding Window Protocol using Network Simulator (NS2)

2. **Objectives:** To introduce NS2 simulator

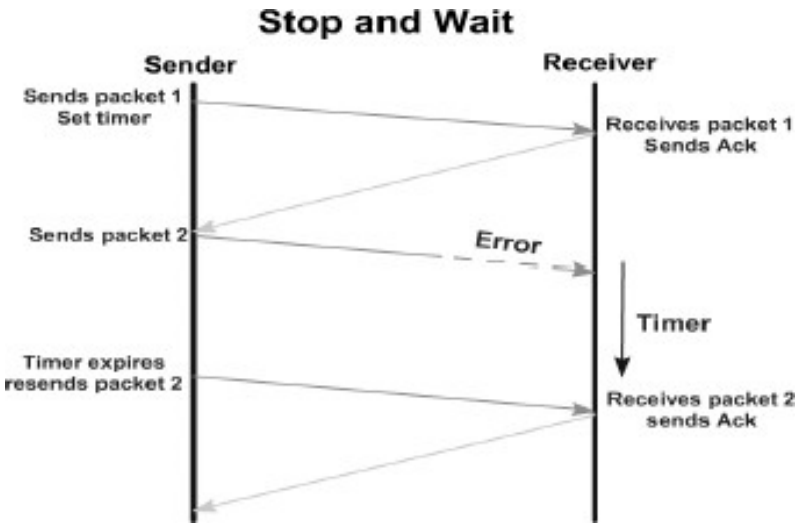
3. **Outcomes:** The learner will be able to
   - Analyze the functioning of Stop and Wait Protocol
   - Analyze the functioning of Sliding Window Protocol
   - Simulate and Study Stop and Wait Protocol / Sliding Window Protocol

4. **Hardware/Software required:** NS2 Simulator

5. **Theory**

The Stop and Wait Protocol is a reliable transmission flow control protocol. This protocol works only in Connection Oriented (Point to Point) Transmission. The Source node has window size of ONE. After transmission of a frame the transmitting (Source) node waits for an Acknowledgement from the destination node. If the transmitted frame reaches the destination without error, the destination transmits a positive acknowledgement. If the transmitted frame reaches the Destination with error, the receiver destination does not transmit an acknowledgement. If the transmitter receives a positive acknowledgement it transmits the next frame if any. Else if its acknowledgement receive timer expires,it retransmits the same frame.

1. Start with the window size of 1 from the transmitting (Source) node
2. After transmission of a frame the transmitting (Source) node waits for a reply (Acknowledgement) from the receiving (Destination) node.
3. If the transmitted frame reaches the receiver (Destination) without error, the receiver (Destination) transmits a Positive Acknowledgement.
4. If the transmitted frame reaches the receiver (Destination) with error, the receiver (Destination)do not transmit acknowledgement.
5. If the transmitter receives a positive acknowledgement it transmits the next frame if any. Else ifthe transmission timer expires, it retransmits the same frame again.
6. If the transmitted acknowledgment reaches the Transmitter (Destination) without error, theTransmitter (Destination) transmits the next frame if any.
7. If the transmitted frame reaches the Transmitter (Destination) with error, the Transmitter (Destination) transmits the same frame.
8. This concept of the Transmitting (Source) node waiting after transmission for a reply from thereceiver is known as STOP and WAIT.



Algorithm:

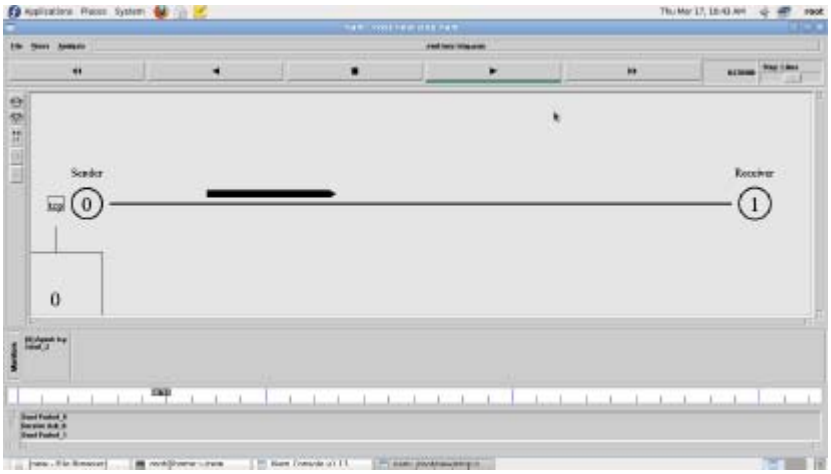
1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam ontrace file.
4. Create two nodes that forms a network numbered 0 and 1
5. Create duplex links between the nodes to form a STAR Topology
6. Setup TCP Connection between n(1) and n(3)
7. Apply CBR Traffic over TCP
8. Schedule events and run the program.

Program:

```
# stop and wait protocol in normal situation
# features : labeling, annotation, nam-graph, and window size
monitoringset ns [new Simulator]
set n0 [$ns node]set
n1 [$ns node]
$ns at 0.0 "$n0 label Sender"
$ns at 0.0 "$n1 label Receiver"
set nf [open stop.nam w]
$ns namtrace-all $nfset
f [open stop.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n1 0.2Mb 200ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns queue-limit $n0 $n1 10
Agent/TCP set nam_tracevar_ true
set tcp [new Agent/TCP]
$tcp set window_ 1
$tcp set maxcwnd_ 1
```

```
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns add-agent-trace $tcp tcp
$ns monitor-agent-trace $tcp
$tcp tracevar cwnd_
$ns at 0.1 "$ftp start"
$ns at 3.0 "$ns detach-agent $n0 $tcp ;
$ns detach-agent $n1 $sink"
$ns at 3.5 "finish"
$ns at 0.0 "$ns trace-annotate \"Stop and Wait with normal operation\""
$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.1\""
$ns at 0.11 "$ns trace-annotate \"Send Packet_0\""
$ns at 0.35 "$ns trace-annotate \"Receive Ack_0\""
$ns at 0.56 "$ns trace-annotate \"Send Packet_1\""
$ns at 0.79 "$ns trace-annotate \"Receive Ack_1\""
$ns at 0.99 "$ns trace-annotate \"Send Packet_2\""
$ns at 1.23 "$ns trace-annotate \"Receive Ack_2  \""
$ns at 1.43 "$ns trace-annotate \"Send Packet_3\""
$ns at 1.67 "$ns trace-annotate \"Receive Ack_3\""
$ns at 1.88 "$ns trace-annotate \"Send Packet_4\""
$ns at 2.11 "$ns trace-annotate \"Receive Ack_4\""
$ns at 2.32 "$ns trace-annotate \"Send Packet_5\""
$ns at 2.55 "$ns trace-annotate \"Receive Ack_5     \""
$ns at 2.75 "$ns trace-annotate \"Send Packet_6\""
$ns at 2.99 "$ns trace-annotate \"Receive Ack_6\""
$ns at 3.1 "$ns trace-annotate \"FTP stops\""
proc finish {} {
global ns nf
$ns flush-trace close $nf
puts "running nam..." exec nam stop.nam & exit 0
}
$ns run
```

Output:



A sliding Window Protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the Data Link Layer(OSI model) as well as in the Transmission Control Protocol (TCP).

Conceptually, each portion of the transmission (packets in most data link layers, but bytes in TCP) is assigned a unique consecutive sequence number, and the receiver uses the numbers to place received packets in the correct order, discarding duplicate packets and identifying missing ones. The problemwith this is that there is no limit on the size of the sequence number that can be required.

By placing limits on the number of packets that can be transmitted or received at any given time, a sliding window protocol allows an unlimited number of packets to be communicated using fixed-size sequence numbers. The term "window" on the transmitter side represents the logical boundary of the total number of packets yet to be acknowledged by the receiver. The receiver informs the transmitter in each acknowledgment packet the current maximum receiver buffer size (window boundary). The TCP header uses a 16 bit field to report the receive window size to the sender. Therefore, the largest window that can be used is 216 = 64 kilobytes. In slow-start mode, the transmitter starts with low packet count and increases the number of packets in each transmission after receiving acknowledgment packets from receiver. For every ack packet received, the window slides by one packet (logically) to transmit one new packet. When the window threshold is reached, the transmitter sends one packet for one ack packet received. If the window limit is 10 packets then in slow start mode the transmitter may start transmitting one packet followed by two packets (before transmitting two packets, one packet ack has to be received), followed by three packets and so on until 10 packets. But after reaching 10 packets, further transmissions are restricted to one packet transmitted for one ack packet received. In a simulation this appears as if the window is moving by one packet distance for every ack packet received. On the receiver side also the window moves one packet for every packet received. The sliding window method ensures that traffic congestion on the network is avoided. The application layer will still be offering data for transmission to TCP without worrying about the network traffic congestion issues as the TCP on sender and receiver side implement sliding windows of packet buffer. The window size may vary dynamically depending on network traffic.

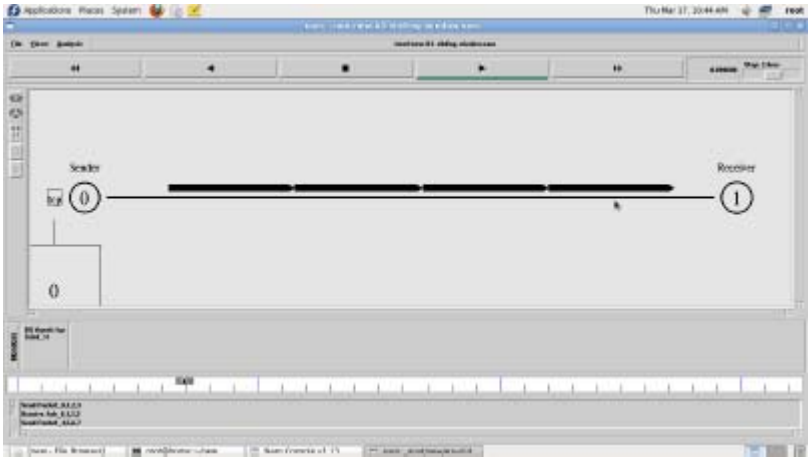
For the highest possible throughput, it is important that the transmitter is not forced to stop sending by the sliding window protocol earlier than one round-trip delay time (RTT). The limit on the amount of data that it can send before stopping to wait for an acknowledgment should be larger than the bandwidth-delay product of the communications link. If it is not, the protocol will limit the effective bandwidth of the link.

Program:

```
# sliding window mechanism with some features
# such as labeling, annotation, nam-graph, and window size
monitoringset ns [new Simulator]
set n0 [$ns
node]set n1
[$ns node]
$ns at 0.0 "$n0 label Sender"
$ns at 0.0 "$n1 label
Receiver"set nf [open
sliding.nam w]
$ns namtrace-all
$nf set f [open
sliding.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n1 0.2Mb 200ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns queue-limit $n0 $n1 10
Agent/TCP set nam_tracevar_
trueset tcp [new Agent/TCP]
$tcp set windowInit_ 4
$tcp set maxcwnd_ 4
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns add-agent-trace $tcp tcp
$ns monitor-agent-trace $tcp
$tcp tracevar cwnd_
$ns at 0.1 "$ftp start"
$ns at 3.0 "$ns detach-agent $n0 $tcp ;
$ns detach-agent $n1 $sink"
$ns at 3.5 "finish"
$ns at 0.0 "$ns trace-annotate \"Sliding Window with window size 4 (normal operation)\""
$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.1\""
$ns at 0.11 "$ns trace-annotate \"Send Packet_0,1,2,3\""
$ns at 0.34 "$ns trace-annotate \"Receive Ack_0,1,2,3\""
$ns at 0.56 "$ns trace-annotate \"Send Packet_4,5,6,7\""
$ns at 0.79 "$ns trace-annotate \"Receive Ack_4,5,6,7\""
```

```
$ns at 0.99 "$ns trace-annotate \"Send Packet_8,9,10,11\""
$ns at 1.23 "$ns trace-annotate \"Receive Ack_8,9,10,11 \""
$ns at 1.43 "$ns trace-annotate \"Send Packet_12,13,14,15\""
$ns at 1.67 "$ns trace-annotate \"Receive Ack_12,13,14,15\""
$ns at 1.88 "$ns trace-annotate \"Send Packet_16,17,18,19\""
$ns at 2.11 "$ns trace-annotate \"Receive Ack_16,17,18,19\""
$ns at 2.32 "$ns trace-annotate \"Send Packet_20,21,22,23\""
$ns at 2.56 "$ns trace-annotate \"Receive Ack_24,25,26,27\""
$ns at 2.76 "$ns trace-annotate \"Send Packet_28,29,30,31\""
$ns at 3.00 "$ns trace-annotate \"Receive Ack_28\""
$ns at 3.1 "$ns trace-annotate \"FTP
stops\""proc finish {} {
global ns
        $ns flush-trace
#       close $nf
puts "running nam..." exec nam sliding.nam & exit 0
}
$ns run
```

Output:



6. Result:

Thus the Stop and Wait protocol and Sliding Window Protocols are Simulated and studied.

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt afterperforming the experiment)

8. Conclusion:

(Students should write conclusion on their own)

9. Viva Questions:

- What is ARQ?
- What is stop and wait protocol?
- What is stop and wait ARQ?
- What is usage of sequence number in reliable transmission?
- What is sliding window?

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e).
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
d. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE)

Experiment No.: 6
Setup a network and configure IP addressing, subnetting, Masking using Cisco Packet Tracer

1. **Aim: S**etup a network and configure IP addressing, Subnetting, Masking using Cisco Packet Tracer

2. **Objectives:** To introduce concepts of configurations IP addressing, Subnetting and Masking.

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of various networking devices.
   - Use the simulation tool Cisco Packet Tracer for building networks.
   - Recognize the need for IP addressing, Subnetting and Masking

4. **Hardware/Software required:** Cisco Packet Tracer

5. **Theory**

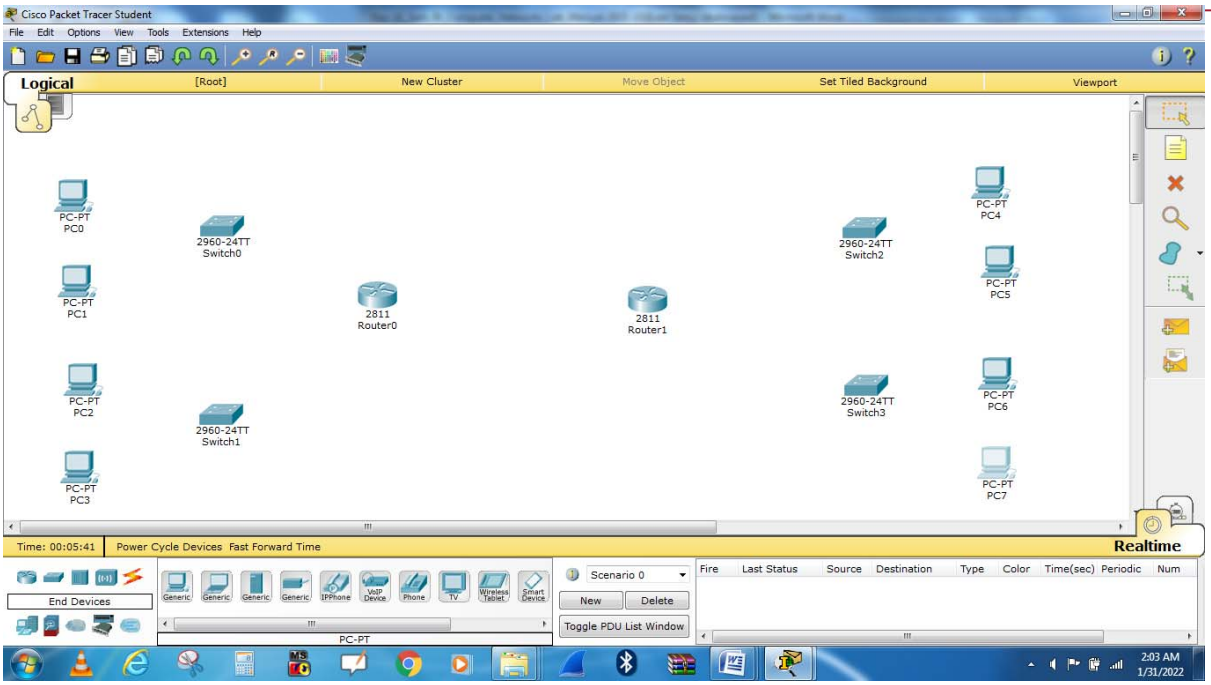
Setting a subnetwork

Four Dept.	
Comp Dept. = 136 Hosts	256 128 64 32 16 8 4 2 1 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
EL Dept. = 95 Hosts	IP Address of Original Network:
ET Dept. = 51 Hosts	192.168.1.0
Insrtu. Dept. = 24 Hosts	
Comp Dept (136 Hosts)	EL Dept (95)
2^8 = 256	2^7 = 128
Total Bits of IP Address = 32 bits	Total Bits of IP Address = 32 bits
32-8 = 24 (Subnet Mask is of 24 Bits)	32-7 = 25 (Subnet Mask is of 25 Bits)
192.168.1.0/24 IP Address: 192.168.1.0	192.168.2.0/25 IP Address: 192.168.2.0
Subnet Mask: 11111111 11111111 11111111 00000000 (Binary)	Subnet Mask: 11111111 11111111 11111111 10000000 (Binary)
Subnet Mask: 255.255.255.0 (Dotted Decimal)	Subnet Mask: 255.255.255.128 (Dotted Decimal)
First Host IP Address: **192.168.1.1**	First Host IP Address: **192.168.2.1**
Last Host IP Address: **192.168.1.254**	Last Host IP Address: **192.168.2.126**
192.168.1.0- Network Address	192.168.2.0 - Network Address

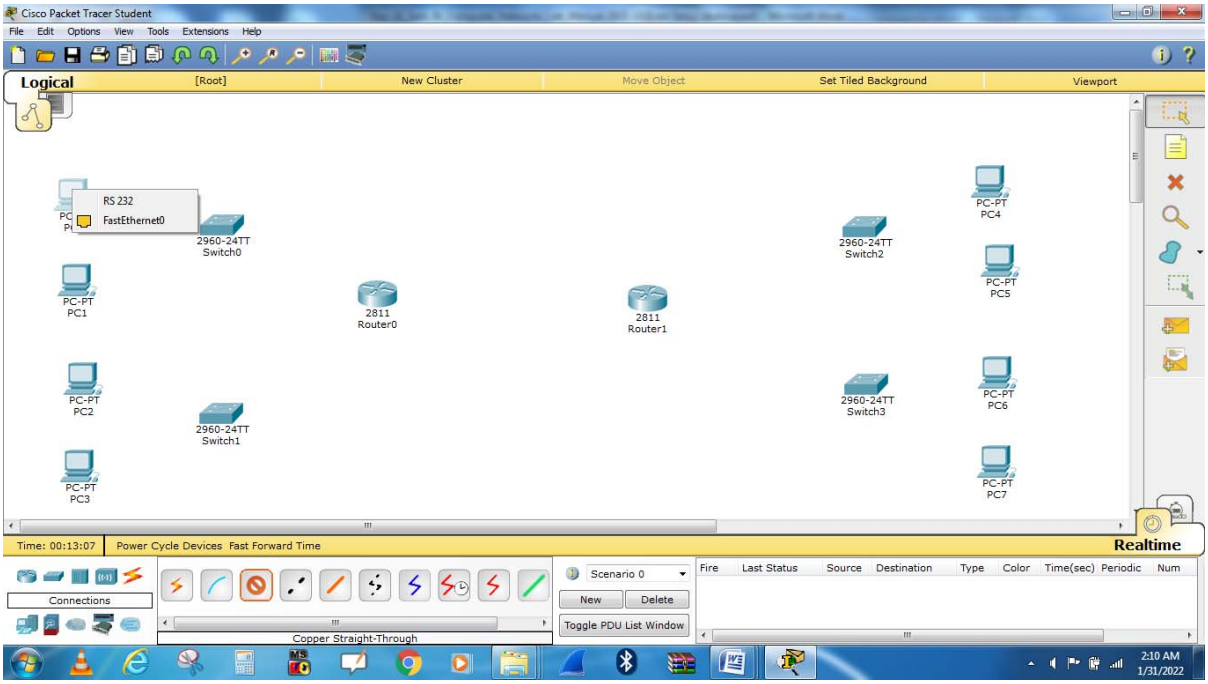
Four Dept.	
Comp Dept. = 136 Hosts	256 128 64 32 16 8 4 2 1 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
EL Dept. = 95 Hosts	IP Address of Original Network:
ET Dept. = 51 Hosts	192.168.1.0
Insrtu. Dept. = 24 Hosts	
ET Dept (51 Hosts)	Instru. Dept (24)
2^6 = 64	2^5 = 32
Total Bits of IP Address = 32 bits	Total Bits of IP Address = 32 bits
32-6 = 26 (Subnet Mask is of 26Bits)	32-5 = 27 (Subnet Mask is of 27 Bits)
192.168.2.0/26 IP Address: 192.168.2.128	192.168.2.192/27 IP Address: 192.168.2.192
Subnet Mask: 11111111 11111111 11111111 11000000 (Binary)	Subnet Mask: 11111111 11111111 11111111 11100000 (Binary)
Subnet Mask: 255.255.255.192 (Dotted Decimal)	Subnet Mask: 255.255.255.224 (Dotted Decimal)
First Host IP Address: **192.168.2.129**	First Host IP Address: **192.168.2.193**
Last Host IP Address: **192.168.2.190**	Last Host IP Address: **192.168.2.222**

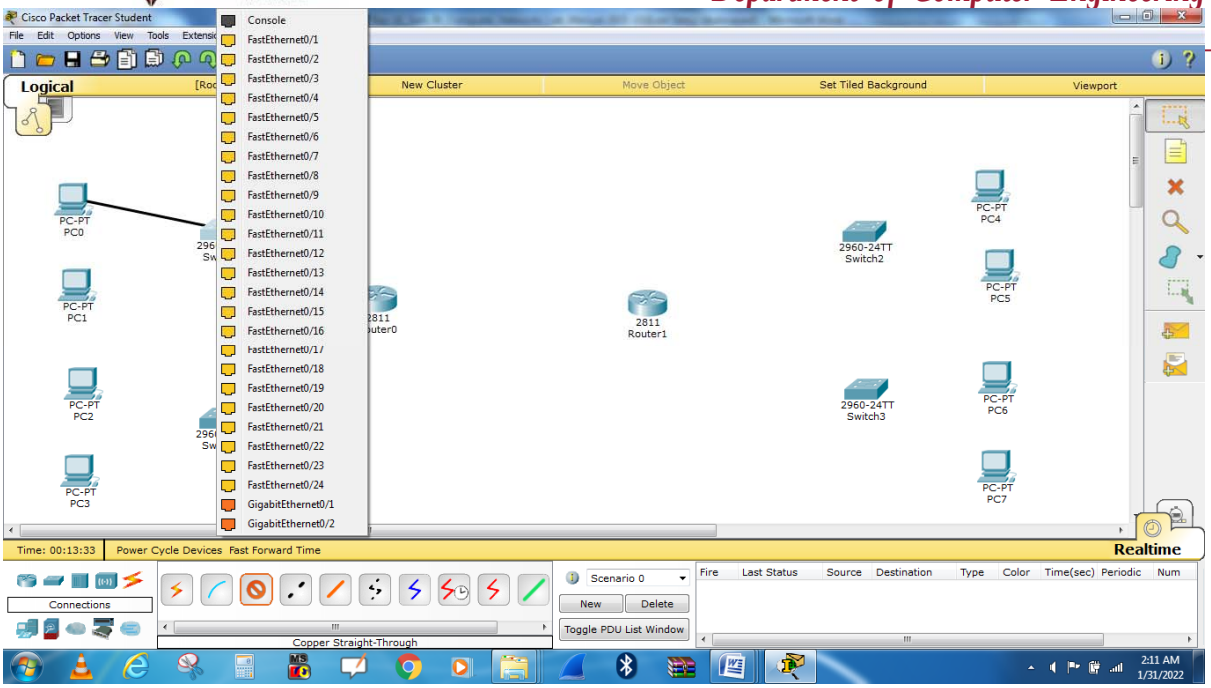
Four Dept.	
Comp Dept. = 136 Hosts	256 128 64 32 16 8 4 2 1 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
EL Dept. = 95 Hosts	IP Address of Original Network:
ET Dept. = 51 Hosts	192.168.1.0
Insrtu. Dept. = 24 Hosts	
Router (2)	
2^2= 4	
Total Bits of IP Address = 32 bits	
32-2 = 30 (Subnet Mask is of 30 Bits)	
192.168.2.192/30 IP Address: 192.168.2.224	
Subnet Mask: 11111111 11111111 11111111 11111100 (Binary)	
Subnet Mask: 255.255.255.252 (Dotted Decimal)	
First Host IP Address: **192.168.2.225**	
Last Host IP Address: **192.168.2.226**	
192.168.2.224 - Network Address 192.168.2.227- Broadcast Address	

1. Drag and drop the router/switch/computer from the bottom of the screen.

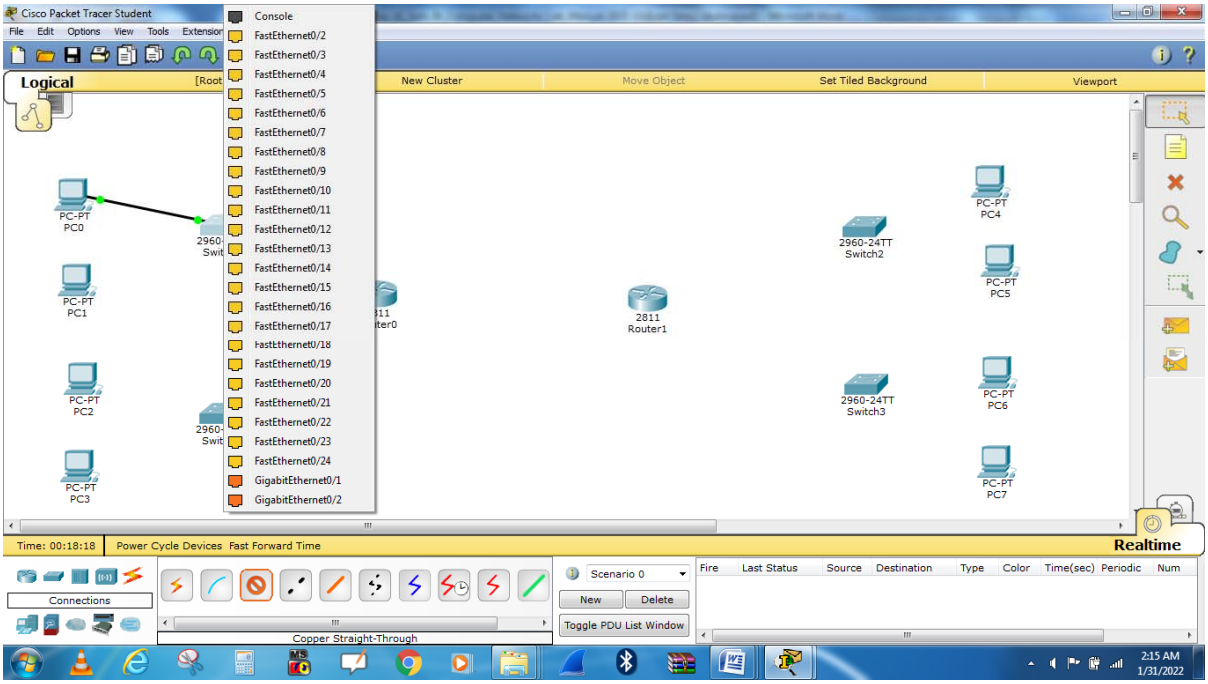


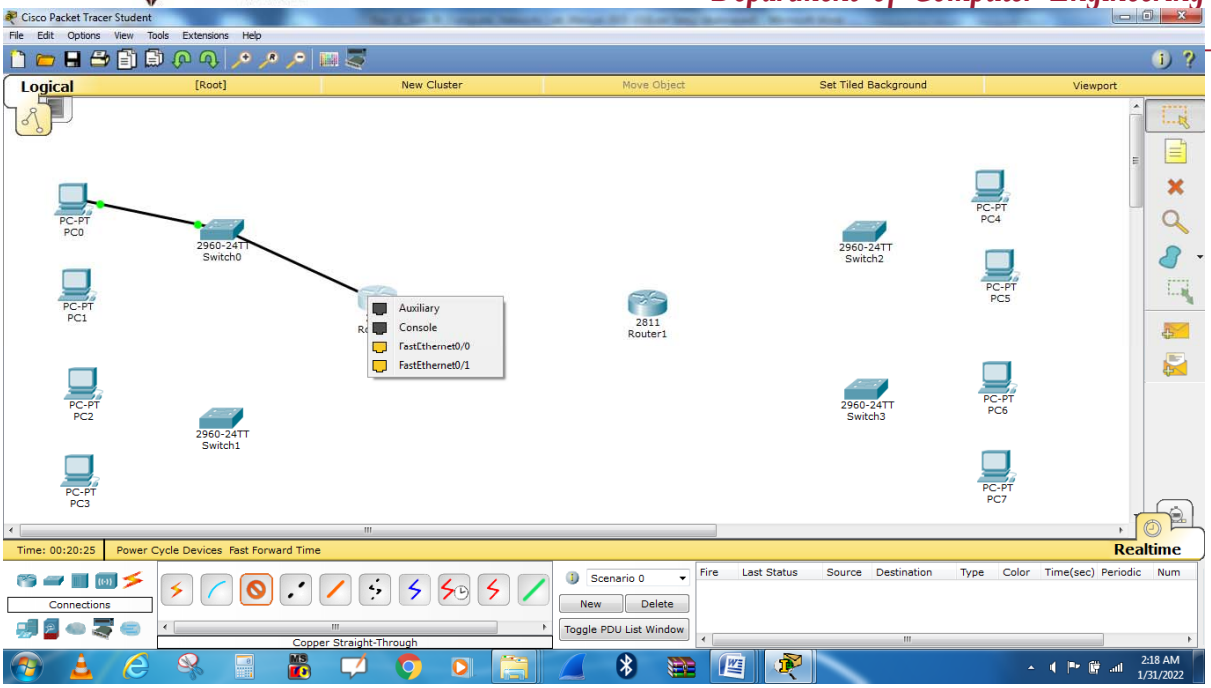
1. Select **end devices** from the bottom left-hand corner and drag it to the sandbox screen.
2. Select **connections** from the same bottom left-hand corner. When you connect like-devices (Such as a router and computer) you use a crossover cable, so you should select Copper straight-Through, Click on PC0 to Switch0 and *Router0*.



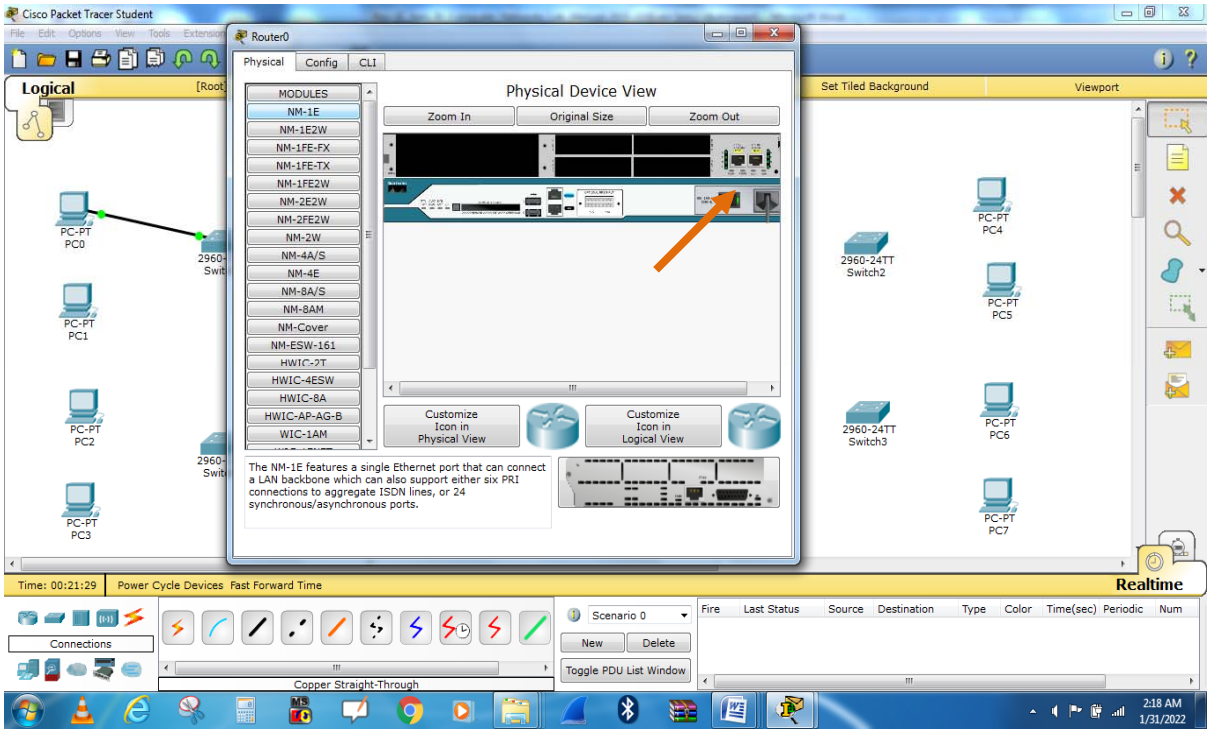


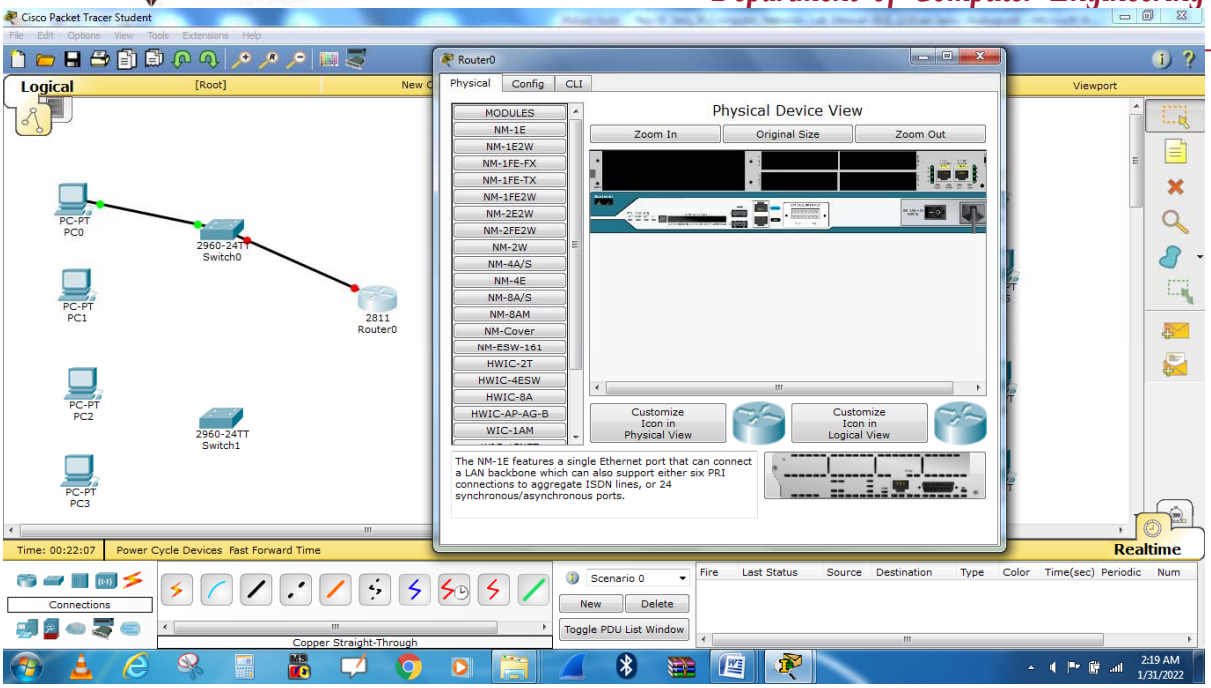
3. Repeat step number 2to connect **PC1,PC2,PC3, and Switch1 to Router0.**



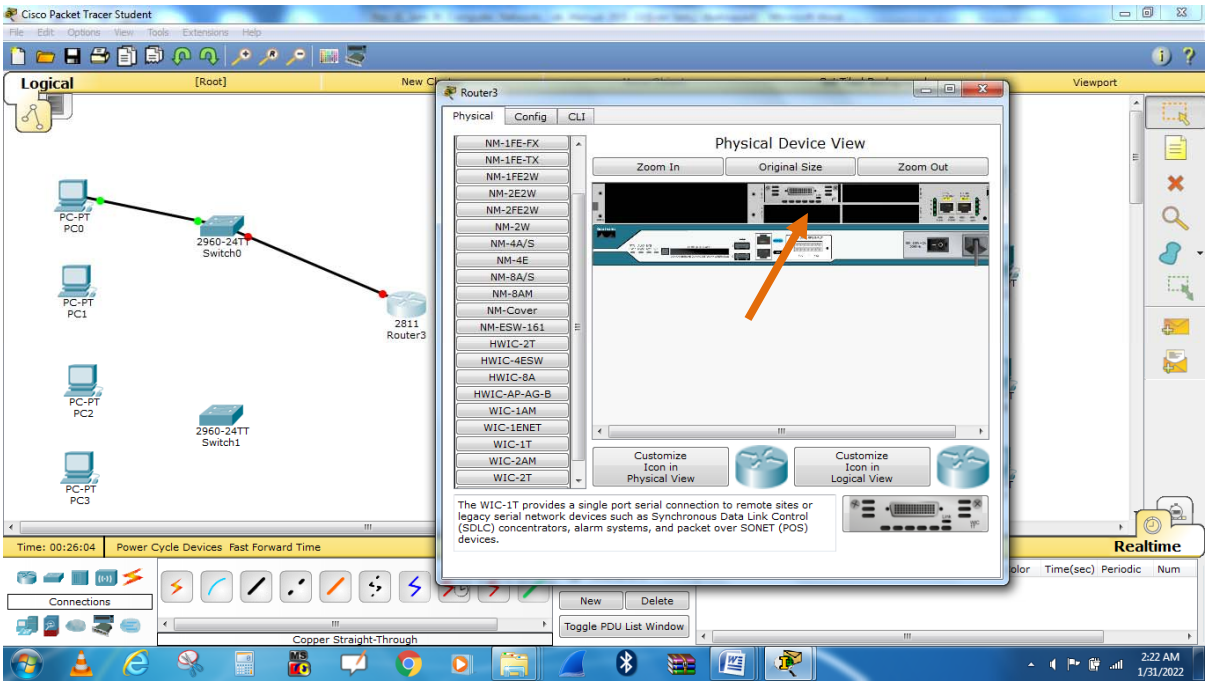


To connect router0 and router1, serial connection is required, Click on router0, First turn off router shown by arrow.

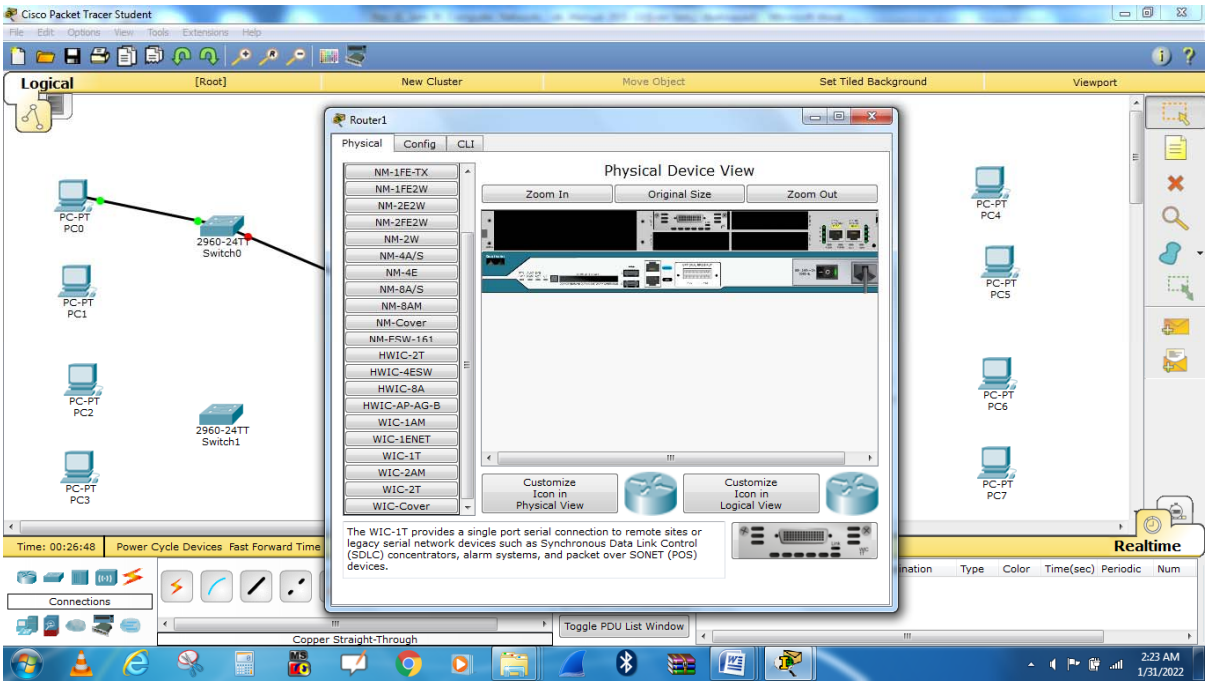


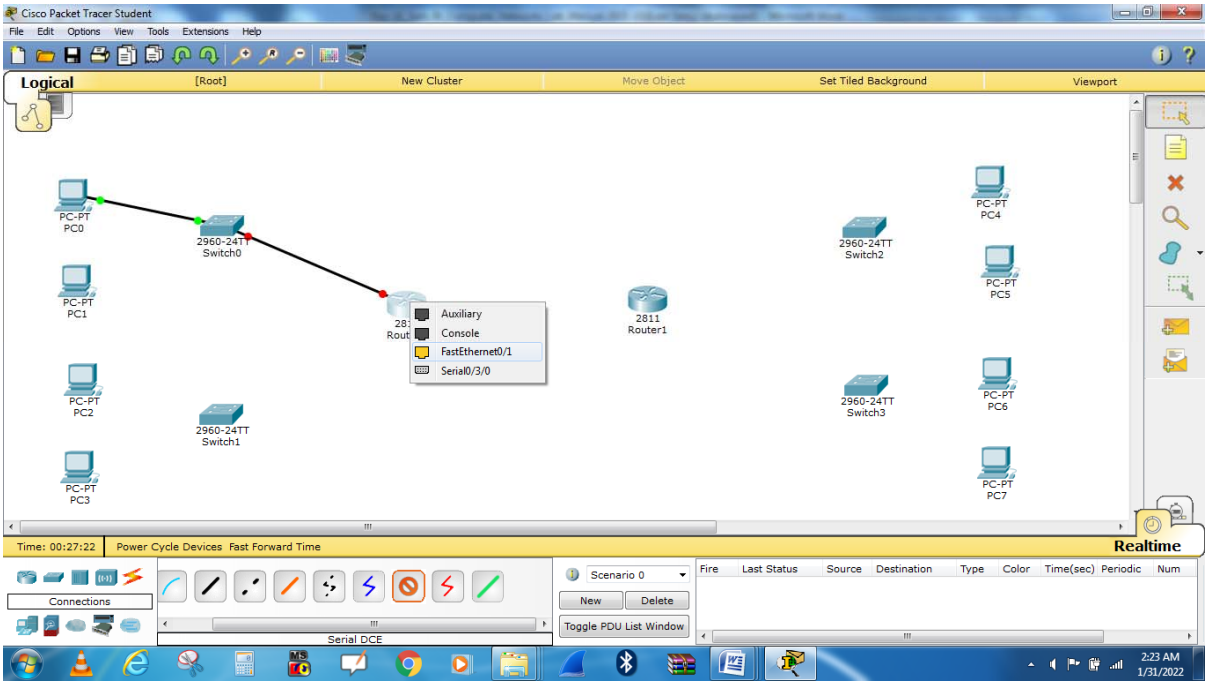


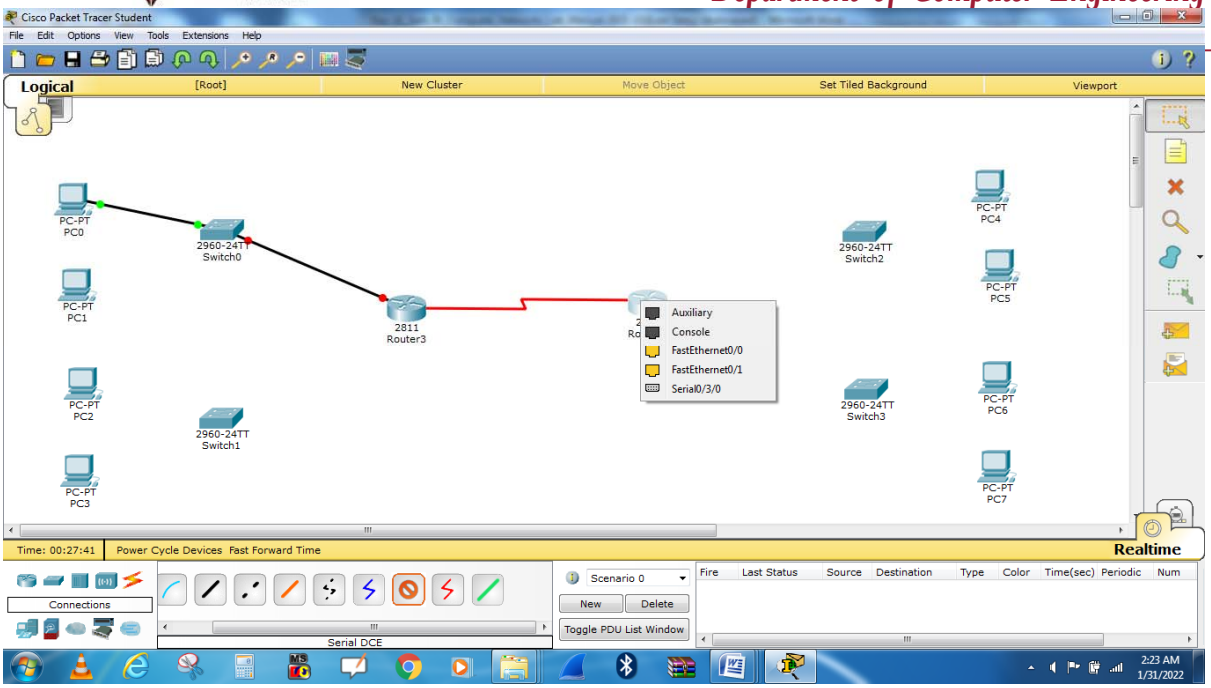
Select WIC 1T interface and Place it to slot shown by arrow.



Repeat same procedure for router1







Configuring the router in packet tracer

4. Next we have to open the Ethernet ports to allow communication. Although they are physically connected, they are in a state that is known as being in **administrative shut down**. Now click on the **CLI** tab to access the configuration menu.

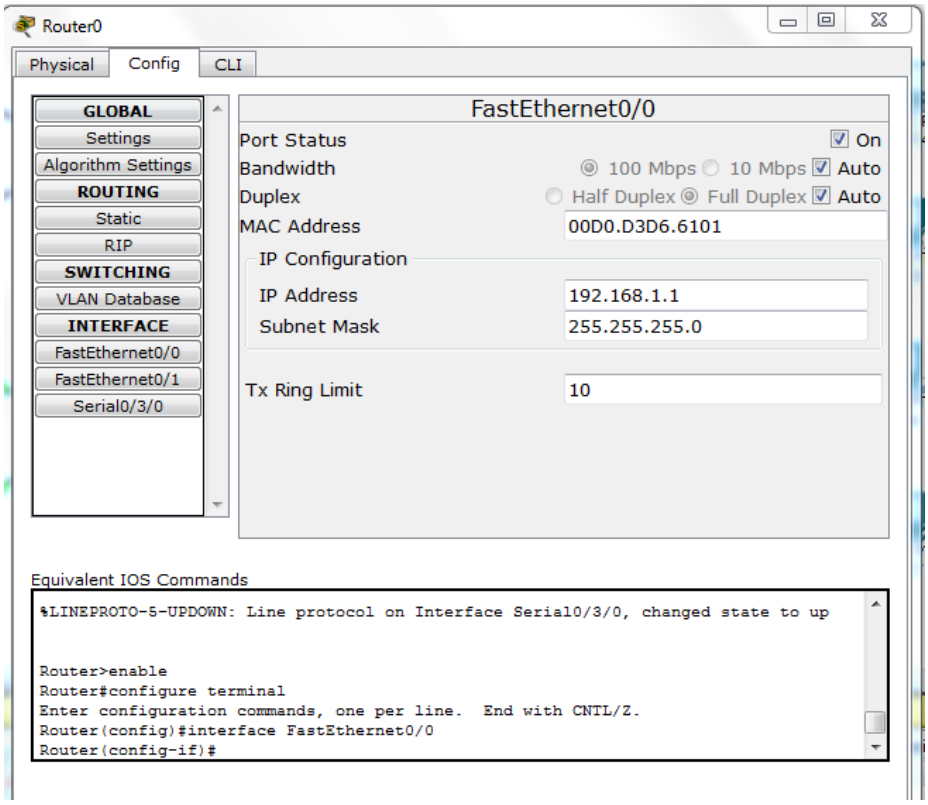
Router0

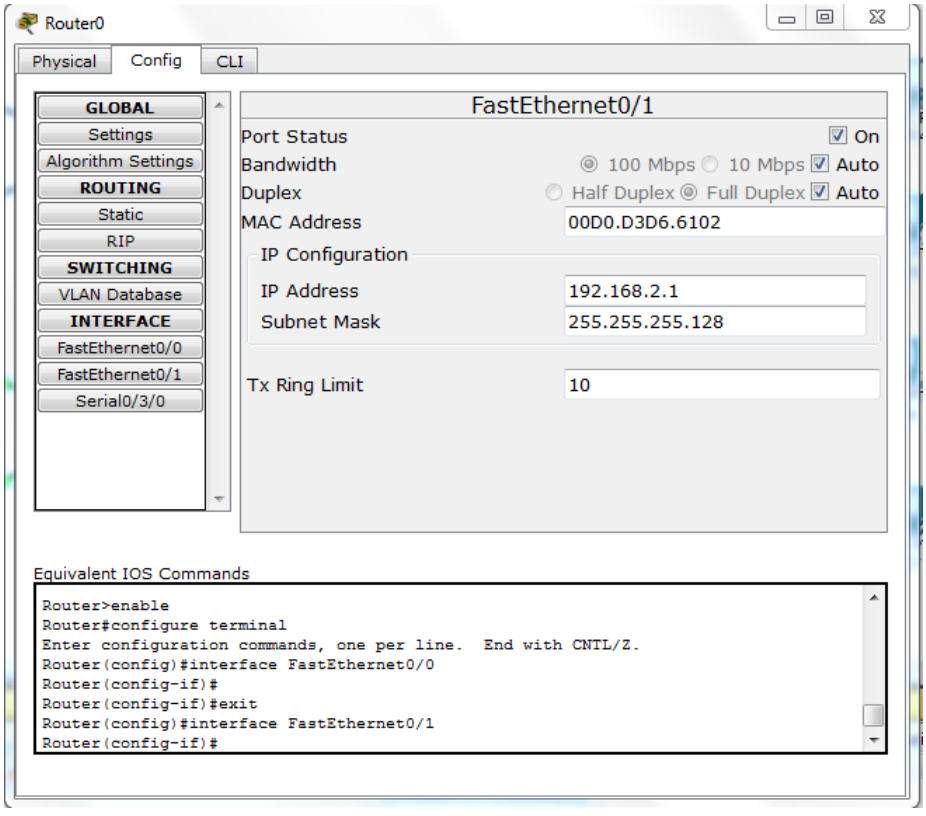
```
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n
Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
ip address 192.168.1.1 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
```

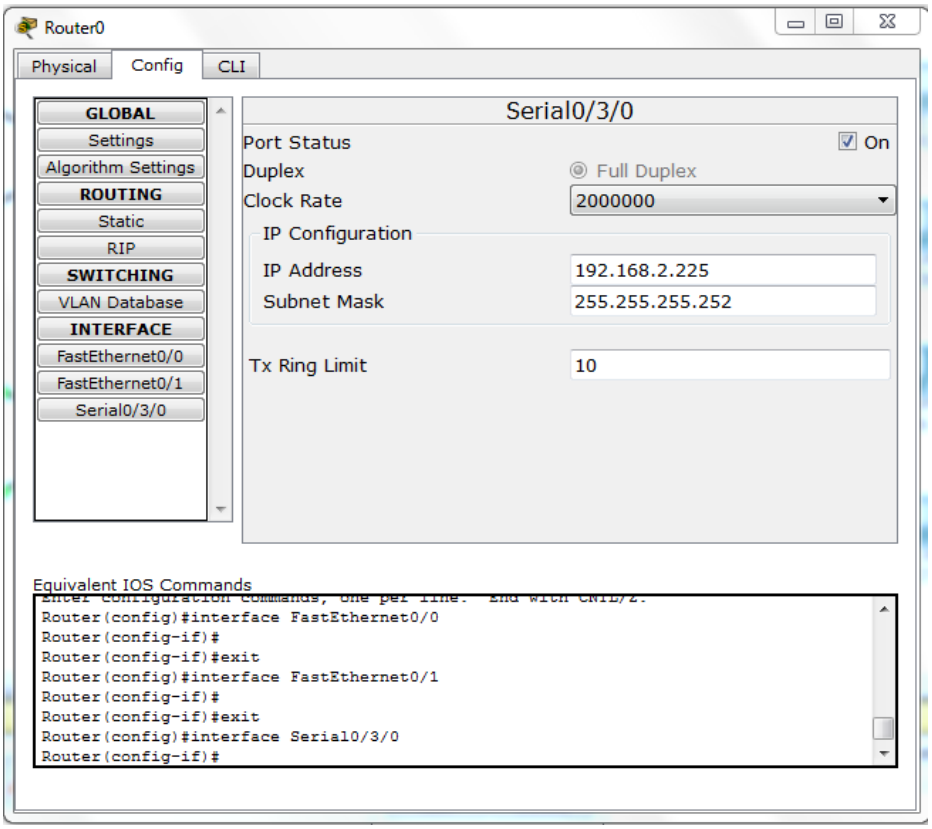
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
ip address 192.168.2.1 255.255.255.0
Router(config-if)#ip address 192.168.2.1 255.255.255.128
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/3/0, changed state to down
Router(config-if)#no ip address
Router(config-if)#ip address 192.168.2.225 255.255.255.128
Router(config-if)#no ip address
Router(config-if)#ip address 192.168.2.225 255.255.255.128
Router(config-if)#ip address 192.168.2.225 255.255.255.252
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to up
Router con0 is now available
Press RETURN to get started.
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 192.168.2.128 255.255.255.192 192.168.2.226
Router(config)#ip route 192.168.2.192 255.255.255.224 192.168.2.226
Router(config)#
Router(config)#end
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router con0 is now available
Press RETURN to get started.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to up


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
Serial0/3/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00D0.D3D6.6101
IP Configuration
IP Address
192.168.1.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to up
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
Serial0/3/0
FastEthernet0/1
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00D0.D3D6.6102
IP Configuration
IP Address
192.168.2.1
Subnet Mask
255.255.255.128
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#


Router0
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
Serial0/3/0
Serial0/3/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IP Address
192.168.2.225
Subnet Mask
255.255.255.252
Tx Ring Limit
10
Equivalent IOS Commands
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#

Router1

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n

Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
no ip address
Router(config-if)#ip address 192.168.2.129 255.255.255.0
Router(config-if)#no ip address
Router(config-if)#ip address 192.168.2.129 255.255.255.0
Router(config-if)#ip address 192.168.2.129 255.255.255.192
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#no shutdown
Router(config-if)#shutdown
Router(config-if)#no shutdown
Router(config-if)#shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

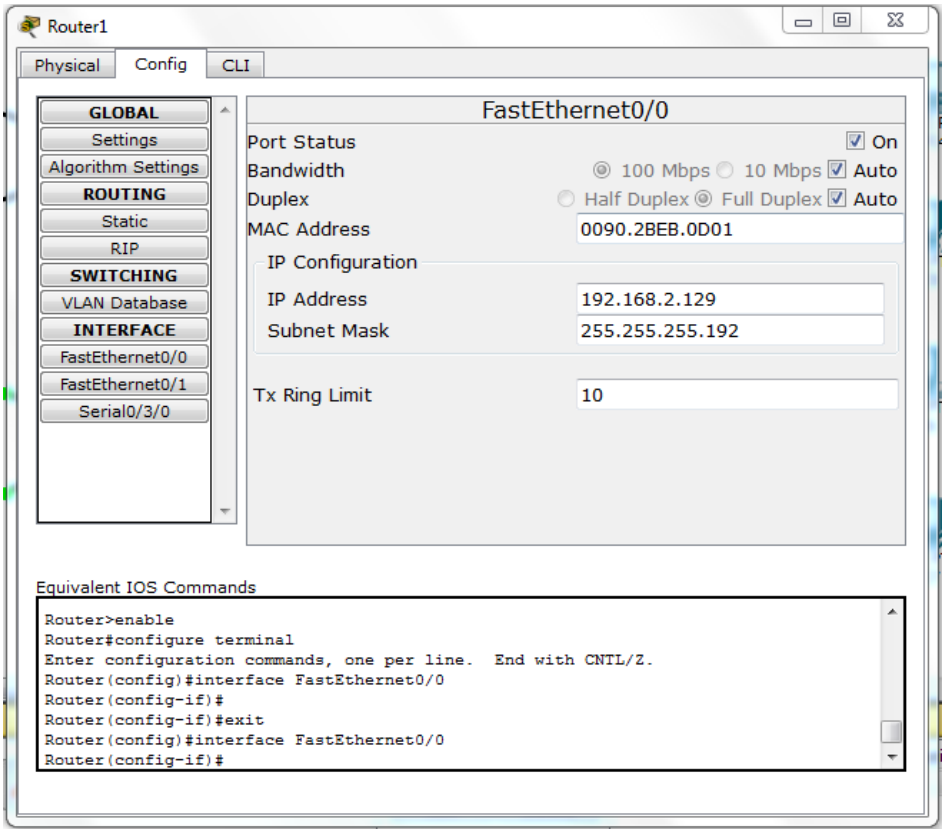
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

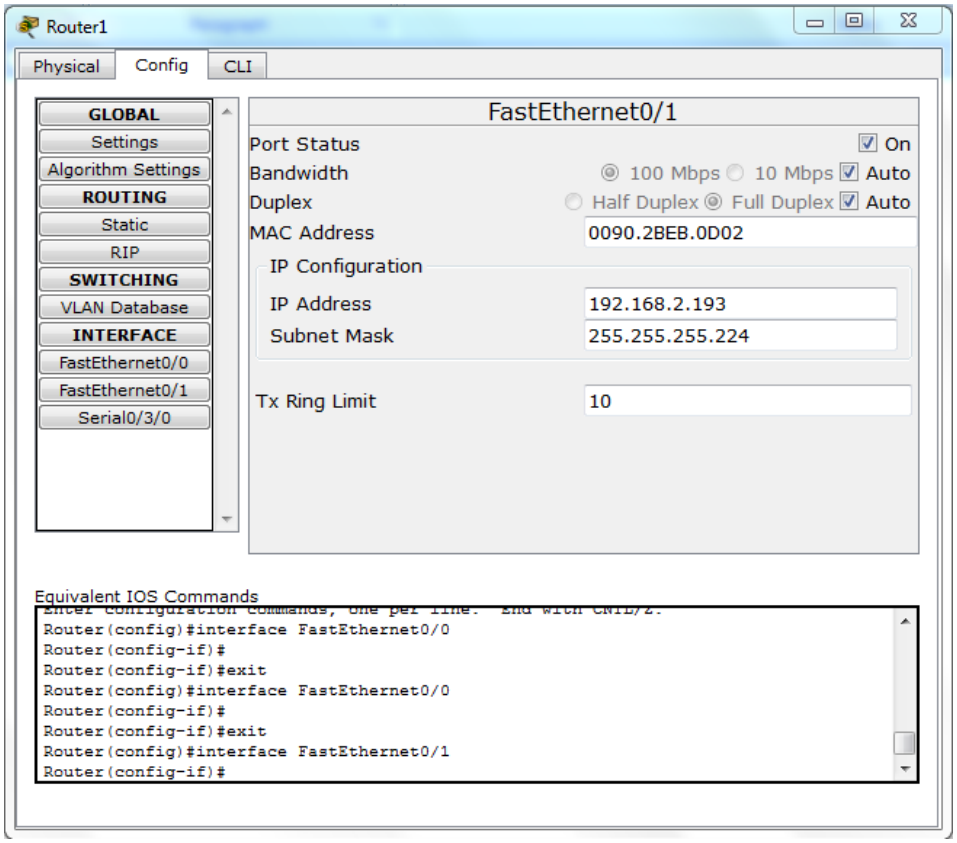
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

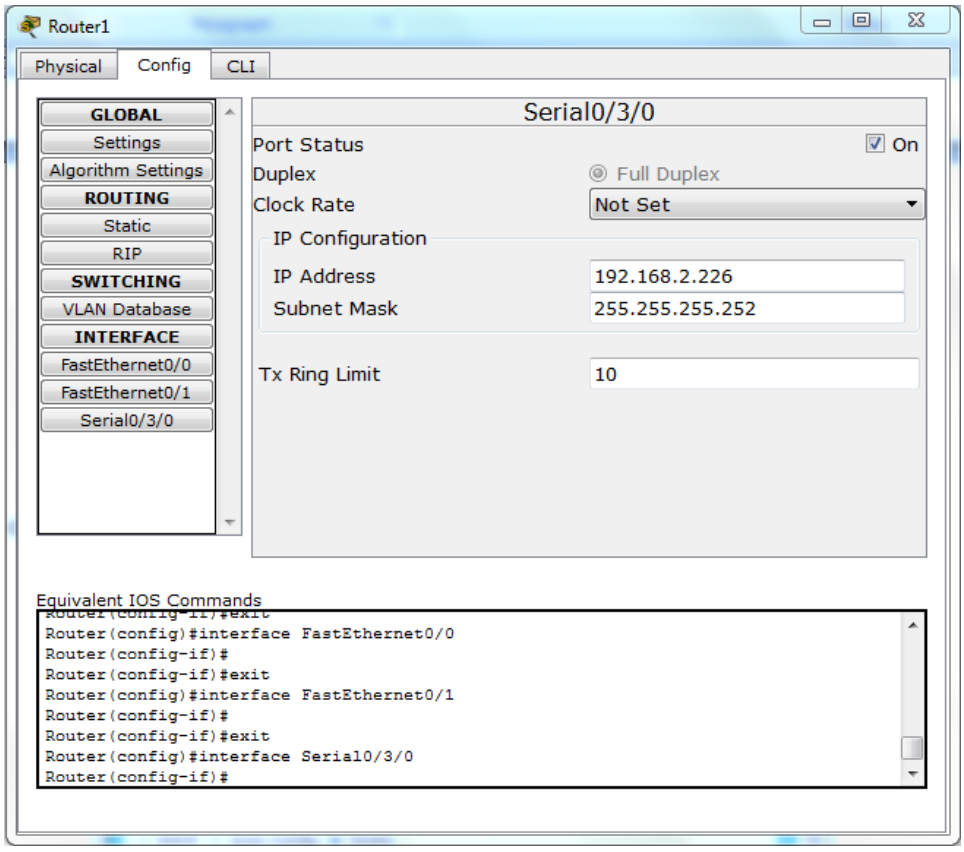
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down no shutdown

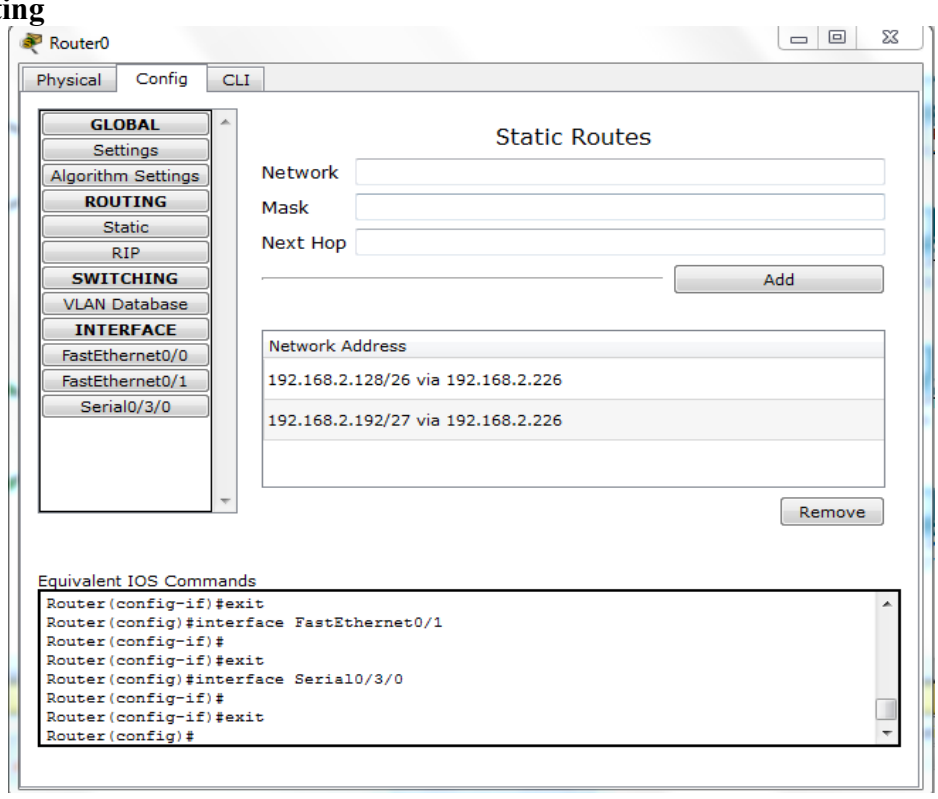
```
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
ip address 192.168.2.193 255.255.255.192
Router(config-if)#ip address 192.168.2.193 255.255.255.224
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to up
ip address 192.168.2.226 255.255.255.192
% 192.168.2.192 overlaps with FastEthernet0/1
Router(config-if)#ip address 192.168.2.226 255.255.255.252
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#
Router con0 is now available
Press RETURN to get started.
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.225
Router(config)#ip route 192.168.2.0 255.255.255.128 192.168.2.225
Router(config)#
Router(config)#end
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router con0 is now available
Press RETURN to get started.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3/0, changed state to up
```

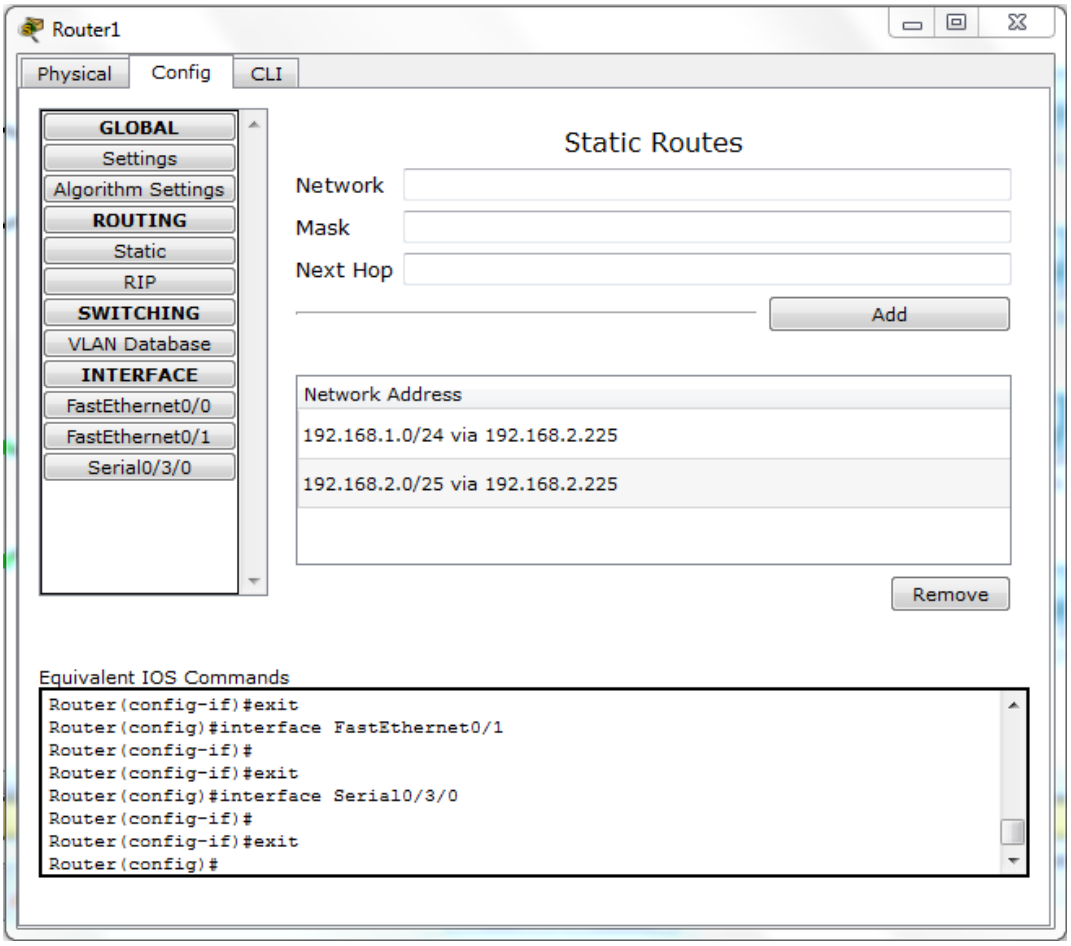

Router1
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
Serial0/3/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0090.2BEB.0D01
IP Configuration
IP Address
192.168.2.129
Subnet Mask
255.255.255.192
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#


Router1
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
Serial0/3/0
FastEthernet0/1
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0090.2BEB.0D02
IP Configuration
IP Address
192.168.2.193
Subnet Mask
255.255.255.224
Tx Ring Limit
10
Equivalent IOS Commands
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#



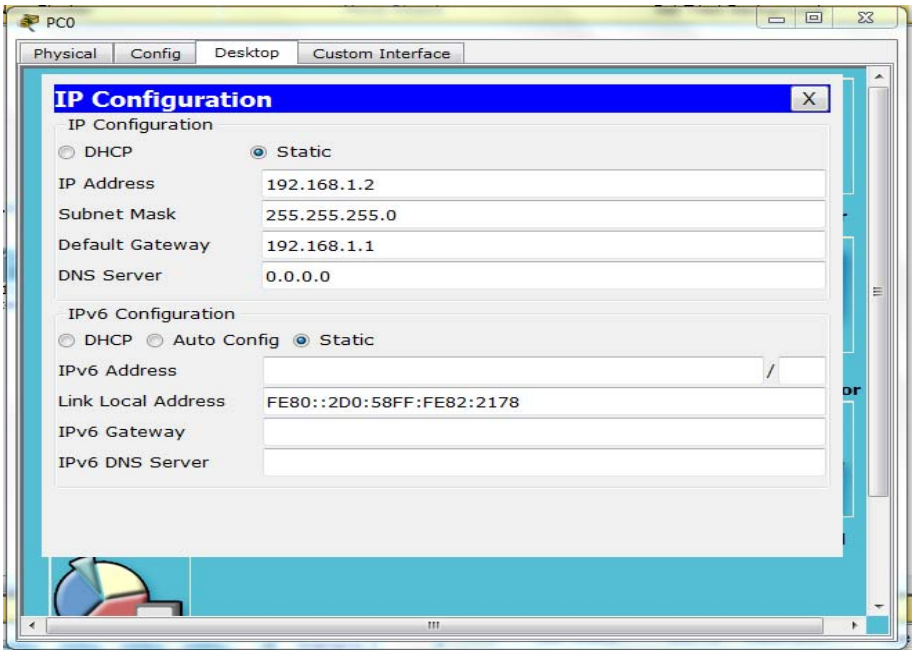
Static Routing

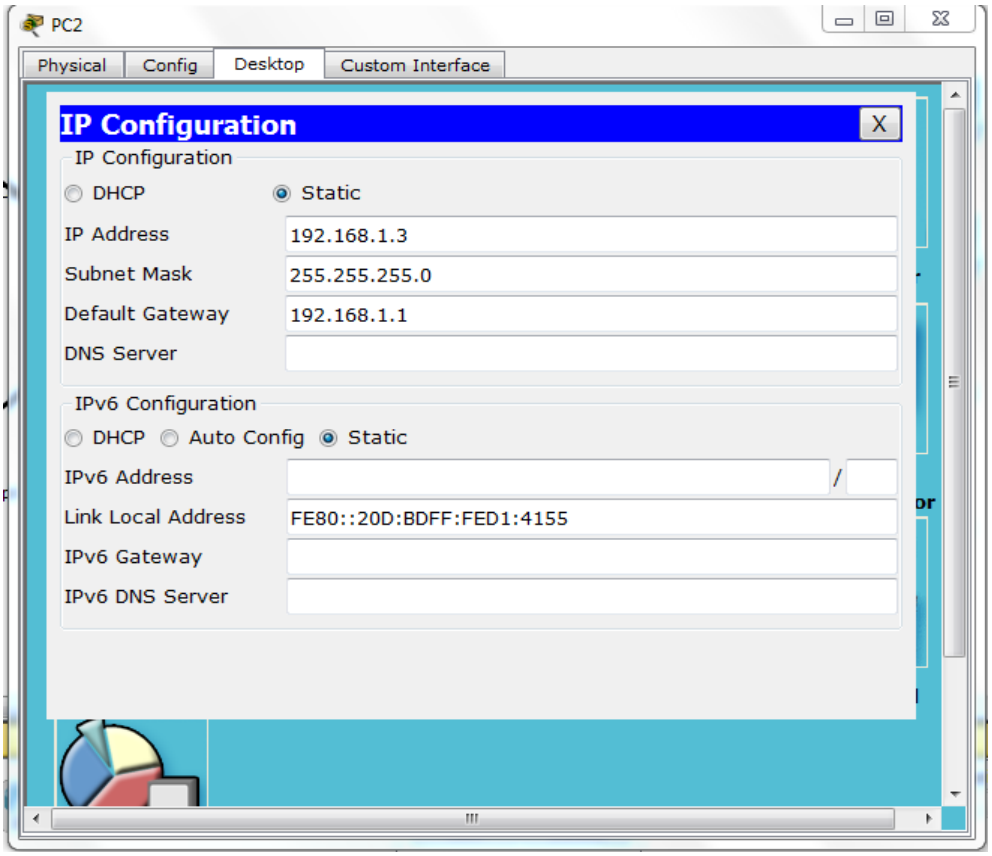


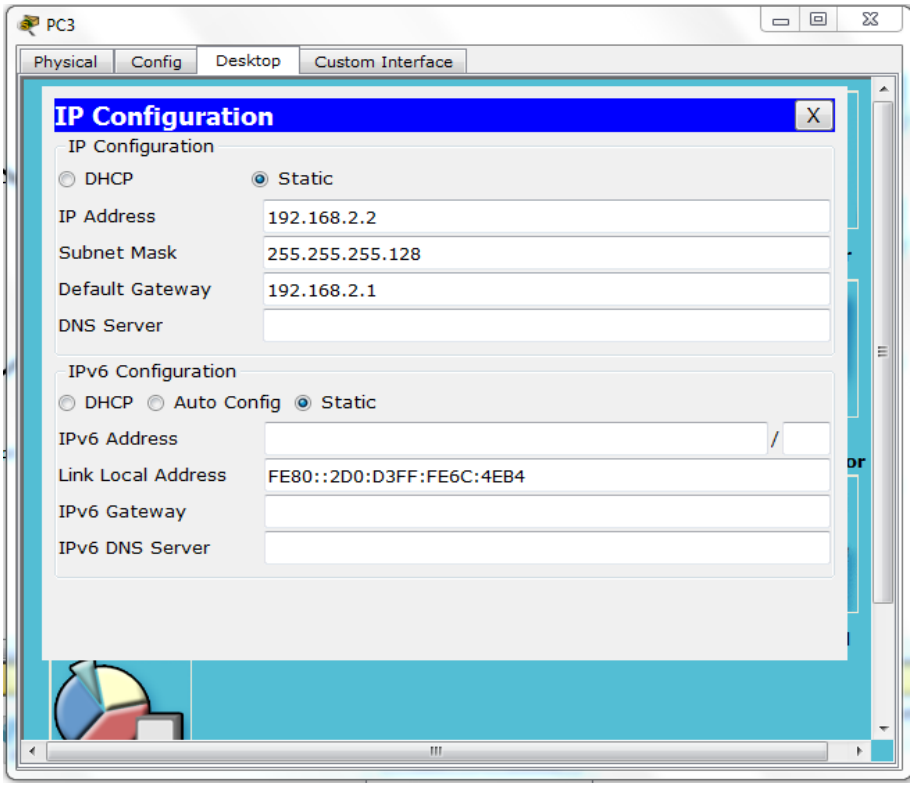


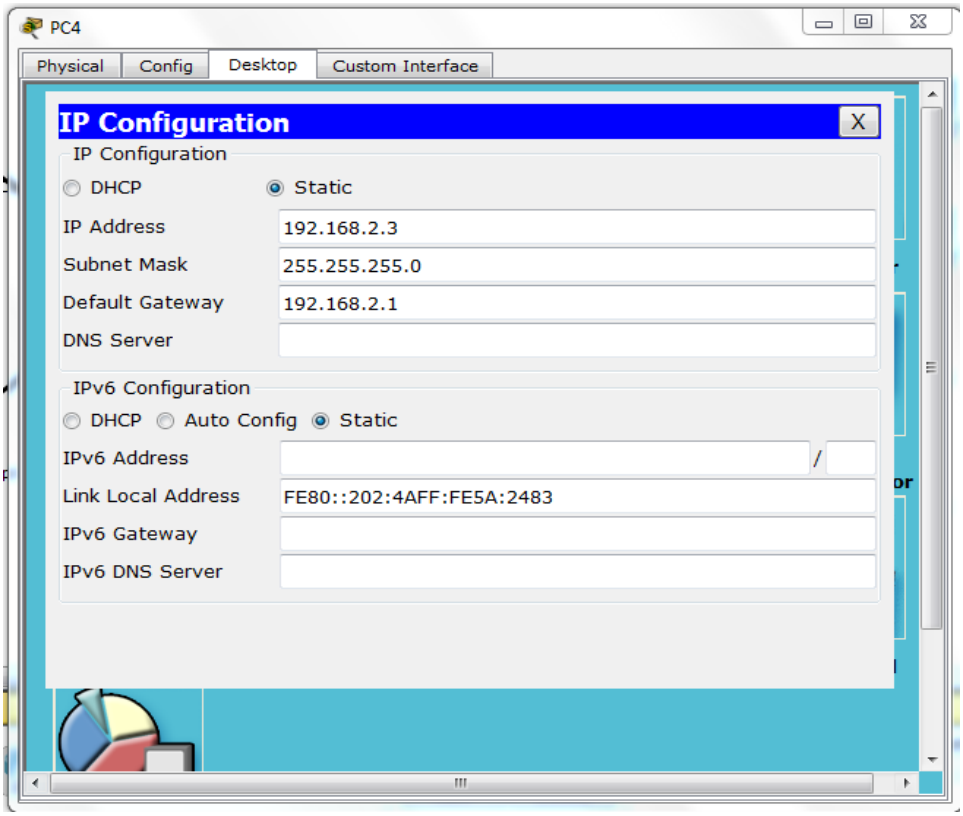
Go to **Config >settings** save configuration of static routing for Router0 and Router1.

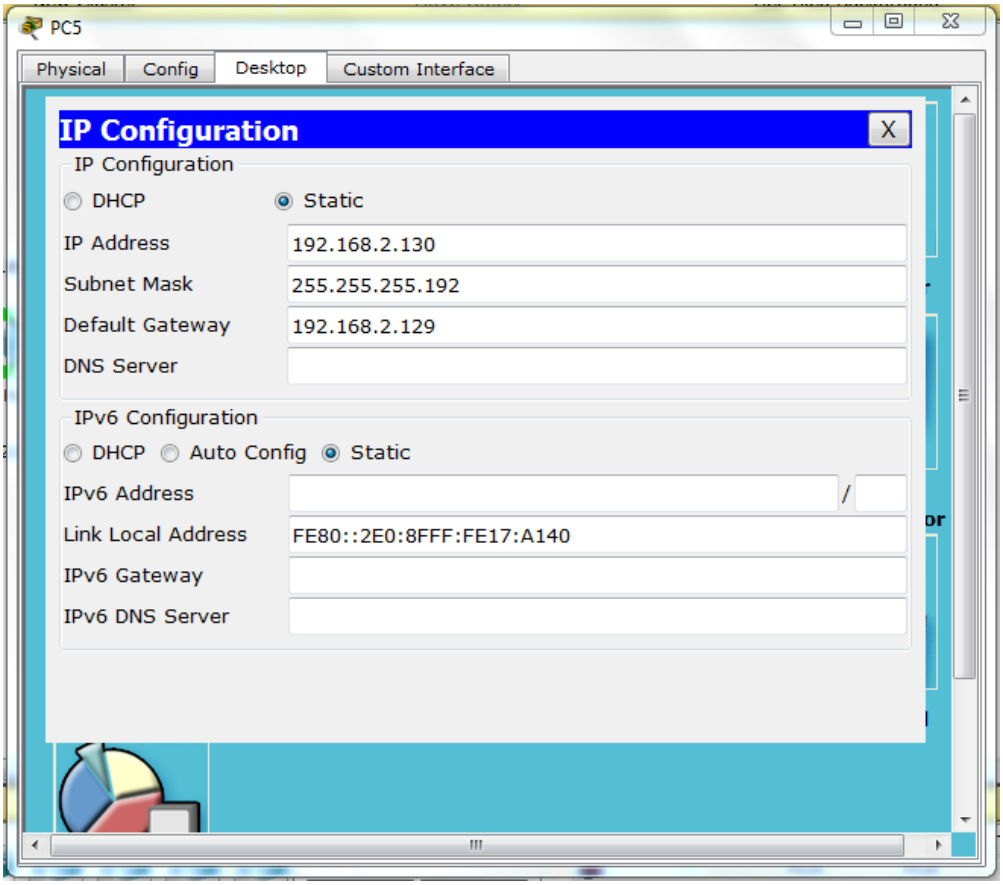
Configuring the PC in packet tracer

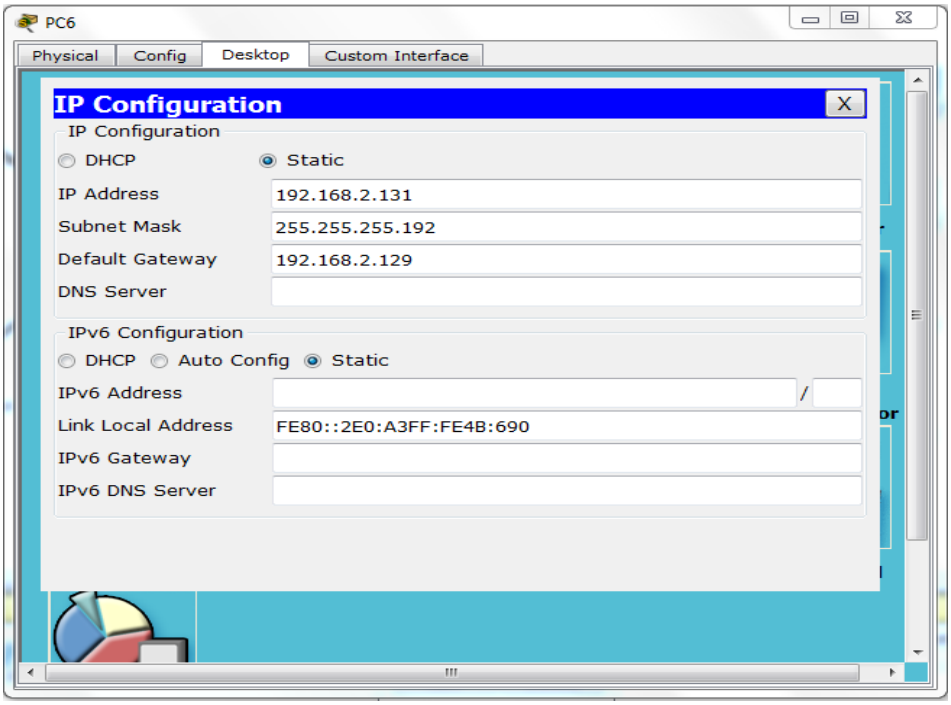


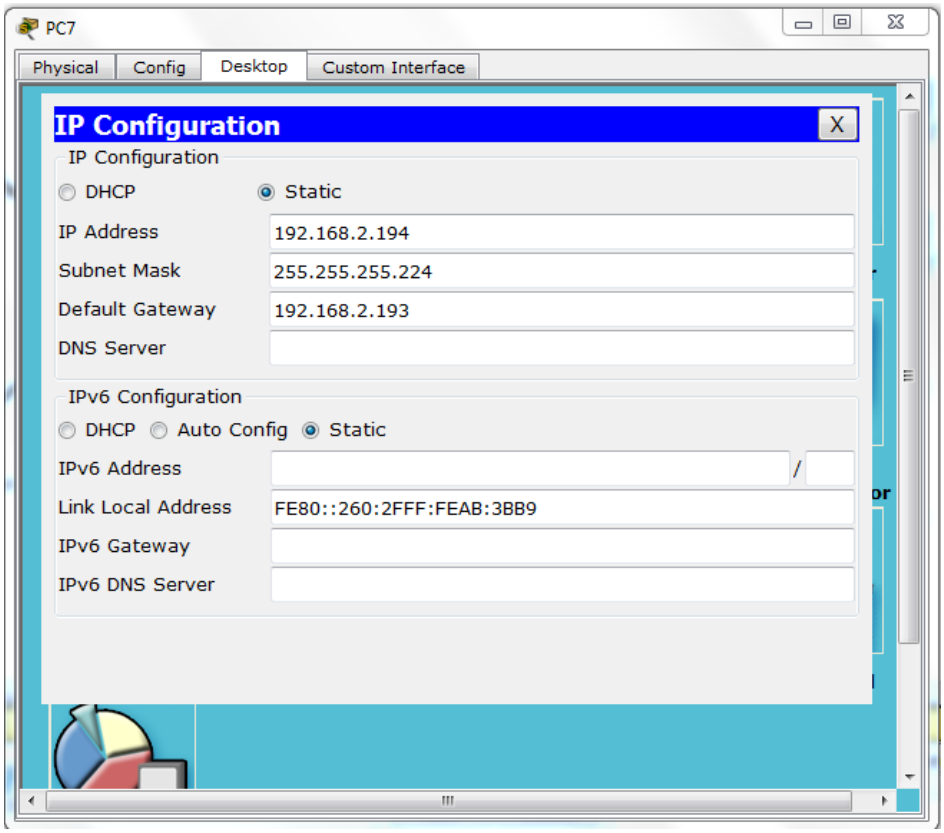


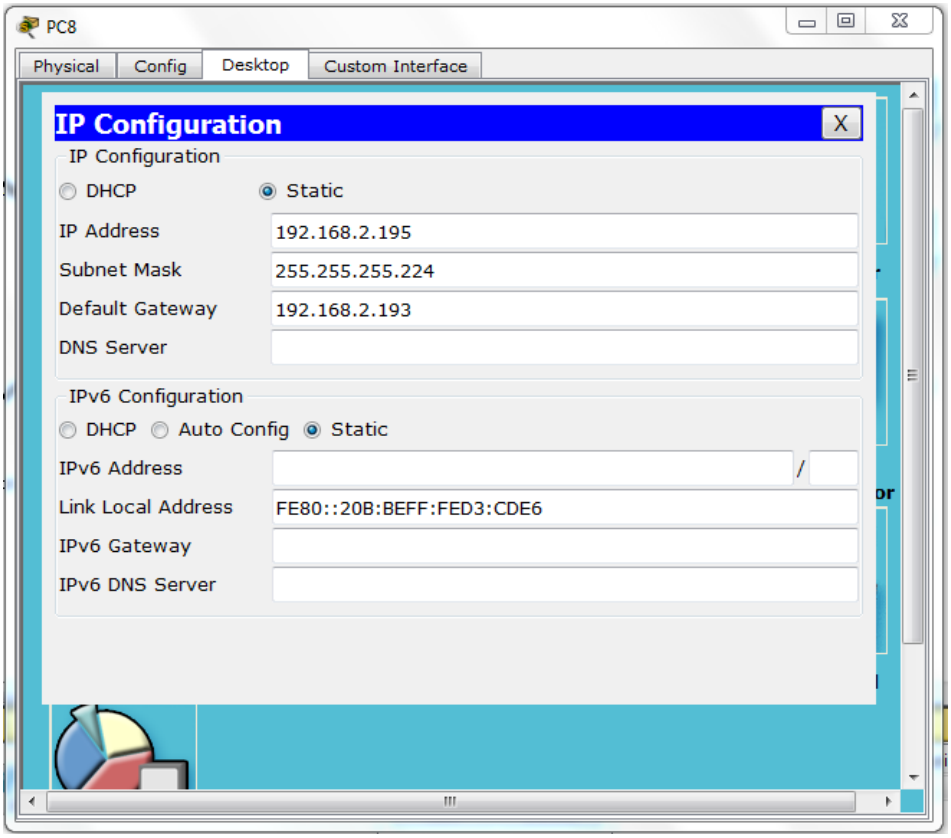

PC3
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.2
Subnet Mask
255.255.255.128
Default Gateway
192.168.2.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2D0:D3FF:FE6C:4EB4
IPv6 Gateway
IPv6 DNS Server


PC4
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.3
Subnet Mask
255.255.255.0
Default Gateway
192.168.2.1
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::202:4AFF:FE5A:2483
IPv6 Gateway
IPv6 DNS Server


PC5
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.130
Subnet Mask
255.255.255.192
Default Gateway
192.168.2.129
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2E0:8FFF:FE17:A140
IPv6 Gateway
IPv6 DNS Server


PC6
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.131
Subnet Mask
255.255.255.192
Default Gateway
192.168.2.129
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::2E0:A3FF:FE4B:690
IPv6 Gateway
IPv6 DNS Server

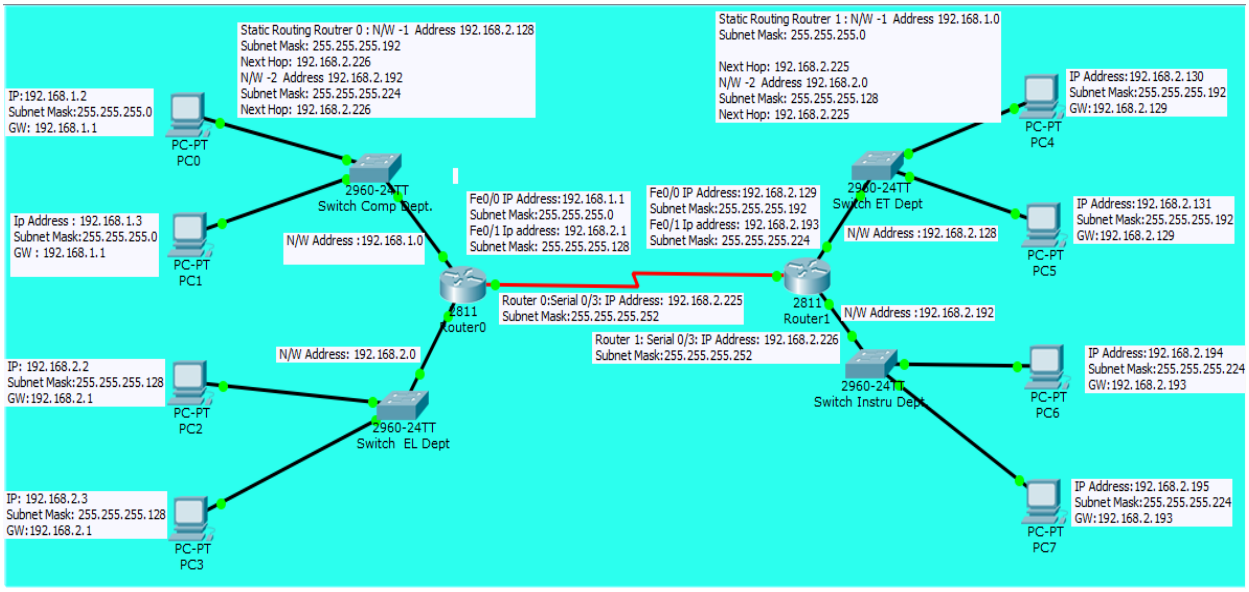

PC7
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.194
Subnet Mask
255.255.255.224
Default Gateway
192.168.2.193
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::260:2FFF:FEAB:3BB9
IPv6 Gateway
IPv6 DNS Server


PC8
Physical
Config
Desktop
Custom Interface
IP Configuration
X
IP Configuration
DHCP
Static
IP Address
192.168.2.195
Subnet Mask
255.255.255.224
Default Gateway
192.168.2.193
DNS Server
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::20B:BEFF:FED3:CDE6
IPv6 Gateway
IPv6 DNS Server

Router Configuration:

Router0

```
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n
Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
ip address 192.168.1.1 255.255.255.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
ip address 192.168.2.1 255.255.255.0
Router(config-if)#no ip address
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#ip address 192.168.2.1 255.255.255.128
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/3/0, changed state to down
Router(config-if)#ip address 192.168.2.225 255.255.255.128
Router(config-if)#ip address 192.168.2.225 255.255.255.252
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/3/0
Router(config-if)#
```



6. Output Analysis:

(Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. Conclusion :

(Students should write conclusion on their own)

9. Viva Questions:

- State advantages and disadvantages of each of bluetooth
- Name different layers of Bluetooth protocol stack.
- What is piconet and scatternet.

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e)
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 7
Implement Go back-N Protocol and Selective Repeat Protocol

1. **Aim:** Implement Go back-N Protocol and Selective Repeat Protocol using Network Simulator (NS2)
2. **Objectives:** To introduce NS2 simulator

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of Go back-N Protocol
   - Analyze the functioning of Selective Repeat Protocol
   - Simulate and Study Go back-N Protocol / Selective Repeat Protocol

4. **Hardware/Software required:** NS2 Simulator

5. **Theory**

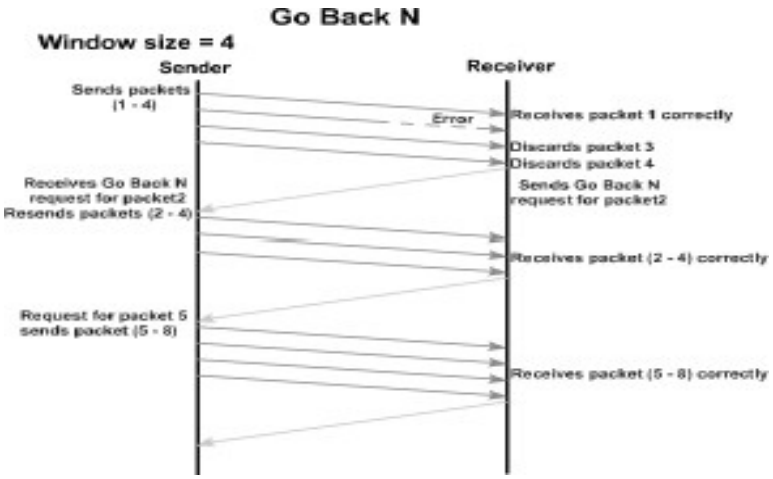
Go back-N is a connection oriented transmission. The sender transmits the frames continuously. Each frame in the buffer has a sequence number starting from 1 and increasing up to the window size. The sender has a window i.e. a buffer to store the frames. This buffer size is the number of frames to be transmitted continuously. The size of the window depends on the protocol designer.

Operation:

1. A station may send multiple frames as allowed by the window size.
2. Receiver sends an ACK i if frame i has an error. After that, the receiver discards all incomingframes until the frame with error is correctly retransmitted.
3. If sender receives an ACK i it will retransmit frame i and all packets i+1, i+2,... which have beensent, but not been acknowledged

Algorithm:

1. The source node transmits the frames continuously.
2. Each frame in the buffer has a sequence number starting from 1 and increasing up to the windowsize.
3. The source node has a window i.e. a buffer to store the frames. This buffer size is the number of frames to be transmitted continuously.
4. The size of the window depends on the protocol designer.
5. For the first frame, the receiving node forms a positive acknowledgement if the frame is receivedwithout error.
6. If subsequent frames are received without error (up to window size) cumulative positive acknowledgement is formed.
7. If the subsequent frame is received with error, the cumulative acknowledgment error-free framesare transmitted. If in the same window two frames or more frames are received with error, the second and the subsequent error frames are neglected. Similarly, even the frames received withouterror after the receipt of a frame with error are neglected.
8. The source node retransmits all frames of window from the first error frame.
9. If the frames are errorless in the next transmission and if the acknowledgment is error free, thewindow slides by the number of error-free frames being transmitted.
10. If the acknowledgment is transmitted with error, all the frames of window at source are retransmitted, and window doesn't slide.
11. This concept of repeating the transmission from the first error frame in the window is called asGOBACKN transmission flow control protocol

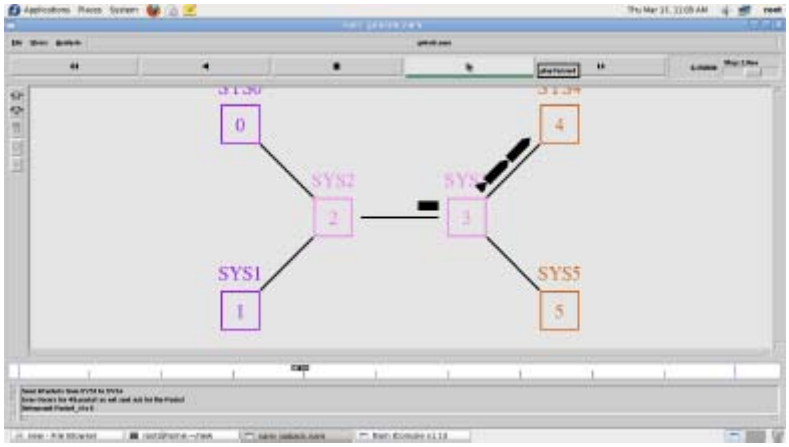


Program:

```
#send packets one by oneset
ns [new Simulator] set n0
[$ns node]
et n1 [$ns node] set
n2 [$ns node]set n3
[$ns node]set n4
[$ns node]set n5
[$ns node]
$n0 color "purple"
$n1 color "purple"
$n2 color "violet"
$n3 color "violet"
$n4 color "chocolate"
$n5 color "chocolate"
$n0 shape box ;
$n1 shape box ;
$n2 shape box ;
$n3 shape box ;
$n4 shape box ;
$n5 shape box ;
$ns at 0.0 "$n0 label SYS0"
$ns at 0.0 "$n1 label SYS1"
$ns at 0.0 "$n2 label SYS2"
$ns at 0.0 "$n3 label SYS3"
$ns at 0.0 "$n4 label SYS4"
$ns at 0.0 "$n5 label SYS5"set
nf [open goback.nam w]
$ns namtrace-all  $nf set f
[open goback.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n2 1Mb 20ms DropTail
```

```
$ns duplex-link-op $n0 $n2 orient right-down
$ns queue-limit $n0 $n2 5
$ns duplex-link $n1 $n2 1Mb 20ms DropTail
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link $n2 $n3 1Mb 20ms DropTail
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link $n3 $n4 1Mb 20ms DropTail
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link $n3 $n5 1Mb 20ms DropTail
$ns duplex-link-op $n3 $n5 orient right-down
Agent/TCP set_nam_tracevar_true
set tcp [new Agent/TCP]
$tcp set fid 1
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.05 "$ftp start"
$ns at 0.06 "$tcp set windowlnit 6"
$ns at 0.06 "$tcp set maxcwnd 6"
$ns at 0.25 "$ns queue-limit $n3 $n4 0"
$ns at 0.26 "$ns queue-limit $n3 $n4 10"
$ns at 0.305 "$tcp set windowlnit 4"
$ns at 0.305 "$tcp set maxcwnd 4"
$ns at 0.368 "$ns detach-agent $n1 $tcp ;
$ns detach-agent $n4 $sink"
$ns at 1.5 "finish"
$ns at 0.0 "$ns trace-annotate \"Goback N end\""
$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.01\""
$ns at 0.06 "$ns trace-annotate \"Send 6Packets from SYS1 to SYS4\""
$ns at 0.26 "$ns trace-annotate \"Error Occurs for 4th packet so not sent ack for the Packet\""
$ns at 0.30 "$ns trace-annotate \"Retransmit Packet_4 to 6\""
$ns at 1.0 "$ns trace-annotate \"FTP stops\""
proc finish {} {
global ns nf
$ns flush-trace
close $nf
puts "filtering..."
#exec tclsh../bin/namfilter.tcl goback.nam
#puts "running nam..."
exec nam goback.nam &
exit 0
}
$ns run
```

Output:



Selective Repeat ARQ is a specific instance of the Automatic Repeat-reQuest (ARQ) Protocol. It may beused as a protocol for the delivery and acknowledgement of message units, or it may be used as a protocol for the delivery of subdivided message sub-units. When used as the protocol for the delivery of messages, the sending process continues to send a number of frames specified by a window size even after a frame loss. Unlike Go-Back-N ARQ, the receiving process will continue to accept and acknowledge frames sentafter an initial error. The receiver process keeps track of the sequence number of the earliest frame it hasnot received, and sends that number with every ACK it sends. If a frame from the sender does not reach the receiver, the sender continues to send subsequent frames until it has emptied its window. The receiver continues to fill its receiving window with the subsequent frames, replying each time with an ACK containing the sequence number of the earliest missing frame. Once the sender has sent all the frames in its window, it re-sends the frame number given by the ACKs, and then continues where it left off. The size of the sending and receiving windows must be equal, and half the maximum sequence number (assuming that sequence numbers are numbered from 0 to n-1) to avoid miscommunication in all cases of packets being dropped. To understand this, consider the case when all ACKs are destroyed. If the receiving window is larger than half the maximum sequence number, some, possibly even all, of the packages that are resent after timeouts are duplicates that are not recognized as such. The sender moves its window forevery packet that is acknowledged.

Advantage over Go back-N:

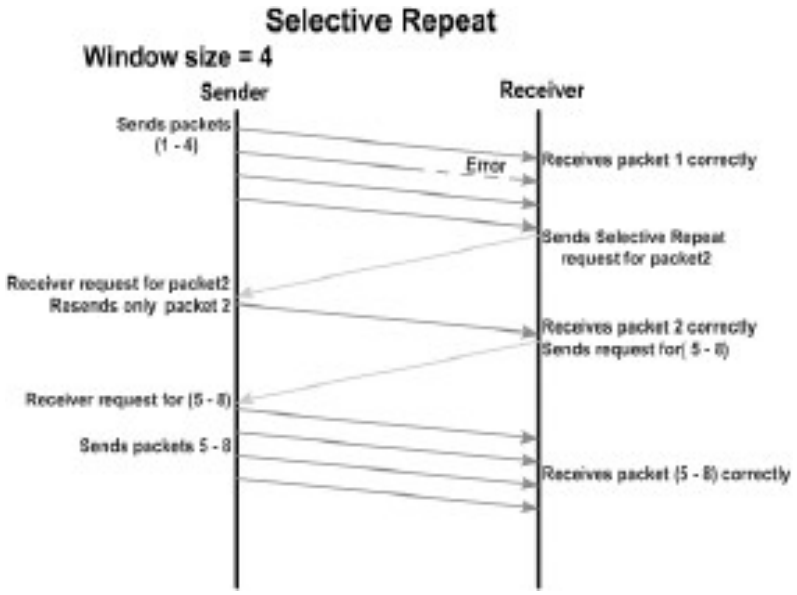
1. Fewer retransmissions.

Disadvantages:

1. More complexity at sender and receiver
2. Receiver may receive frames out of sequence

Algorithm:

1. The source node transmits the frames continuously.
2. Each frame in the buffer has a sequence number starting from 1 and increasing up to the windowsize.
3. The source node has a window i.e. a buffer to store the frames. This buffer size is the number offrames to be transmitted continuously.
4. The receiver has a buffer to store the received frames. The size of the buffer depends upon thewindow size defined by the protocol designer.
5. The size of the window depends according to the protocol designer.
6. The source node transmits frames continuously till the window size is exhausted. If any of the frames are received with error only those frames are requested for retransmission (with a negativeacknowledgement)
7. If all the frames are received without error, a cumulative positive acknowledgement is sent.
8. If there is an error in frame 3, an acknowledgement for the frame 2 is sent and then only Frame 3is retransmitted. Now the window slides to get the next frames to the window.
9. If acknowledgment is transmitted with error, all the frames of window are retransmitted. Else ordinary window sliding takes place. (* In implementation part, Acknowledgment error is not considered)
10. If all the frames transmitted are errorless the next transmission is carried out for the new window.
11. This concept of repeating the transmission for the error frames only is called Selective Repeat transmission flow control protocol.

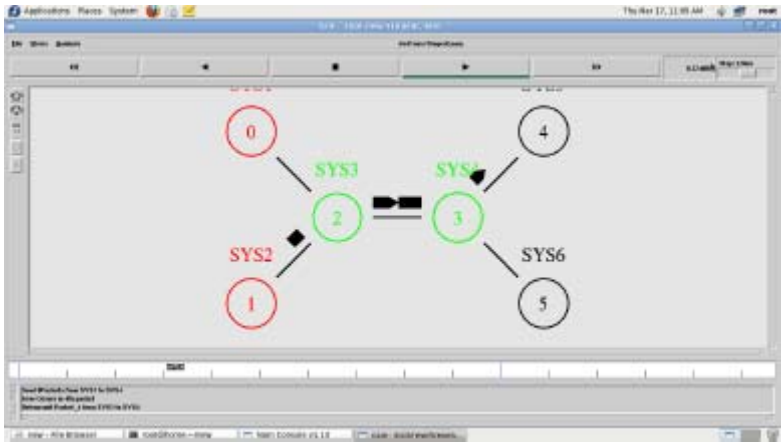


Program:

```
#send packets one by oneset
ns [new Simulator] set n0
[$ns node]
set n1 [$ns node]set
n2 [$ns node]set n3
[$ns node] set n4
[$ns node] set n5
[$ns node]
$n0 color "red"
$n1 color "red"
$n2 color "green"
$n3 color "green"
$n4 color "black"
$n5 color "black"
$n0 shape circle ;
$n1 shape circle ;
$n2 shape circle ;
$n3 shape circle ;
$n4 shape circle ;
$n5 shape circle ;
$ns at 0.0 "$n0 label SYS1"
$ns at 0.0 "$n1 label SYS2"
$ns at 0.0 "$n2 label SYS3"
$ns at 0.0 "$n3 label SYS4"
$ns at 0.0 "$n4 label SYS5"
$ns at 0.0 "$n5 label SYS6"set
nf [open Srepeat.nam w]
$ns namtrace-all $nf
set f [open Srepeat.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n2 orient right-down
$ns queue-limit $n0 $n2 5
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link $n3 $n5 1Mb 10ms DropTail
$ns duplex-link-op $n3 $n5 orient right-down
Agent/TCP set_nam_tracevar_true
set tcp [new Agent/TCP]
$tcp set fid 1
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.05 "$ftp start"
$ns at 0.06 "$tcp set windowlnit 8"
$ns at 0.06 "$tcp set maxcwnd 8"
$ns at 0.25 "$ns queue-limit $n3 $n4 0"
$ns at 0.26 "$ns queue-limit $n3 $n4 10"
$ns at 0.30 "$tcp set windowlnit 1"
$ns at 0.30 "$tcp set maxcwnd 1"
$ns at 0.30 "$ns queue-limit $n3 $n4 10"
$ns at 0.47 "$ns detach-agent $n1 $tcp;$ns detach-agent $n4 $sink"
$ns at 1.75 "finish"
$ns at 0.0 "$ns trace-annotate \"Select and repeat\""
$ns at 0.05 "$ns trace-annotate \"FTP starts at 0.01\""
$ns at 0.06 "$ns trace-annotate \"Send 8Packets from SYS1 to SYS4\""
$ns at 0.26 "$ns trace-annotate \"Error Occurs in 4th packet \""
$ns at 0.30 "$ns trace-annotate \"Retransmit Packet_4 from SYS1 to SYS4\""
$ns at 1.5 "$ns trace-annotate \"FTP stops\""
proc finish {} {
global ns nf
$ns flush-trace
close $nf
puts "filtering..."
#exec tclsh../bin/namfilter.tcl srepeat.nam
#puts "running nam..."
exec nam Srepeat.nam &
exit 0
}
$ns run
```

Output:



6. Result:

Thus the Go back-N and Selective Repeat protocols were simulated and studied.

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt afterperforming the experiment)

8. Conclusion:

(Students should write conclusion on their own)

9. Viva Questions:

- What is Go back-N protocol?
- What is Go back-N protocol ARQ?
- What is flow control?
- What is fixed size framing?
- What is pipelining?

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e)
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e)
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

1.

Experiment No.: 8
Implement TCP or UDP Sockets

1. **Aim:** Write a Programme to implement Socket Programming using TCP or UDP.

2. **Objectives:** To understand working of the transport layer protocol**.**

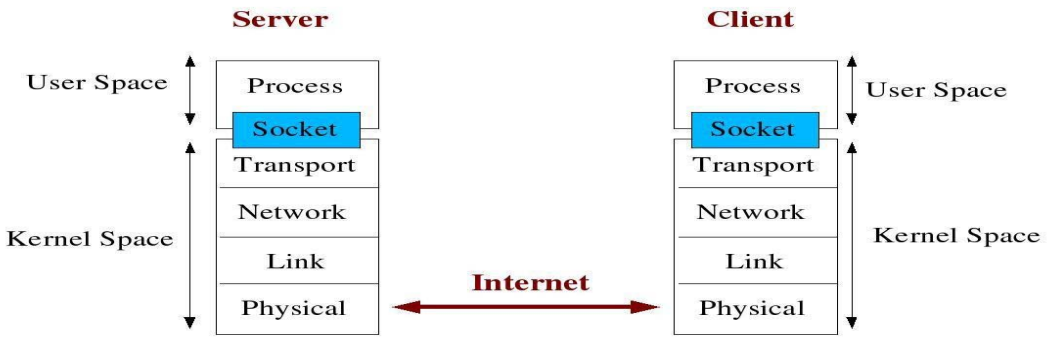
3. **Outcomes:** The learner will be able to
   - Demostrate the transport layer with the help of socket programming.
   - Recognize the need for sockets in life-long learning.

4. **Hardware/Software required:** JDK 1.8, Python 3.6

5. **Theory:**

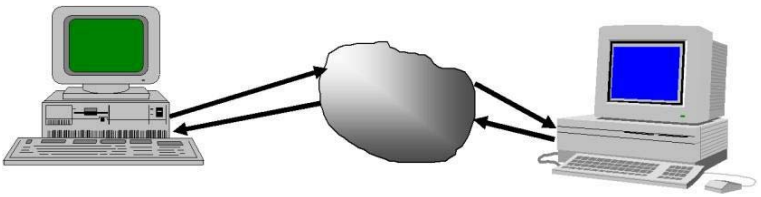
What is a socket?
● Socket: An interface between an application process and transport layer. The application process can send/receive messages to/from another application process (local

or remote) via a socket.



Client/Server Communication

At a basic level, network-based systems consist of a server, client, and a media for communication as shown in Figure 1. A computer running a program that makes request for services is called client machine. A computer running a program that offers requested services from one or more clients is called server machine. The media for communication can be wired or wireless network.



In computer network system the communication is takes place only through the message passing. Usually the communication takes place among client machine and server machine. The socket

programming is used to communicate between client and server.

Here, we are going to make one-way client and server communication. In this application, client sends a message to the server, server reads the message and prints it. Here, two classes are being used: Socket and ServerSocket.
The Socket class is used to communicate client and server. Through this class, we can read and write message. The ServerSocket class is used at server-side. The accept() method of ServerSocket class blocks the console until the client is connected. After the successful connection of client, it returns the instance of Socket at server-side.

#Socket class
A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.

#ServerSocket class
The ServerSocket class can be used to create a server socket. This object is used to establish communication with the clients.

Creating Server:
To create the server application, we need to create the instance of ServerSocket class. Here, we are using 6666 port number for the communication between the client and server. You may also choose any other port number. The accept() method waits for the client. If clients connects with the given port number, it returns an instance of Socket.
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes connection and waits for the client

Creating Client:
To create the client application, we need to create the instance of Socket class. Here, we need to pass the IP address or hostname of the Server and a port number. Here, we are using "localhost" because our server is running on same system.
Socket s=**new** Socket("localhost",6666);

Code:

MyServer.java file

```
import java.io.*; import java.net.*; public class MyServer
public static void main(String[] args){ try
{
ServerSocket ss=new ServerSocket(6666); Socket s=ss.accept();//establishes connection
DataInputStream dis=new DataInputStream(s.getInputStream()); String str=(String)dis.readUTF();
System.out.println("message= "+str); ss.close();
}
catch(Exception e){System.out.println(e);}
}
}
```

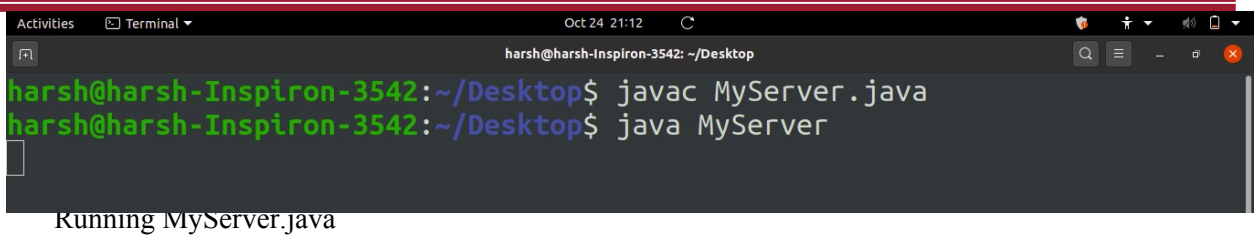
MyClient.java file

```
import java.io.*; import java.net.*; public class MyClient
{
public static void main(String[] args)
{
try
{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream()); dout.writeUTF("Hello Server");
dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}
```

Output:

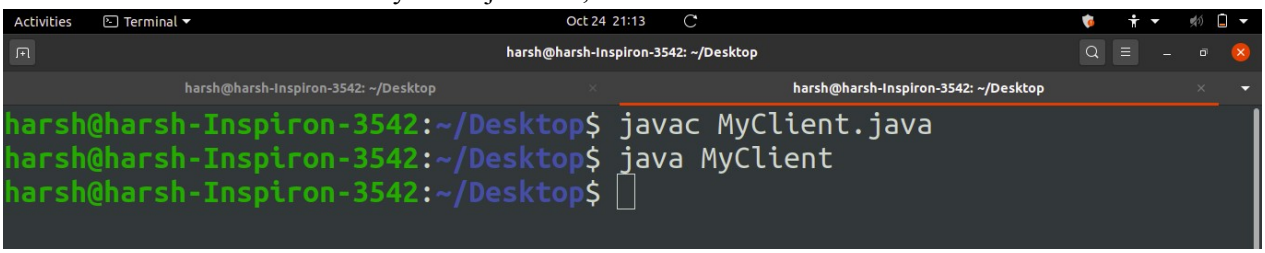
To execute this program open two command prompts and execute each program at each command prompt as displayed in the below figures.

First run Myserver.java file in terminal/cmd,



Running MyServer.java

Then in new terminal/cmd run MyClient.java file,



Running MyClient.java

As soon as you run MyClient program a message is sent to server and displayed in MyServer Terminal/CMD as shown below,

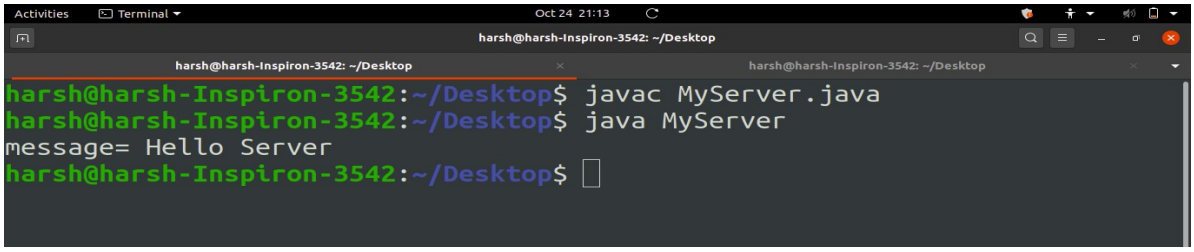


Fig. Message displayed in MyServer after running MyClient

Output
Client and server communication output window.

6. **Output Analysis:**
   (Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. **Additional Learning:**
   (Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. **Conclusion :**
   (Students should write conclusion on their own)

9. **Viva Questions:**

- State the promitives used for communication between client and server.
- Communication is always initiated by client. Jusify.
- Server has to be in passive mode. Justify.
- Explain the conept of socket in detail.

10. References:

a. A.S. Tanenbaum, “Computer Networks”, Pearson Education, (4e).
b. B.A. Forouzan, “Data Communications and Networking”, TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 9

File transfer Protocol (FTP)

1. **Aim**: Perform file transfer and access file using FTP.

2. **Objectives:** To make students understand the concept of and to access/transfer files using FTP Server.

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of FTP.
   - Use the commands for building networks.
   - Recognize the working of FTP in life-long learning.

4. **Hardware/Software required:** FTP Server, internet

5. **Theory:**

FTP (File Transfer Protocol) is a set of rules for transferring files between computers, usually connected through the Internet. If your computer supports FTP and is connected to the Internet, then you can access a large number of computer systems containing a large number of files. This facility was one of the main reasons for the initial popularity of the Internet. The program that supports the transfer of files using FTP is **ftp**. You can access information on virtually any topic using **ftp**. In addition, you can access a multitude of archives of software for most computer systems (DOS, UNIX, Macintosh, etc.).

FTP uses the client / server model that was discussed with **gopher**. Both client and server computers need to be running the **ftp** software. Unfortunately most of the non-Windows versions of **ftp** are not particularly easy to use. Ftp sites can also be connected to using Netscape and Internet Explorer, but only for downloading of files.

FTP installation steps

Install ProFTPD Server

root@www:~# apt-get -y install proftpd //Standalone
root@www:~# vi /etc/proftpd/proftpd.conf
line 11: turn off if not needed
UseIPv6 off
line 15: change to your hostname
ServerName "www.srv.world"
line 34: uncomment (specify root directory for chroot)
DefaultRoot ~
root@www:~# vi /etc/ftpusers
add users you prohibit FTP connection
test
root@www:~# systemctl restart proftpd

Install FTP Client.

root@client:~# apt-get -y install lftp
lftp [option] [hostname]

```
xerus@client:~$ lftp -u ubuntu www.srv.world    // lftp -u pc_user server_ip

Password:

# password of the user

lftp ubuntu@www.srv.world:~>

# show current directory on FTP server
lftp ubuntu@www.srv.world:~> pwd
ftp://ubuntu@www.srv.world

# show current directory on local server
lftp ubuntu@www.srv.world:~> !pwd
/home/ubuntu

# show files in current directory on FTP server
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_html
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py

# show files in current directory on local server
lftp ubuntu@www.srv.world:~> !ls -l
total 12
-rw-rw-r-- 1 ubuntu ubuntu 10 Jul 20 14:30 ubuntu.txt
-rw-rw-r-- 1 ubuntu ubuntu 10 Jul 20 14:59 test2.txt
-rw-rw-r-- 1 ubuntu ubuntu 10 Jul 20 14:59 test.txt

# change directory
lftp ubuntu@www.srv.world:~> cd public_html
lftp ubuntu@www.srv.world:~/public_html> pwd
ftp://ubuntu@www.srv.world/%2Fhome/ubuntu/public_html

# upload a file to FTP server
# "-a" means ascii mode ( default is binary mode )
lftp ubuntu@www.srv.world:~> put -a ubuntu.txt test.txt
22 bytes transferred
Total 2 files transferred
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_html
-rw-r--r--    1 1000     1000           10 Jul 20 17:01 ubuntu.txt
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:01 test.txt

# upload some files to FTP server
```

```
lftp ubuntu@www.srv.world:~> mput -a test.txt test2.txt

22                                  bytes                                  transferred
Total                          2                        files                        transferred
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_html
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt

# download a file from FTP server

# "-a" means ascii mode ( default is binary mode )

lftp ubuntu@www.srv.world:~> get -a test.py

416 bytes transferred
# download some files from FTP server

lftp ubuntu@www.srv.world:~> mget -a test.txt test2.txt

20                                  bytes                                  transferred
Total 2 files transferred
# create a directory in current directory on FTP Server

lftp ubuntu@www.srv.world:~> mkdir testdir
mkdir                          ok,                          `testdir'                          created
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_html
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt
drwxr-xr-x    2 1000     1000            6 Jul 20 17:16 testdir
226 Directory send OK.

# delete a direcroty in current directory on FTP Server

lftp                ubuntu@www.srv.world:~>                rmdir                testdir
rmdir ok, `testdir' removed

lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_html
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt
```

```
# delete a file in current directory on FTP Server

lftp ubuntu@www.srv.world:~> rm test2.txt
rm ok, `test2.txt' removed
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x  2 1000  1000    23 Jul 19 01:33 public_html
-rw-r--r--  1 1000  1000   399 Jul 20 16:32 test.py
-rw-r--r--  1 1000  1000    10 Jul 20 17:06 test.txt

# delete some files in current directory on FTP Server

lftp ubuntu@www.srv.world:~> mrm ubuntu.txt test.txt

rm ok, 2 files removed
lftp ubuntu@www.srv.world:~> ls
drwxr-xr-x  2 1000  1000    23 Jul 19 01:33 public_html

# execute commands with "![command]"

lftp ubuntu@www.srv.world:~> !cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
...
ubuntu:x:1001:1001::/home/ubuntu:/bin/bash

# exit

lftp ubuntu@www.srv.world:~> quit

221 Goodbye.
```

6. **Output Analysis:**
   (Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. **Additional Learning:**
   (Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. **Conclusion :**
   (Students should write conclusion on their own)

9. Viva Questions:

- State the steps to install FTP client and server.
- FTP operates at which layer of OSI model.

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e).
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 10

Perform Remote login using Telnet server/ SSH server

1. **Aim**: Perform Remote login using Telnet server/ SSH server using Linux Command Prompt.

2. **Objectives:** To make students understand the concept of and perform remote login using TELNET Server.

3. **Outcomes:** The learner will be able to
   - Analyze the functioning of Telnet.
   - Use the Telnet for building networks and performing remote login.
   - Recognize the need for remote login.

4. **Hardware/Software required:** Telnet Server

5. **Theory:**

Remote Login

A **shell** is a piece of software which provides a user interface to the computer's operating system, so that we may enter commands. There are many ways of accessing the shell on a remote computer from a local computer, including two programs called **telnet** and **ssh**. The remote computer would run **telnet** and/or **ssh *server*** software, and the local computer would run a **telnet** and/or **ssh *client*** software. We interact with the telnet ***client*** software on our local computer and it sends requests to the corresponding ***server*** software on the remote computer. Most multiuser computers run one of these server programs so that users can connect and use the machine.

As an aside, you have probably used and have on your home computer several other pieces of client software. You use an email client program (Outlook Express, Mozilla Thunderbird, Netscape's email client) to access an email server to retrieve and send mail. Your web browser contains a web client which makes requests for html and other documents from a web server.

The shell access client software (the one running on your local computer) gives you a terminal ***window*** that along with a keyboard behaves the same as the old monitors used in the original terminals.

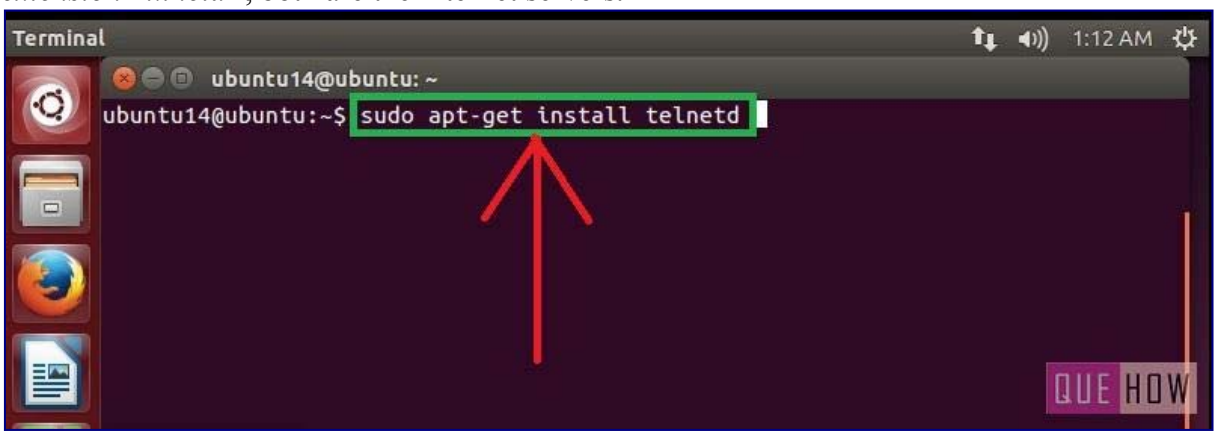
Shell access, such as through the **ssh** or **telnet** program is a simple and ***low-bandwidth*** way of connecting to a remote computer. Low-bandwidth means that only a small amount of information is transferred per second. It doesn't require a fast connection to the remote computer, and a client program can be run from any computer that is connected to a network, even if both computer are using different Graphical User Interface's (Windows, MAC OS, X-Windows etc.) or not using a GUI at all.

Telnet is a protocol that allows you to connect to remote computers (called hosts) over a TCP/IP network (such as the internet). Using telnet client software on your computer, you can make a connection to a telnet server (i.e., the remote host). Once your telnet client establishes a connection to the remote host, your client becomes a virtual terminal, allowing you to

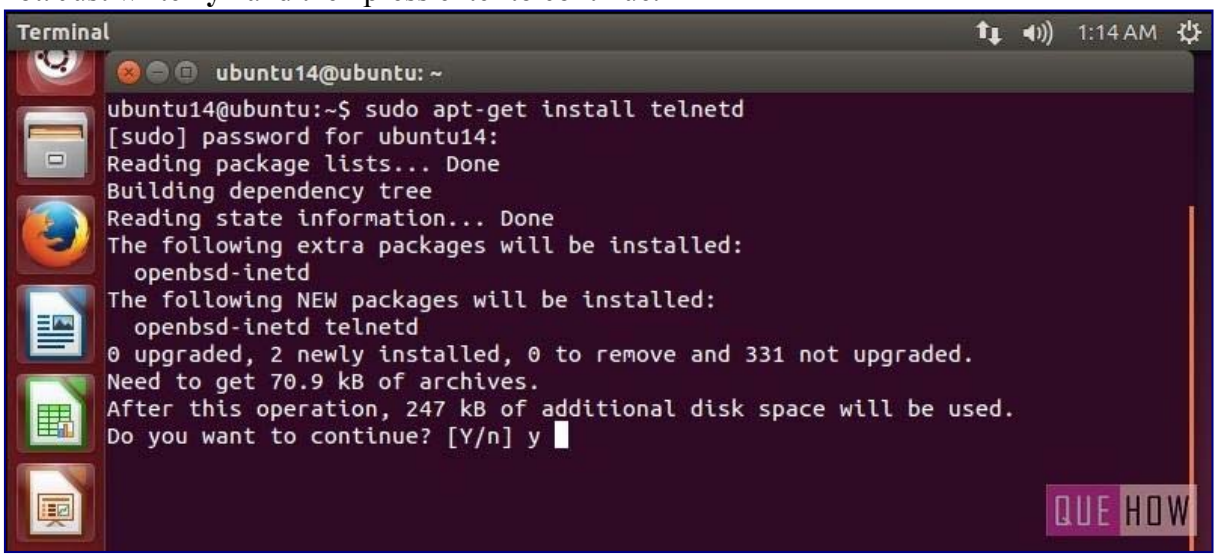
communicate with the remote host from your computer. In most cases, you'll need to log into the remote host, which requires that you have an account on that system. Occasionally, you can log in as guest or public without having an account.

Steps to Install and Use Telnet in Ubuntu

Step 1: Firstly, open the **"Terminal"** window by pressing **"Ctrl + Alt + T"**. In the figure, you may see **"$"** that signifies that you are not logged in as a root user.
So, I'll write **"sudo apt-get install telnetd"** and press enter. If you are a root user, then you don't need to write sudo in Ubuntu. **"telnetd"** is a daemon that gets invoked by *"inetd" or its extension "xinetd"*, both are the internet servers.

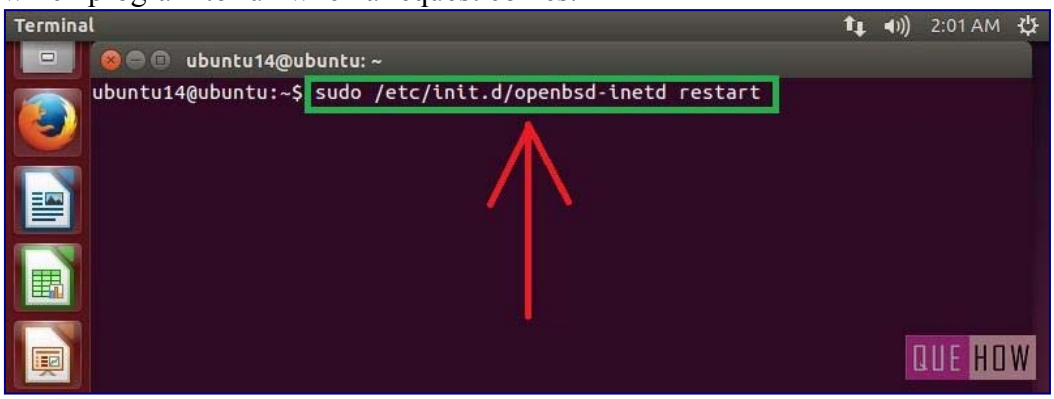


Step 2: Then you are asked to enter the user password and then press enter. Processing will start as soon as you press enter. After this, I have noticed a line **"274 KB additional disk space will be used"** on the terminal screen.
You may also observe some sort of a message like this and then you'll be asked to continue or not. Just write **"y"** and then press enter to continue.

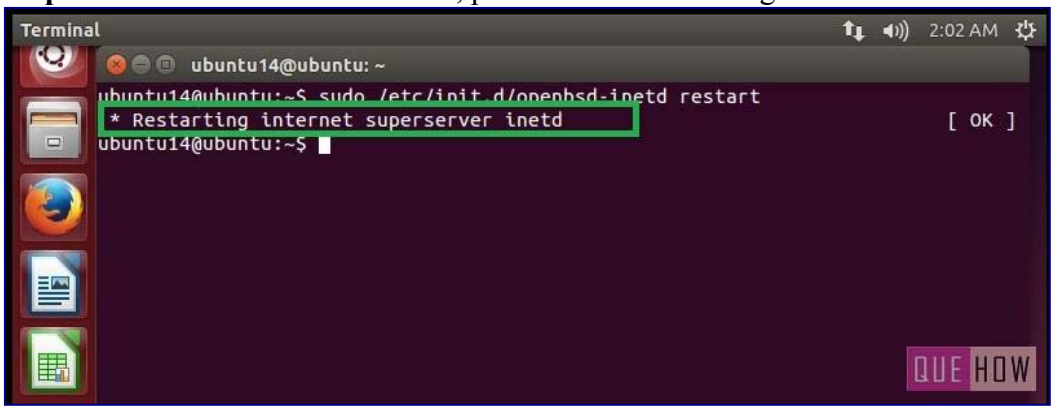


Step 3: Now when you are done with it, **restart "inetd"**. Type **"sudo /etc/init.d.open-bsd-inetd restart"**.

"inetd" is daemon used for *dealing with incoming network* and it is responsible for deciding which program to run when a request comes.



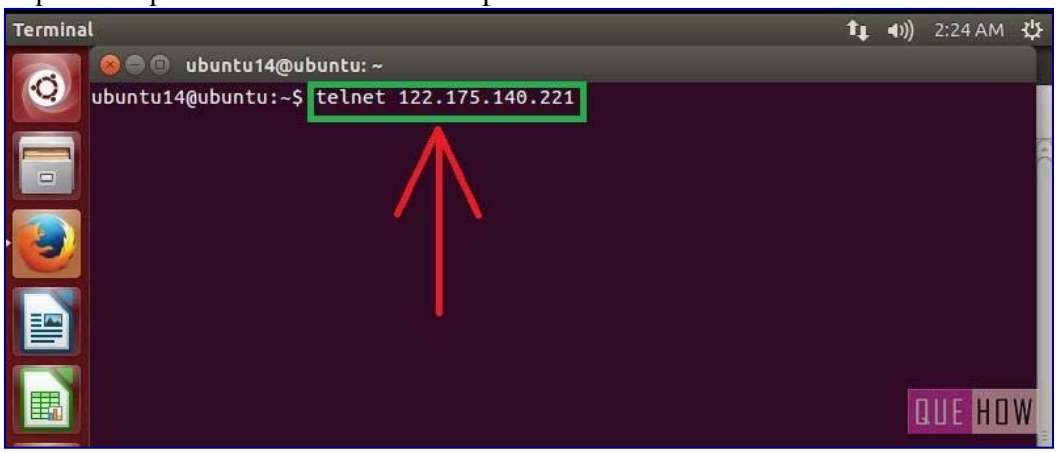
Step 4: To ensure "inetd" is started, press enter after writing the above command.

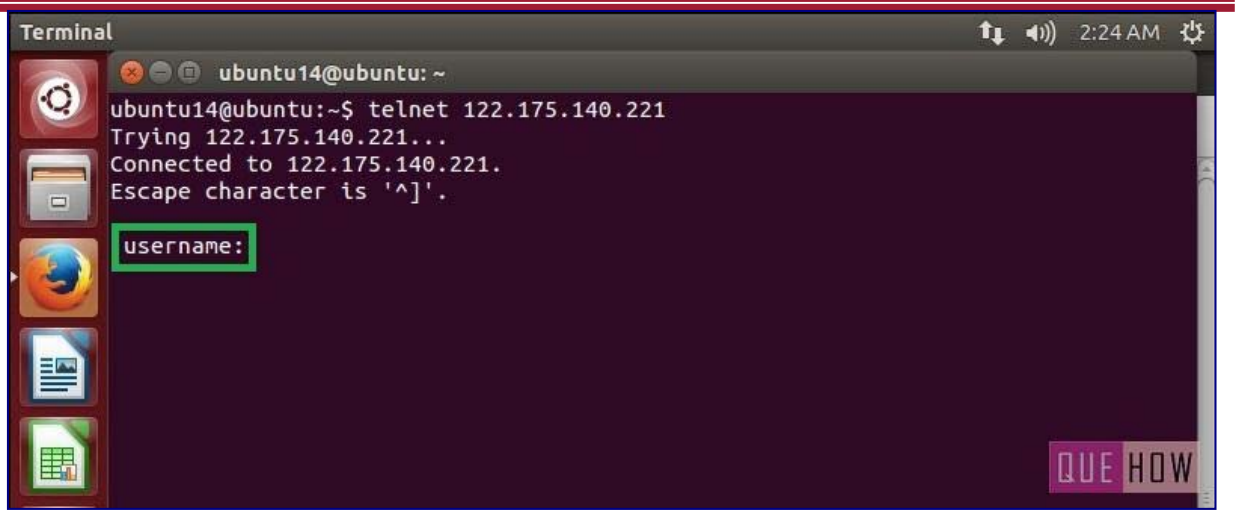


To connect with any remote client:

Step 5: Just type:**"telnet hostipaddress"**. For an example: "telnet 122.175.140.221" and press enter.

Step 6: Then you"ll see, it is connected to **"host ip address"**. For security reasons, you are required to provide "username" and "password" as well.





6. **Output Analysis:**
   (Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. **Additional Learning:**
   (Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. **Conclusion :**
   (Students should write conclusion on their own)

9. **Viva Questions:**
   - State the steps to install Telnet.
   - Telnet operates at which layer of OSI model.

10. **References:**
    a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e)
    b. B.A. Forouzan, "Data Communications and Networking", TMH (5e)

Experiment No.: 11

Enabling IP forwarding using IPtables in Linux

Aim: Perform packet filtering by enabling IP forwarding using IPtables in Linux.

a. Set up multiple IP addresses on a single LAN.
b. Using nestat and route commands of Linux, do the following:
- View current routing table
- Add and delete routes
- Change default gateway

Theory:

First, let us find the IP address of the network card. In my Ubuntu 15.10 server, I use only one network card.

Run the following command to find out the IP address:

```
sudo ip addr
```

Sample output:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default

 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

 inet 127.0.0.1/8 scope host lo

 valid_lft forever preferred_lft forever

 inet6 ::1/128 scope host

 valid_lft forever preferred_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000

 link/ether 08:00:27:2a:03:4b brd ff:ff:ff:ff:ff:ff

 inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3

 valid_lft forever preferred_lft forever

 inet6 fe80::a00:27ff:fe2a:34e/64 scope link
```

```
valid_lft forever preferred_lft forever
```

Or

```
sudo ifconfig
```

Sample output:

```
enp0s3 Link encap:Ethernet HWaddr 08:00:27:2a:03:4b

inet addr:192.168.1.103 Bcast:192.168.1.255 Mask:255.255.255.0

inet6 addr: fe80::a00:27ff:fe2a:34e/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:186 errors:0 dropped:0 overruns:0 frame:0

TX packets:70 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:21872 (21.8 KB) TX bytes:9666 (9.6 KB)



lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:65536 Metric:1

RX packets:217 errors:0 dropped:0 overruns:0 frame:0

TX packets:217 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:0

RX bytes:38793 (38.7 KB) TX bytes:38793 (38.7 KB)
```

As you see in the above output, my network card name is **enp0s3**, and its IP address is **192.168.1.103**.

Now let us add an additional IP address, for example **192.168.1.104**, to the Interface card.

Open your Terminal and run the following command to add additional IP.

```
sudo ip addr add 192.168.1.104/24 dev enp0s3
```

Now, let us check if the IP is added using command:

```
sudo ip address show enp0s3
```

Sample output:

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000

 link/ether 08:00:27:2a:03:4e brd ff:ff:ff:ff:ff:ff

 inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3

 valid_lft forever preferred_lft forever

 inet 192.168.1.104/24 scope global secondary enp0s3

 valid_lft forever preferred_lft forever

 inet6 fe80::a00:27ff:fe2a:34e/64 scope link

 valid_lft forever preferred_lft forever
```

Similarly, you can add as many IP addresses as you want.

Let us ping the IP address to verify it.

```
sudo ping 192.168.1.104
```

Sample output:

```
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.

64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.901 ms

64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.571 ms

64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.521 ms

64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=0.524 ms
```

To check the routing table

Command: *nestat -rn*

```
$ netstat -rn

Kernel IP routing table

Destination  Gateway      Genmask        Flags MSS Window irtt Iface

0.0.0.0      192.168.0.1  0.0.0.0        UG    0   0      0    wlan0

192.168.0.0  0.0.0.0      255.255.255.0  U     0   0      0    wlan0
```

Adding route

```
sudo route add -net 192.168.3.0 gw 192.168.1.1 netmask 255.255.255.0 dev eth0
```

Deleting route

```
sudo route del -net 192.168.3.0 gw 192.168.1.1 netmask 255.255.255.0 dev eth0
```

A quick way to add default route

```
route add default gw 192.168.1.1
```

A quick way to delete defualt route

```
route del default gw 192.168.1.1
```

6. Output Analysis:

(Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. Conclusion :

(Students should write conclusion on their own)

9. Viva Questions:

- State the header format of DLL, IP, TCP and UDP and put values of one the packet that you have captured using Wireshark
- Explain how layers are interrelated.

10. References:

a. A.S. Tanenbaum, "Computer Networks", Pearson Education, (4e).
b. B.A. Forouzan, "Data Communications and Networking", TMH (5e).
c. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).

Experiment No.: 12

Network Packet Analyzer

Understand the operation of TCP/IP layers

Aim: Use **Wireshark** to understand the operation of TCP/IP layers :

- Ethernet Layer : Frame header, Frame size etc.
- Data Link Layer : MAC address, ARP (IP and MAC address binding)
- Network Layer : IP Packet (header, fragmentation), ICMP (Query and Echo)
- Transport Layer: TCP Ports, TCP handshake segments etc.
- Application Layer: DHCP, FTP, HTTP header formats

1. **Objectives:** To explore the network packet analyzer Wireshark on different TCP/IP layer. To introduce concepts and fundamentals of data communication and computer networks.

2. **Outcomes:** The learner will be able to
   - Analyze the functioning of various protocols.
   - Use of protocols in networks.
   - Recognize the need for networking TCP/IP protocols.

3. **Hardware/Software required: Wireshark**

4. **Theory:**

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Wireshark is very similar to tcpdump, but has a graphical front-end, plus some integrated sorting and filtering options.

Wireshark allows the user to put the network interfaces that support promiscuous mode into that mode, in order to see all traffic visible on that interface, not just traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic. However, when capturing with a packet analyzer in promiscuous mode on a port on a network switch, not all of the traffic traveling through the switch will necessarily be sent to the port on which the capture is being done, so capturing in promiscuous mode will not necessarily be sufficient to see all traffic on the network. Port mirroring or various network taps extend capture to any point on net; simple passive taps are extremely resistant to malware tampering.

Wireshark Installation steps:

1. Enter admin mode by following command. If not entered in admin mode then packets will not get captured in wireshark.
   sudo su

2. Command to install wireshark on ubuntu

sudo apt-get install wireshark

3. Double click on the wireshark icon. We get an open window as given below.

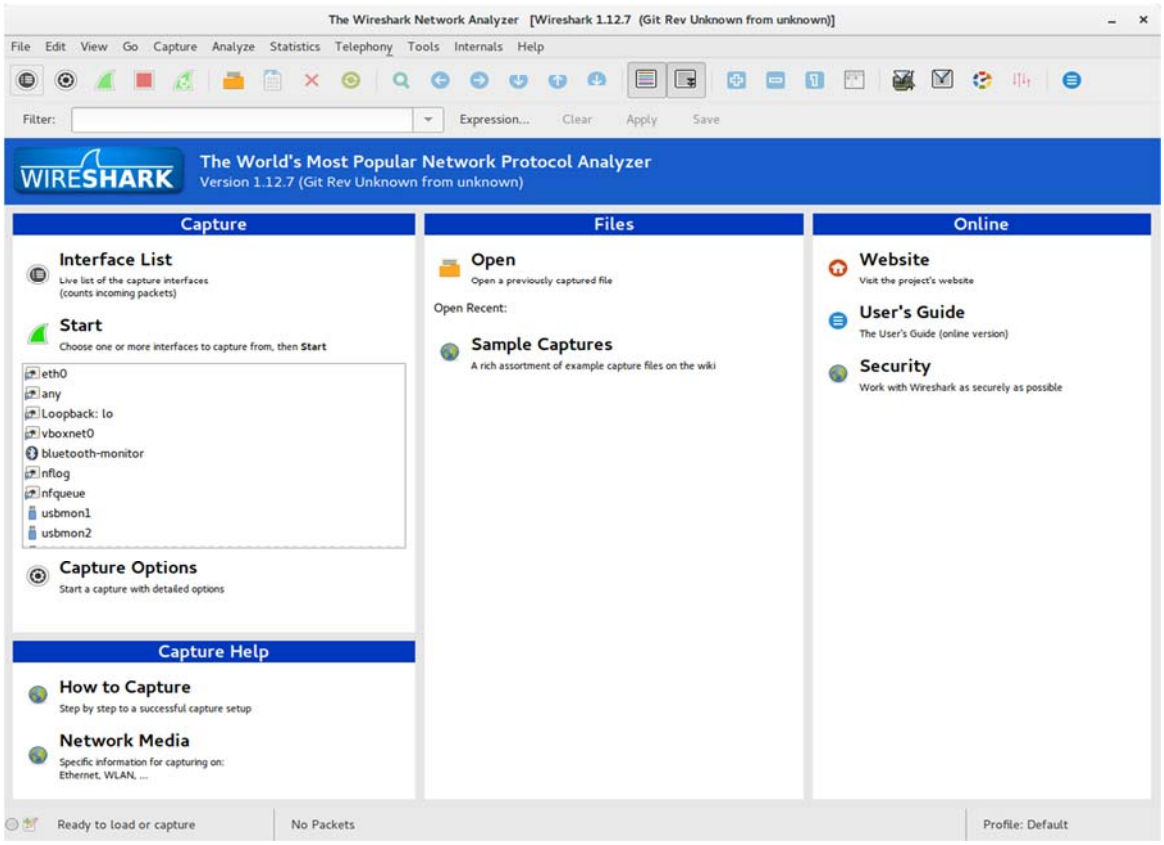


Figure 1.1: Wireshark initial showing interfaces (sudo mode)

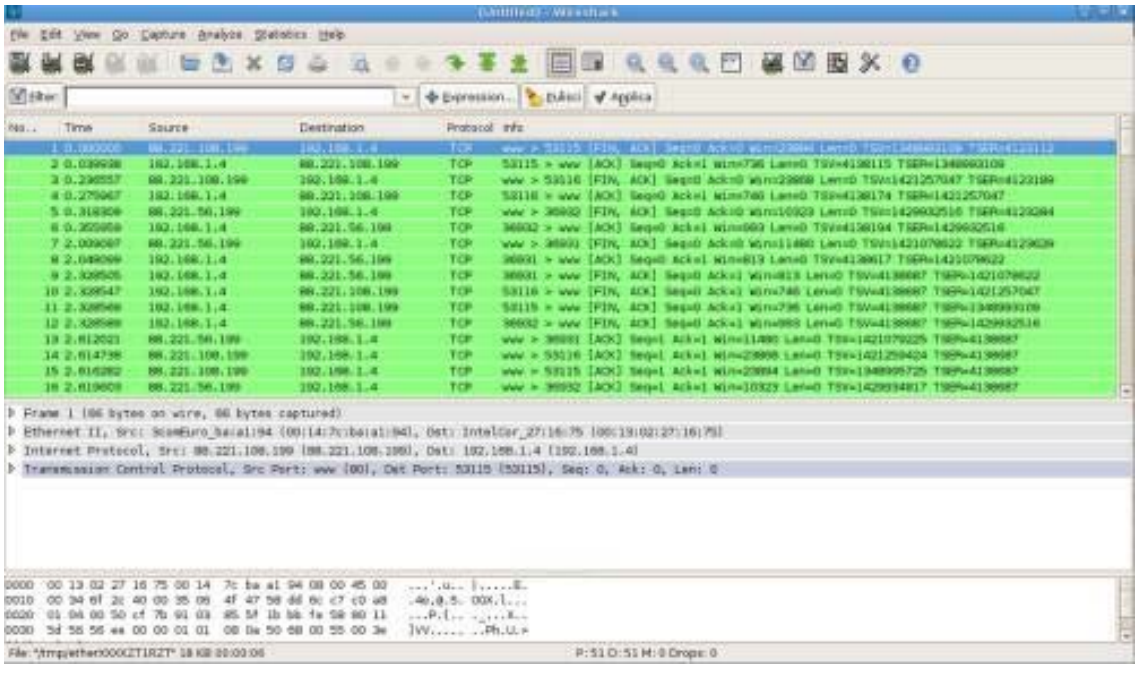


Figure 1.2. An example of a Wireshark capture.

Frame 4: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
Arrival Time: Jul 17, 2008 03:50:25.136434000 Eastern Daylight Time
Epoch Time: 1216281025.136434000 seconds
[Time delta from previous captured frame: 0.000188000 seconds]
[Time delta from previous displayed frame: 0.000188000 seconds]
[Time since reference or first frame: 0.000265000 seconds]
Frame Number: 4
Frame Length: 122 bytes (976 bits)
Capture Length: 122 bytes (976 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ip:tcp:mysql]
[Coloring Rule Name: TCP]
[Coloring Rule String: tcp]

Figure 2. The summary before the protocols in a Wireshark packet. Information about the packet characteristic.

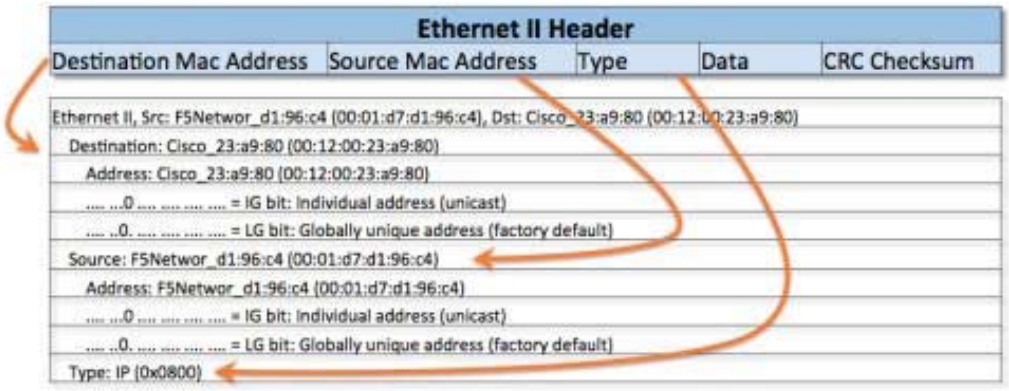


Figure 3. Ethernet II (Layer 2) header along with the Wireshark

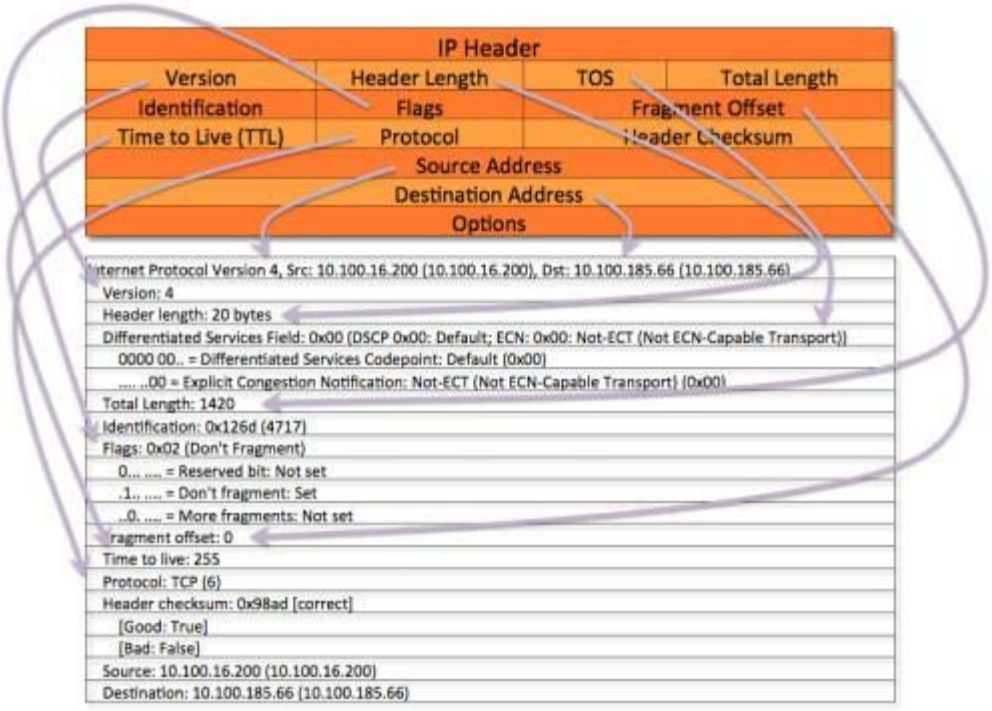


Figure 4. IP Header (Layer-3)

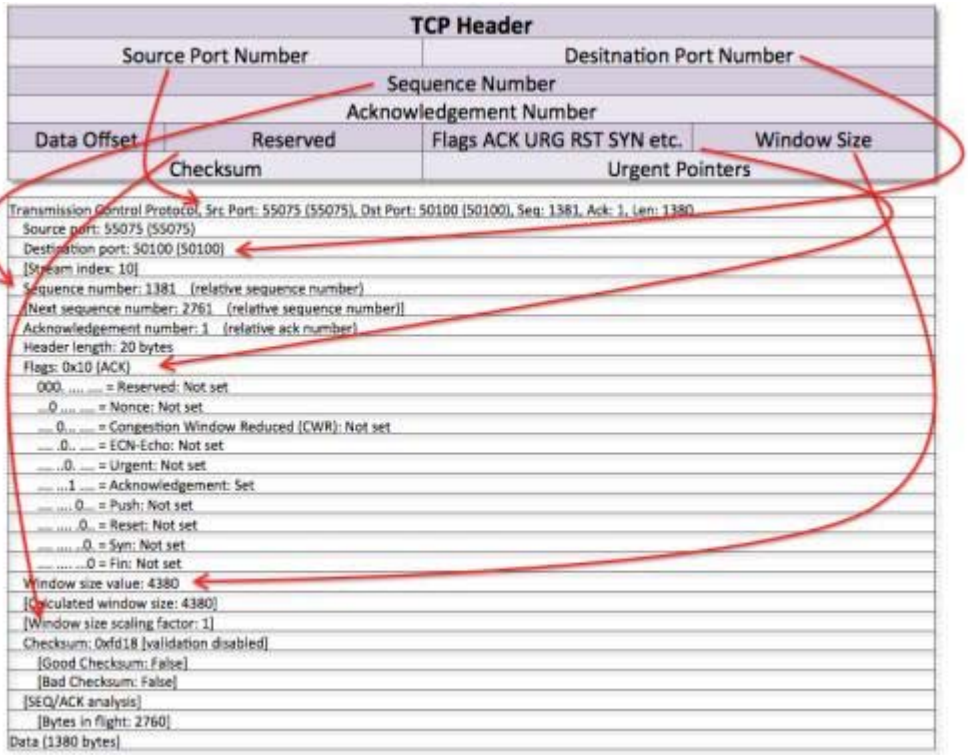
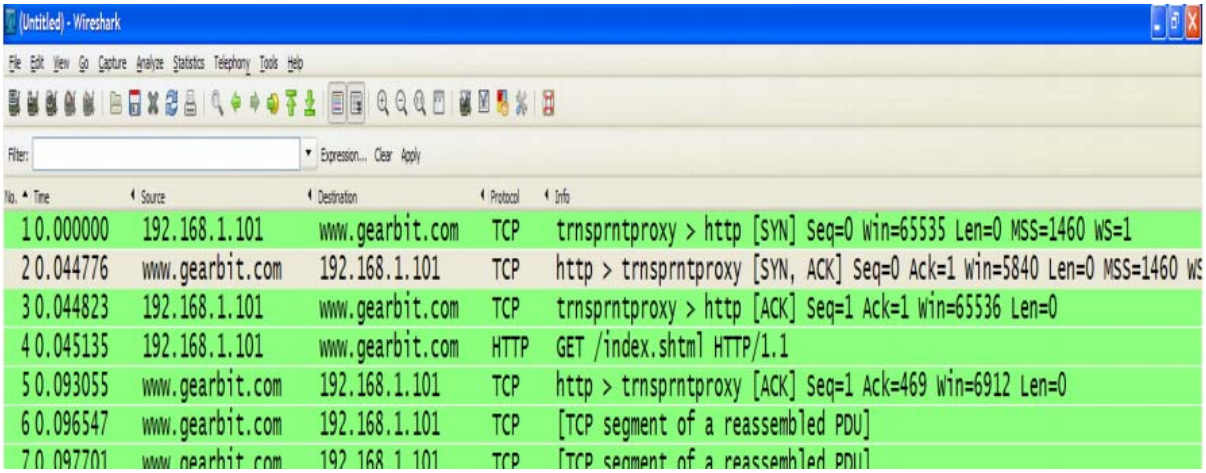


Figure 5. TCP headers.

TCP Three-way Handshake

The delta value between frames 1 and 2 can be used as a TCP transport connect baseline value. Other important information gathered from this handshake: • Window Size • SACK • Maximum Segment Size • Window Scale Option value



6. Output Analysis:

(Students should write output analysis based on the working of different topology and different networking devices used in simulation. Specify each scenario explicitly with output analysis)

7. Additional Learning:

(Students should write additional learning on their own based on what additionally they learnt after performing the experiment)

8. Conclusion :

(Students should write conclusion on their own)

9. Viva Questions:

- State the header format of DLL, IP, TCP and UDP and put values of one the packet that you have captured using Wireshark
- Explain how layers are interrelated.

10. References:

a. A.S. Tanenbaum, “Computer Networks”, Pearson Education, (4e)
b. B.A. Forouzan, “Data Communications and Networking”, TMH (5e)
d. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE)