



# Computer Networks Lab Manual

Computer Network (University of Mumbai)

Mahatma Education Society's

**Pillai HOC College of Engineering and Technology, Rasayani**

Department of Computer Engineering

## Experiment List

---

**Subject:** Computer Networks

**Semester:** V

Serial No.	Name of Experiment	Page No.
1.	Study of Networking Commands and Network Configuration Files.	2
2.	Setup a network and configure IP addressing, subnetting, Masking. (Eg. CISCO <b>Packet Tracer</b> )	8
3.	Study and Build a simple network topology to configure it for static routing protocol using packet tracer.	21
4.	Understand the operation of packet sniffer tools like wireshark	29
5.	i)Installation & Configuration of NS2 in Linux Environment. ii) Implement DVR(Distance Vector Routing).	33
6.	i) Setting up multiple IP Addresses on a single LAN. ii) Using netstat and route commands.	43
7.	A client-server application using socket programming in Java	47
8.	Study and implementation of CRC algorithm	51
9.	Perform File Transfer and Access using FTP	58
10.	Perform Remote Login using Telnet server.	61
<b>H/W Requirements</b>		RAM 512 MB, Printer, Cartridges
<b>S/W Requirements</b>		JDK 1.3, Cisco Packet Ttracer 6.3, NS-2.35

## EXPERIMENT NO: 1

Name of the Student:-

Roll No. \_\_\_\_\_

Subject:-

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with  
Date & Marks :-

--

**TITLE: Study of Networking Commands and Network Configuration Files.**

**Aim:** Study of Networking Commands and Network Configuration Files.

- i) Ping.
- ii) Traceroute.
- iii) Netstat.
- iv) Nslookup.
- v) ARP.
- vi) Telnet

### Theory:

#### PING

Sends ICMP ECHO\_REQUEST packets to network hosts. 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.

The ping utility uses the ICMP protocol's mandatory ECHO\_REQUEST datagram to elicit an ICMP ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of "pad" bytes used to fill out the packet.

#### Syntax

```
ping [-LRUbdnqrVvaAB] [-c count] [-m mark] [-i interval] [-l preload]
[-p pattern] [-s packetsize] [-t ttl] [-w deadline] [-F flowlabel]
[-I interface] [-M hint] [-N nioption] [-Q tos] [-S sndbuf]
[-T timestamp option] [-W timeout] [hop ...] destination
```

#### Options

## Examples

### 1. Ping the host to see if its alive

#### \$ ping google.com

PING google.com (74.125.200.102) 56(84) bytes of data.

64 bytes from plus.google.com (74.125.200.102): icmp\_req=1 ttl=128 time=172 ms

64 bytes from plus.google.com (74.125.200.102): icmp\_req=2 ttl=128 time=164 ms

64 bytes from plus.google.com (74.125.200.102): icmp\_req=4 ttl=128 time=165 ms

--- google.com ping statistics ---

4 packets transmitted, 3 received, 25% packet loss, time 3013ms

rtt min/avg/max/mdev = 164.618/167.289/172.010/3.364 ms

### 2. Send N packets and stop

#### \$ ping -c 4 google.com

PING google.com (74.125.135.100) 56(84) bytes of data.

64 bytes from plus.google.com (74.125.135.100): icmp\_req=1 ttl=128 time=251 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=2 ttl=128 time=180 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=3 ttl=128 time=179 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=4 ttl=128 time=179 ms

--- google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 179.569/197.734/251.433/31.005 ms

### 3. Timeout

The following example will ping for 5 seconds. i.e ping command will exit after 5 seconds irrespective of how many packets are sent or received.

#### \$ ping -w 5 localhost

## TRACEROUTE

Traceroute prints the route that packets take to a network host. The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route your packets follow (or finding a gateway that's discarding your packets) can be difficult. Traceroute utilizes the IP protocol "time to live" field and attempts to elicit an ICMP TIME\_EXCEEDED response from each gateway along the path to some host. The only mandatory parameter is the destination host name or IP number. The default probe

datagram length is 40 bytes, but this may be increased by specifying a packet size (in bytes) after the destination host name. Traceroute attempts to trace the route an IP packet would follow to some internet host by launching probe packets with a small ttl (time to live) then listening for an ICMP "time exceeded" reply from a gateway. It starts its probes with a ttl of one and increases this by one until it gets an ICMP "port unreachable" (or TCP reset), which means we got to the "host", or hit a max (which defaults to 30 hops). Three probes (by default) are sent at each ttl setting and a line is printed showing the ttl, address of the gateway and round trip time of each probe. The address can be followed by additional information when requested. If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a 5.0 seconds (default), an "\*" (asterisk) is printed for that probe. After the trip time, some additional annotation can be printed: !H, !N, or !P (host, network or protocol unreachable), !S (source route failed), !F (fragmentation needed), !X (communication administratively prohibited), !V (host precedence violation), !C (precedence cutoff in effect), or !<num> (ICMP unreachable code <num>). If almost all the probes result in some kind of unreachable, traceroute will give up and exit.

You don't want the destination host to process the UDP probe packets, so the destination port is set to an unlikely value (you can change it with the -p flag). There is no such a problem for ICMP or TCP tracerouting (for TCP we use half-open technique, which prevents our probes to be seen by applications on the destination host). In the modern network environment the traditional traceroute methods can not be always applicable, because of widespread use of firewalls. Such firewalls filter the "unlikely" UDP ports, or even ICMP echoes. To solve this, some additional tracerouting methods are implemented (including tcp); see LIST OF AVAILABLE METHODS below. Such methods try to use particular protocol and source/destination port, in order to bypass firewalls (to be seen by firewalls just as a start of allowed type of a network session).

## Syntax

```
traceroute [-46dFITUnreAV] [-f first_ttl] [-g gate,...] [-i device]
[-m max_ttl] [-p port] [-s src_addr] [-q nqueries]
[-N squeries] [-t tos] [-l flow_label] [-w waittime]
[-z sendwait] [-UL] [-D] [-P proto] [--sport=port] [-M method]
[-O mod_options] [--mtu] [--back] host [packet_len]
```

## Examples

### 1. Ping the host to see if its alive

#### # traceroute 4.2.2.2

```
traceroute to 4.2.2.2 (4.2.2.2), 30 hops max, 60 byte packets
1 192.168.50.1 (192.168.50.1) 0.217 ms 0.624 ms 0.133 ms
2 227.18.106.27.mysipl.com (27.106.18.227) 2.343 ms 1.910 ms 1.799 ms
3 221-231-119-111.mysipl.com (111.119.231.221) 4.334 ms 4.001 ms 5.619 ms
4 10.0.0.5 (10.0.0.5) 5.386 ms 6.490 ms 6.224 ms
5 gi0-0-0.dgw1.bom2.pacific.net.in (203.123.129.25) 7.798 ms 7.614 ms 7.378 ms
6 115.113.165.49.static-mumbai.vsnl.net.in (115.113.165.49) 10.852 ms 5.389 ms 4.322 ms
7 ix-0-100.tcore1.MLV-Mumbai.as6453.net (180.87.38.5) 5.836 ms 5.590 ms 5.503 ms
8 if-9-5.tcore1.WYN-Marseille.as6453.net (80.231.217.17) 216.909 ms 198.864 ms 201.737 ms
9 if-2-2.tcore2.WYN-Marseille.as6453.net (80.231.217.2) 203.305 ms 203.141 ms 202.888 ms
```

### 2. Disable IP address and host name mapping

## **\$ traceroute google.com -n**

traceroute to google.com (173.194.36.7), 30 hops max, 60 byte packets

```
1 220.224.141.129 109.352 ms 109.280 ms 109.248 ms
2 115.255.239.65 131.633 ms 131.598 ms 131.573 ms
3 124.124.251.245 131.554 ms 131.529 ms 131.502 ms
4 115.255.239.45 131.478 ms 131.464 ms 199.741 ms
5 72.14.212.118 199.674 ms 199.637 ms 199.603 ms
6 209.85.241.52 199.578 ms 199.549 ms 209.838 ms
7 209.85.241.187 199.488 ms 177.264 ms 177.196 ms
8 173.194.36.7 177.159 ms 187.463 ms 187.434 ms
```

## **3. Configure Response Wait Time**

The -w option expects a value which the utility will take as the response time to wait for. In this example, the wait time is 0.1 seconds and the traceroute utility was unable to wait for any response and it printed all the \*'s.

## **\$ traceroute google.com -w 0.1**

traceroute to google.com (74.125.236.101), 30 hops max, 60 byte packets

```
1 ***
2 ***
3 ***
..
29 * * *
30 * * *
```

## **NSLOOKUP :**

The nslookup command is used to query internet name servers interactively for information. nslookup, which stands for "name server lookup", is a useful tool for finding out information about a named domain. By default, nslookup will translate a domain name to an IP address (or vice versa). For instance, to find out what the IP address of microsoft.com is, you could run the command:

## **nslookup microsoft.com**

...and you would receive a response like this:

```
Server: 8.8.8.8
Address: 8.8.8.8#53
Non-authoritative answer:
Name: microsoft.com
Address: 134.170.185.46
Name: microsoft.com
Address: 134.170.188.221
```

Here, 8.8.8.8 is the address of our system's Domain Name Server. This is the server our system is configured to use to translate domain names into IP addresses. "#53" indicates that we are communicating with it on port 53, which is the standard port number domain name servers use to accept queries.

Below this, we have our lookup information for microsoft.com. Our name server returned two entries, 134.170.185.46 and 134.170.188.221. This indicates that microsoft.com uses a round robin setup to distribute server load. When you access microsoft.com, you may be directed to either of these servers and your packets will be routed to the correct destination.

You can see that we have received a "Non-authoritative answer" to our query. An answer is

"authoritative" only if our DNS has the complete zone file information for the domain in question. More often, our DNS will have a cache of information representing the last authoritative answer it received when it made a similar query; this information is passed on to you, but the server qualifies it as "non-authoritative": the information was recently received from an authoritative source, but the DNS server is not itself that authority.

Syntax

nslookup [-option] [name | -] [server]

## NETSTAT:

netstat – Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.netstat (network statistics) is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. netstat is available on all Unix-like Operating Systems and also available on Windows OS as well. It is very useful in terms of network troubleshooting and performance measurement. netstat is one of the most basic network service debugging tools, telling you what ports are open and whether any programs are listening on ports.

Syntax :

netstat [-a] [-n] [-v]

netstat [-g | -m | -p | -s | -f address\_family ] [-n] [-P protocol]

netstat [-i ] [ -I interface ] [ interval ]

netstat -r [-a] [-n] [-v ]

netstat -M [-n] [-s ]

netstat -D [ -I interface ]

## Options

-a	Show the state of all sockets and all routing table entries; normally, sockets used by server processes are not shown and only interface, host, network, and default routes are shown.
-n	Show network addresses as numbers. netstat normally displays addresses as symbols. This option may be used with any of the display formats.
-v	Verbose. Show additional information for the sockets and the routing table.
-g	Show the multicast group memberships for all interfaces.
-m	Show the STREAMS statistics.
-p	Show the address resolution (ARP) tables.
-s	Show per-protocol statistics. When used with the -M option, show multicast routing statistics instead.
-i	Show the state of the interfaces that are used for TCP/IP traffic.
-r	Show the routing tables.
-M	Show the multicast routing tables. When used with the -s option, show multicast routing statistics instead.
-d	Show the state of all interfaces that are under Dynamic Host Configuration Protocol (DHCP) control.
-D	Show the status of DHCP configured interfaces.
-P protocol	Limit display of statistics or state of all sockets to those applicable to protocol.

## ARP:

arp – manipulate the system Address Resolution Protocol (ARP) cache.“arp” manipulates the kernel’s ARP cache in various ways. The primary options are clearing an address mapping

entry and manually setting up one. For debugging purposes, the arp program also allows a complete dump of the ARP cache.

Syntax :

```
arp [-evn] [-H type] [-i if] -a [hostname]
arp [-v] [-i if] -d hostname [pub]
arp [-v] [-H type] [-i if] -s hostname hw_addr [temp]
arp [-v] [-H type] [-i if] -s hostname hw_addr [netmask nm] pub
arp [-v] [-H type] [-i if] -Ds hostname ifa [netmask nm] pub
arp [-vnD] [-H type] [-i if] -f [filename]
```

### **TELNET:**

The telnet program is a user interface to the TELNET protocol. The telnet command is used for interactive communication with another host using the TELNET protocol. It begins in command mode, where it prints a telnet command prompt ("telnet>"). If telnet is invoked with a host argument, it performs an open command implicitly

**Syntax**

```
telnet [-468ELadr] [-S tos] [-b address] [-e escapechar] [-l user]
[-n tracefile] [host [port]]
```

### **Networking Configuration Files :**

The primary network configuration files are as follows:

- 1. /etc/hosts :** The main purpose of this file is to resolve hostnames that cannot be resolved any other way. It can also be used to resolve hostnames on small networks with no DNS server. Regardless of the type of network a computer is using, this file should contain a line specifying the IP address of the loopback device (127.0.0.1) as localhost. localdomain.
- 2. /etc/resolv.conf :** This file specifies the IP addresses of the DNS servers and the search domain.
- 3. /etc/sysconfig/network :** This file specifies routing and host information for all network interfaces.
- 4. /etc/host.conf:** This file lists resolver options.
- 5. /etc/nss.conf :** The nss.conf is used for Name Switch Service configuration. It defines order of name resolution options.
- 6. /etc/sysconf/networking :** This file holds network configuration files managed by redhat-config-network.
- 7. /etc/sysctl.conf :** It is used for IP forwarding configuration.
- 8. /etc/services :** It lists available network Services such as FTP and Telnet, and the ports they use.
- 9. /etc/protocols :** It lists protocols available on your system.

### **Conclusion:**

---

---

### **Frequently Asked Questions:**

1. What is ping command?
-



2. What is ICMP protocol?

---

---

---

3. Explain traceroute command.

---

---

---

4. What is nslookup command?

---

---

---

5. What is netstat command and how to find listening ports?

---

---

---

6. What are the uses of ARP and RARP?

---

---

---

7. Explain any four network configuration files.

---

---

---

## Experiment No. 02

Name of the Student:-

\_\_\_\_\_

Roll No. \_\_\_\_\_ Subject:- Computer Network

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date:

**Aim:** Setup a network and configure IP addressing, subnetting, Masking. (Eg. CISCO Packet Tracer)

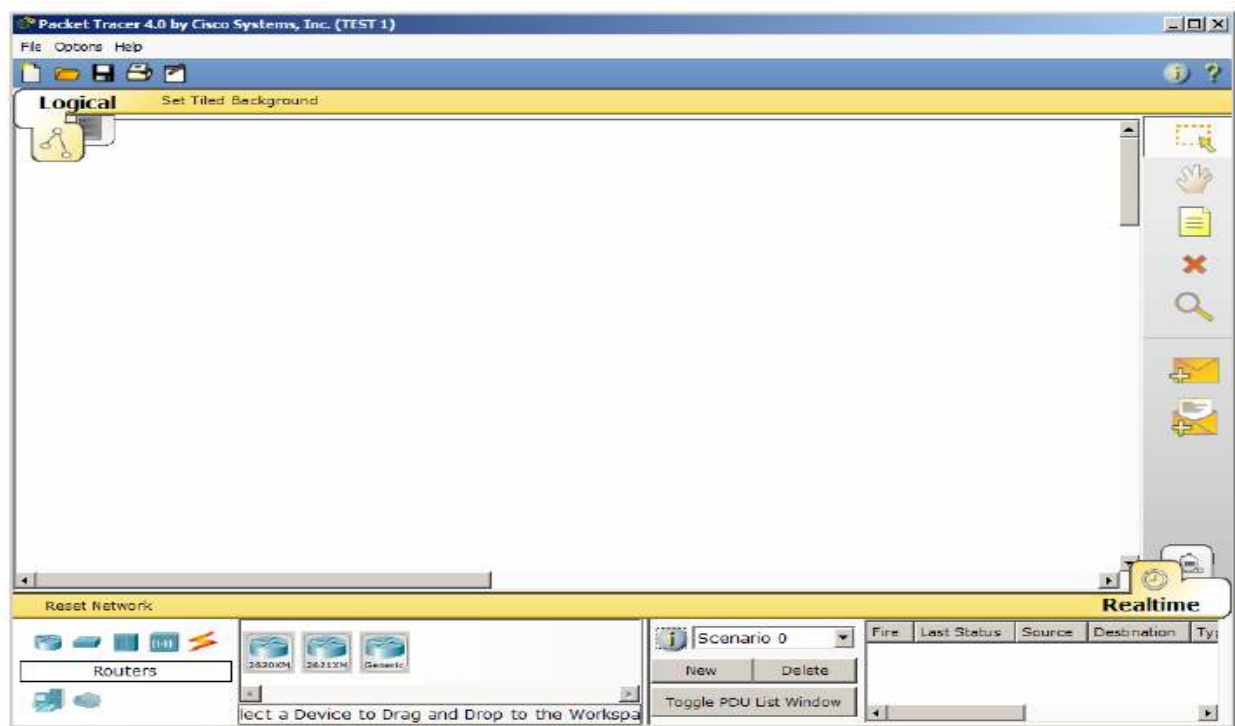
**Apparatus (Software):** Cisco Packet tracer Software

### **Theory:**

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

### **Introduction to the Packet Tracer Interface using a star Topology**

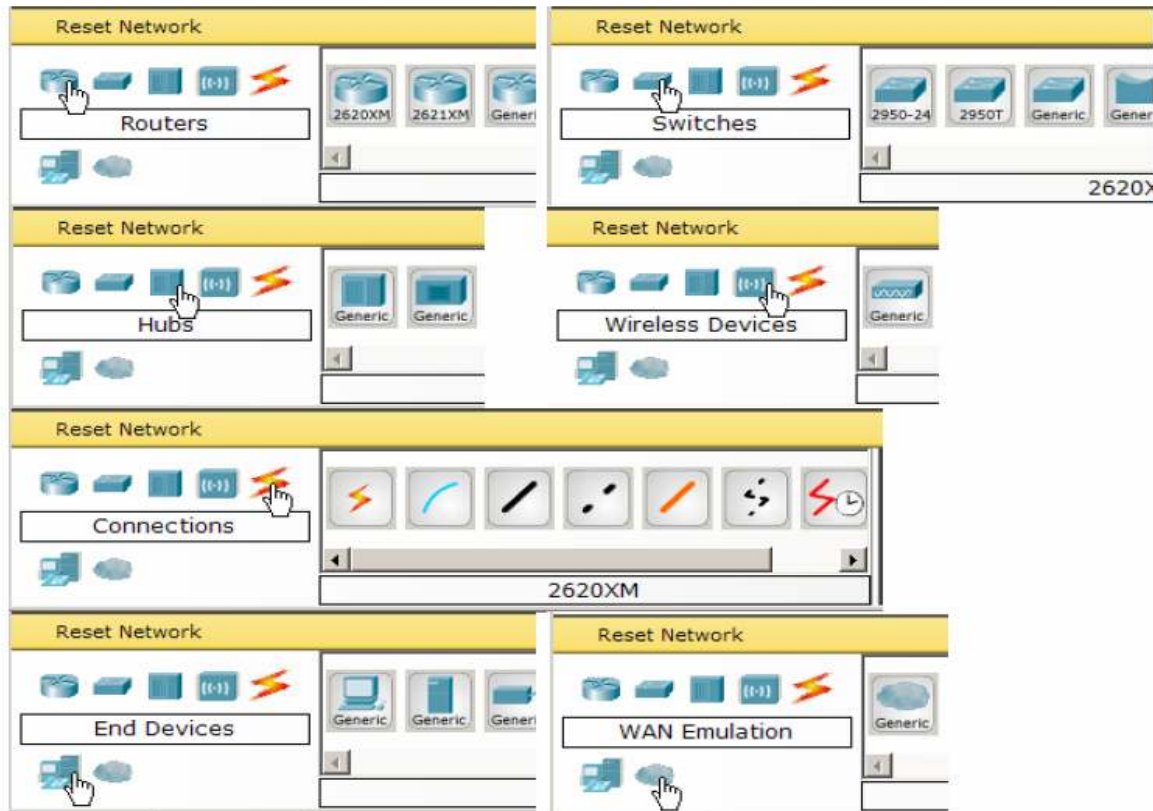
Step 1: Start Packet Tracer and Entering Simulation Mode



## Step 2: Choosing Devices and Connections

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

Single click on each group of devices and connections to display the various choices.



## Step 3: Building the Topology – Adding Hosts

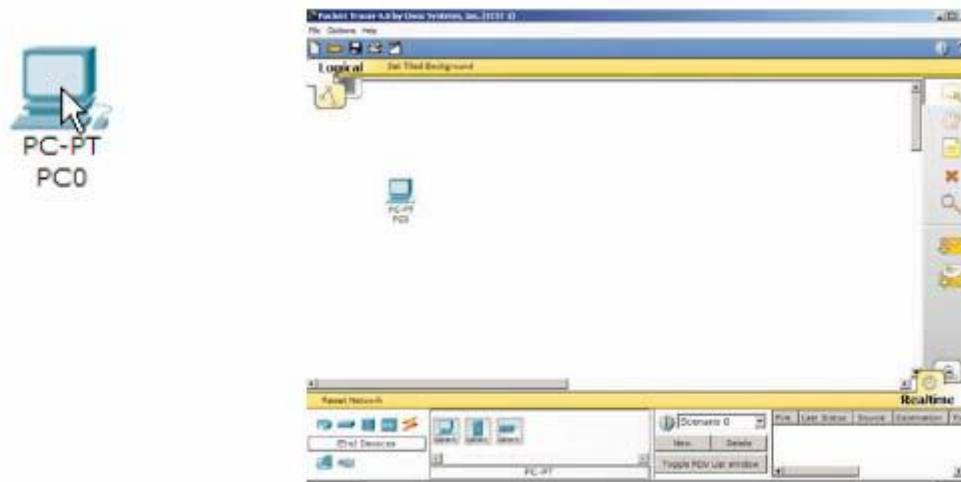
Single click on the End Devices.



Single click on the **Generic** host.



Move the cursor into topology area. You will notice it turns into a plus “+” sign. Single click in the topology area and it copies the device.



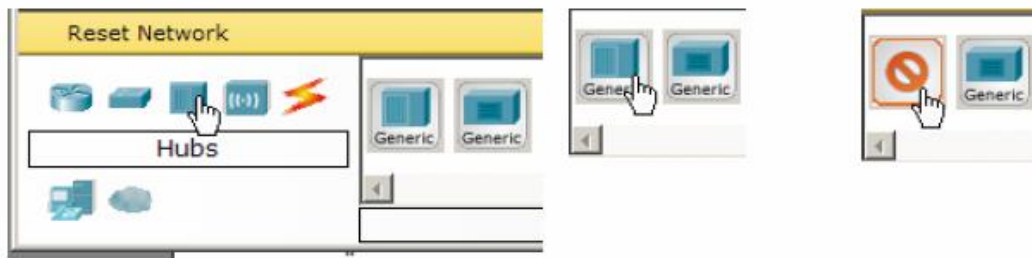
Add three more hosts.



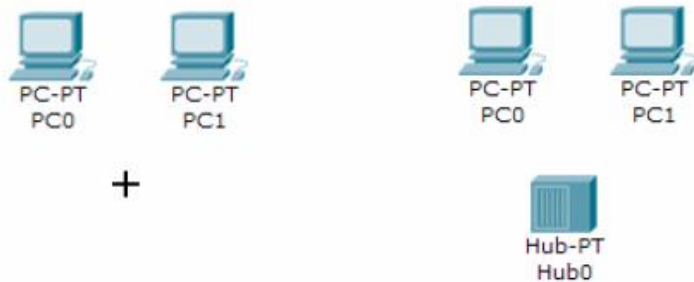
## Step 4: Building the Topology – Connecting the Hosts to Hubs and Switches

### Adding a Hub

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



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



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



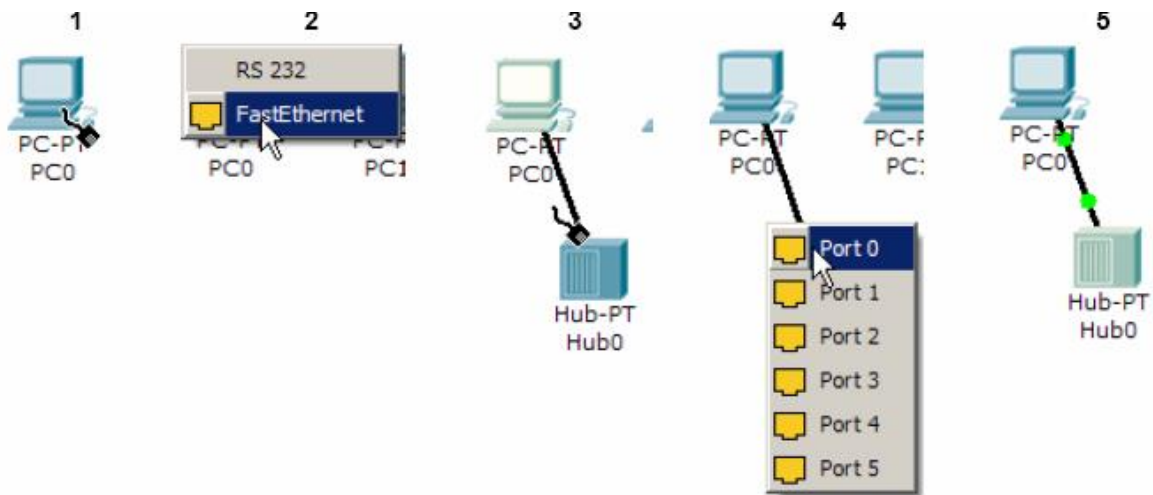
Click once on the **Copper Straight-through** cable.



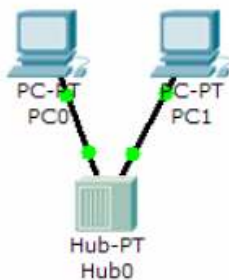
Perform the following steps to connect PC0 to Hub0:

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

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

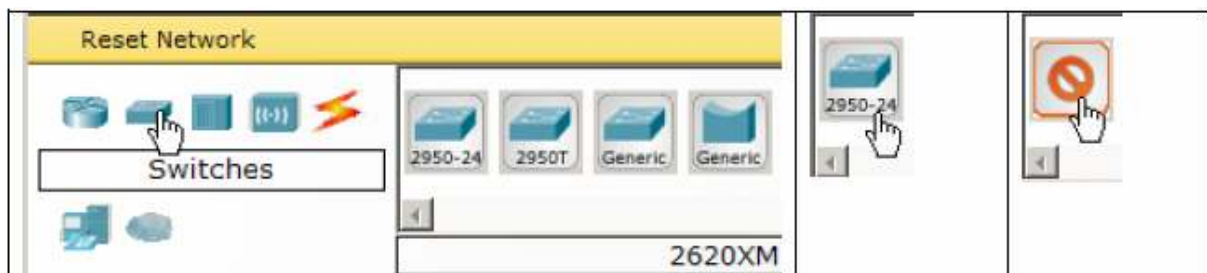


Repeat the steps above for **PC1** connecting it to **Port 1** on **Hub0**. (The actual hub port you choose does not matter.)

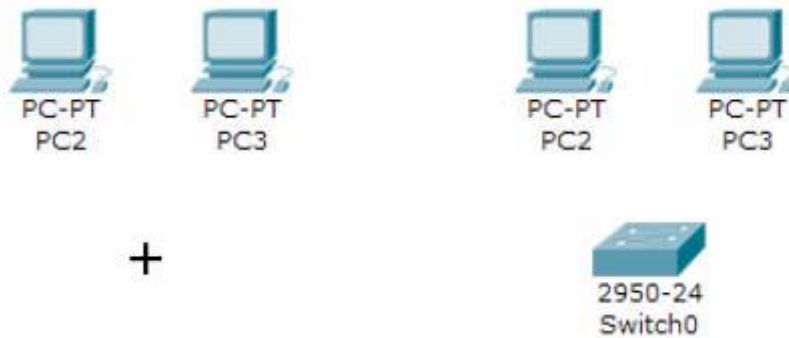


## Adding a Switch

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



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



Connect PC2 to Hub0 by first choosing Connections.



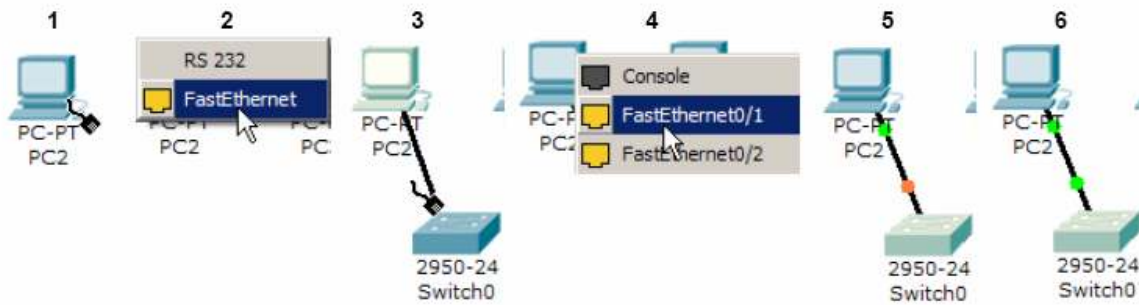
Click once on the Copper Straight-through cable.



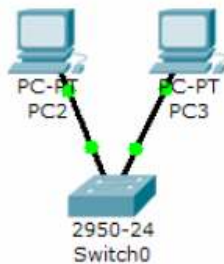
Perform the following steps to connect PC2 to Switch0:

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

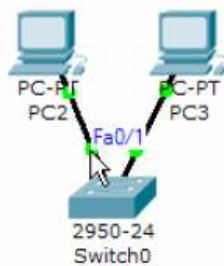




Repeat the steps above for **PC3** connecting it to **Port 3** on **Switch0** on port **FastEthernet0/2**. (The actual switch port you choose does not matter.)



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

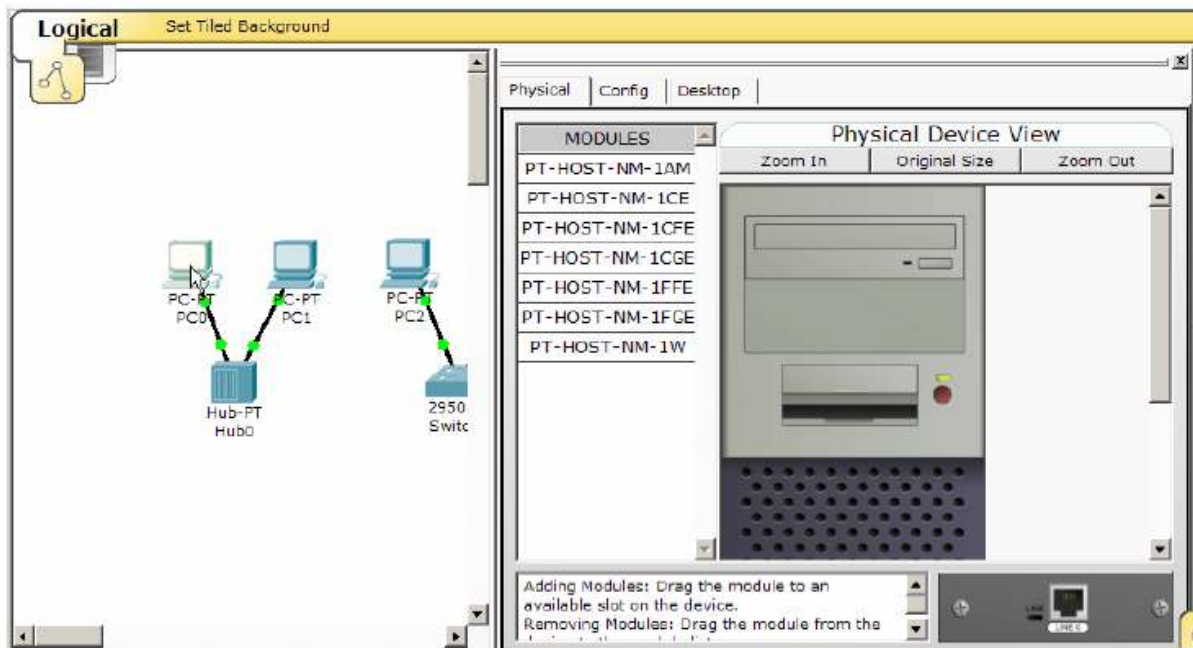


## Step 5: Configuring IP Addresses and Subnet Masks on the Hosts

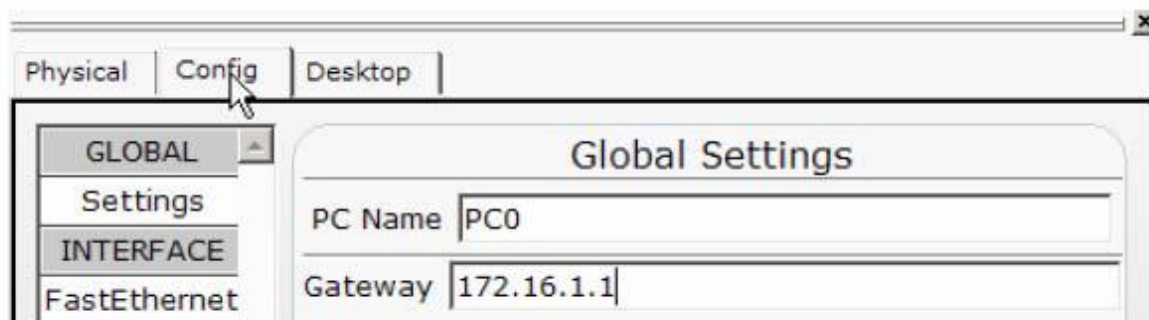
Before we can communicate between the hosts we need to configure IP Addresses and Subnet Masks on the devices.

Click once on PC0.

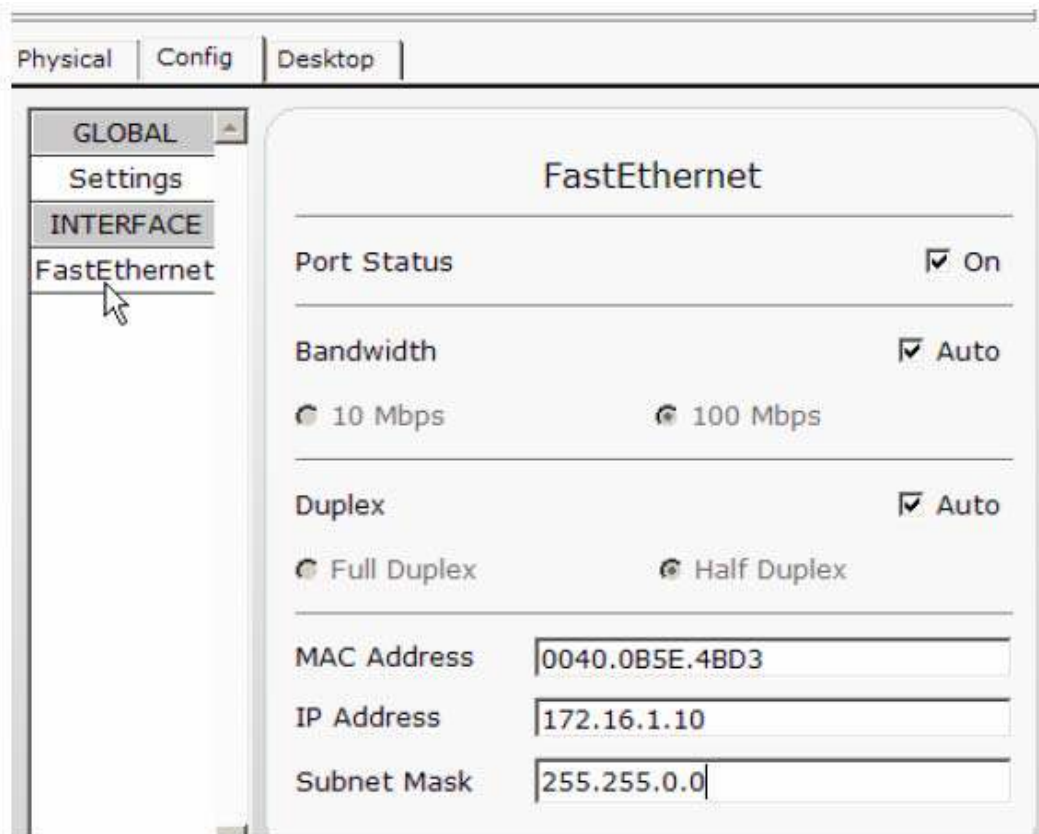




**Choose the Config tab.** It is here that you can change the name of PC0. It is also here where you would enter a Gateway IP Address, also known as the default gateway. We will discuss this later, but this would be the IP address of the local router. If you want, you can enter the IP Address 172.16.1.1,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



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



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

### **Bandwidth - Auto**

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

### **Duplex - Auto**

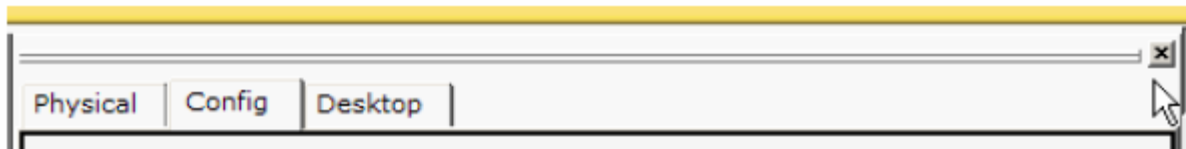
Hub: If the host is connected to a hub, then the Ethernet NIC on the host will choose Half Duplex.

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

Ethernet NIC on the host will choose Half Duplex. (Full Duplex is a much more efficient option.)

The information is automatically saved when entered.

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



Repeat these steps for the other hosts. Use the information below for IP Addresses and Subnet Masks. Host	IP Address	Subnet Mask
PC0	172.16.1.10	255.255.0.0
PC1	172.16.1.11	255.255.0.0
PC2	172.16.1.12	255.255.0.0
PC3	172.16.1.13	255.255.0.0



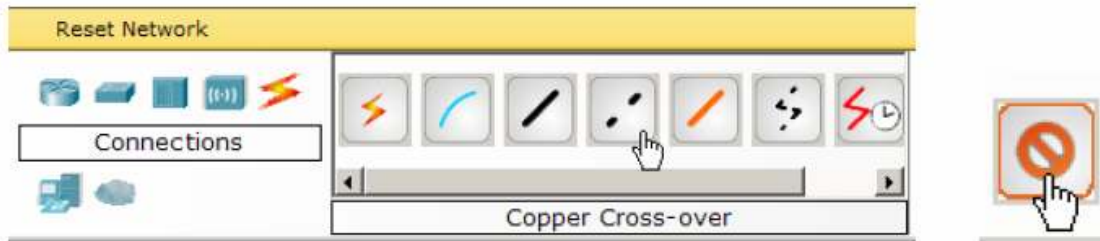
## Deleting a Device or Link

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

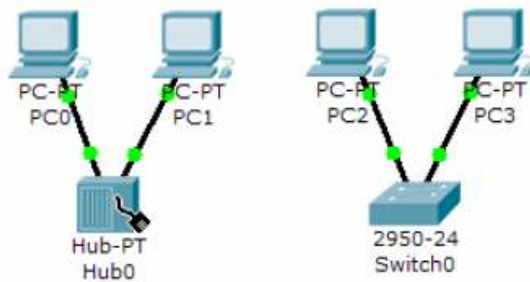


## Step 6: Connecting Hub0 to Switch0

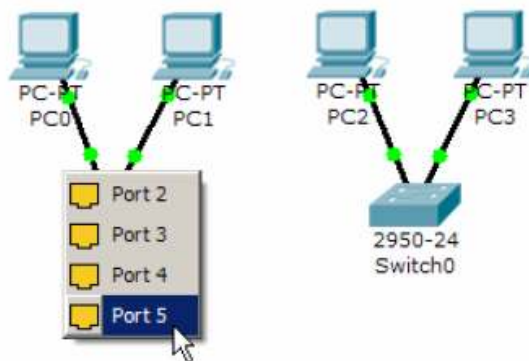
To connect like-devices, like a Hub and a Switch, we will use a Cross-over cable. Click once the Cross-over Cable from the Connections options.



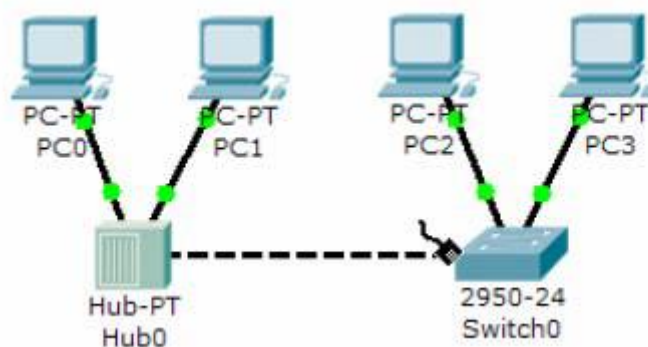
Move the Connections cursor over **Hub0** and click once.



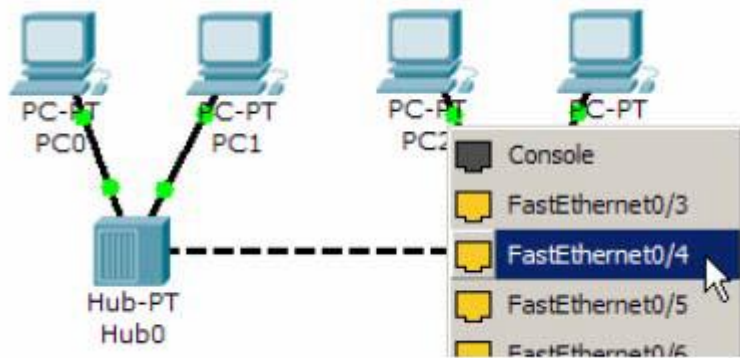
Select **Port 5** (actual port does not matter).



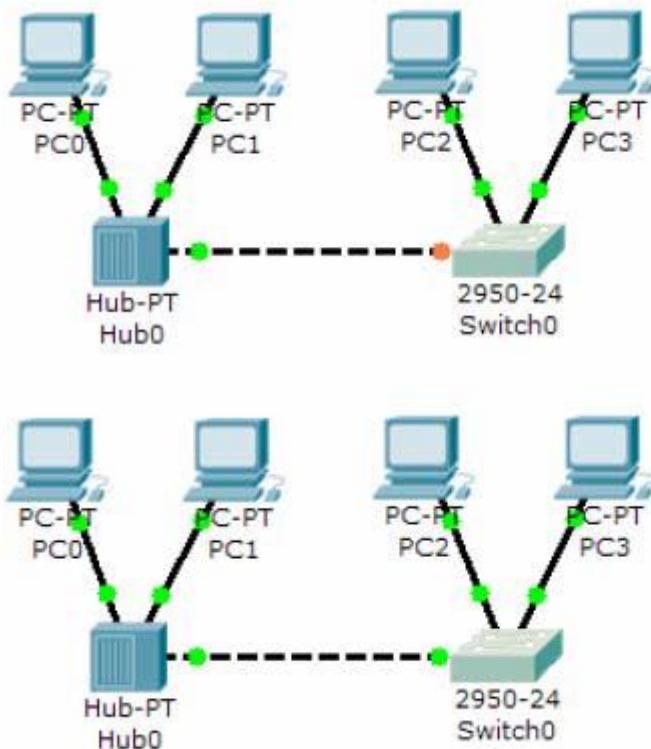
Move the Connections cursor to Switch0.



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



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



### Conclusion:

---

---

## Questions

1. What is Ciscos packet tracer? Where it is used?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
2. List different types of cable available in packet tracer. When to use cross-over cable.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
3. Explain different types of end devices available in packet tracer and where generic hub is used?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
4. Simulate a simple wireless network and test using ping command. Print the log.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
5. Which cable is used to connect router to router and router to host?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
6. List different types of ports are available with router.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



## Experiment No. 03

Name of the Student:-

\_\_\_\_\_

Roll No. \_\_\_\_\_ Subject:- \_\_\_\_\_ Computer Network

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date:

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

### **Theory:**

Topology is a pattern of network devices and describes the way in which these devices are connected. Topologies can be physical or logical. Logical topology is also called as signal topology. Every LAN has a topology, or the way that the devices on a network are arranged and how they communicate with each other. The way that the workstations are connected to the network through the actual cables that transmit data the physical structure of the network is called the physical topology. The logical topology, in contrast, is the way that the signals act on the network media, or the way that the data passes through the network from one device to the next without regard to the physical interconnection of the devices.

As Multipoint topologies share a common channel, each device needs a way to identify itself and the device to which it wants to send information. The method used to identify senders and receivers is called addressing. Following types of physical topologies are used in computer networking:

1. Bus Topology
2. Star Topology
3. Ring Topology
4. Mesh Topology
5. Tree Topology
6. Hybrid Topology

### 1. **Bus Topology**

A bus network is a network topology in which nodes are directly connected to a common linear (or branched) duplex link called a bus. Devices share a common backbone cable to



send and receive data. A thick co-axial cable is used to connect all devices. A host on a bus network is called a Station or workstation. In a bus network, every station will receive all network traffic, and the traffic generated by each station has equal transmission priority. A bus network forms a single network segment and collision domain. Bus topology uses daisy chain scheme to add more workstation in a network. In a daisy chain scheme, workstation 1 is connected to workstation 2; workstation 2 is connected to workstation 3. The first and last workstations are connected to terminator. Figure 1 shows the structure of bus topology

### Features:

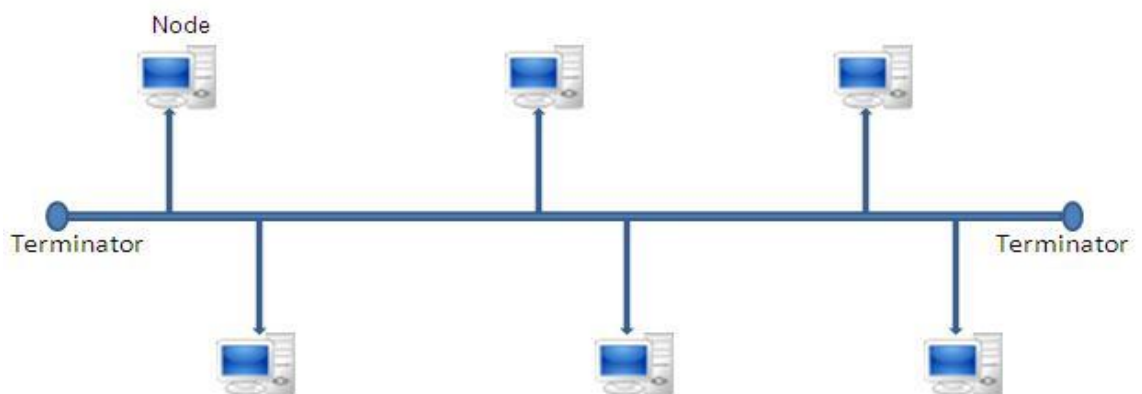
1. It transmits data only in one direction.
2. Every device is connected to a single cable

### Advantages:

1. It is cost effective.
2. Cable required is least compared to other network topology.
3. Used in small networks.
4. It is easy to understand.
5. Easy to expand joining two cables together.

### Disadvantages:

1. Cables fails then whole network fails.
2. If network traffic is heavy or nodes are more the performance of the network decreases.
3. Cable has a limited length.
4. It is slower than the ring topology.



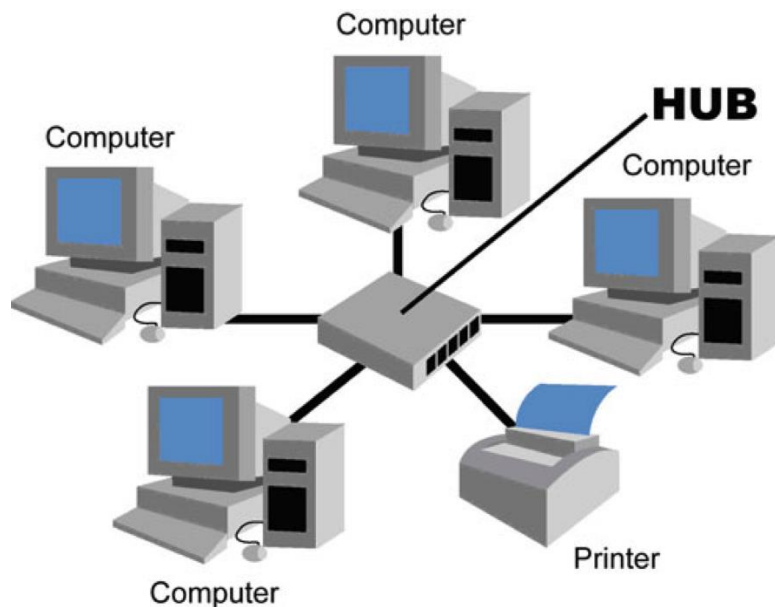
**Figure 1.1 Bus Topology**

## 2. Star Topology

In star topology, multiple devices are connected to a central connection point known as hub or switch. Devices are attached to the switch by using either copper cables or fiber optics cable. Star networks provide a cost-effective method for sharing information between different users. Star topology can be in airline reservation counters and small business offices where employees want an access to common application and files. Figure 2 shows the star topology.

### **Features:**

1. Every node has its own dedicated connection to the hub.
2. Hub acts as a repeater for data flow.
3. Can be used with twisted pair, Optical Fiber or coaxial cable.



**Figure 1.2 Star Topology**

### **Advantages:**

1. Fast performance with few nodes and low network traffic.
2. Hub can be upgraded easily.
3. Easy to troubleshoot.
4. Easy to setup and modify.
5. Only that node is affected which has failed, rest of the nodes can work smoothly.

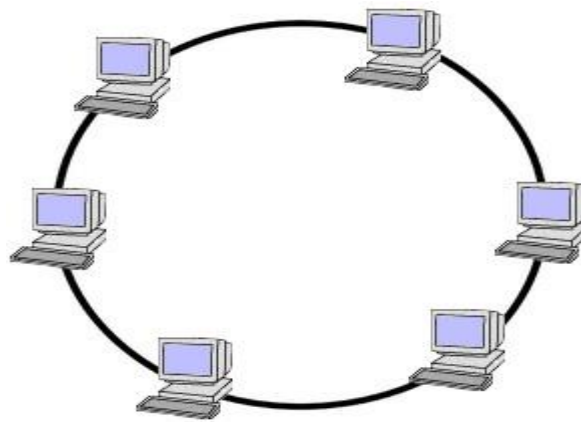
### **Disadvantages:**

1. Cost of installation is high.

2. Expensive to use.
3. If the hub fails then the whole network is stopped because all the nodes depend on the hub.
4. Performance is based on the hub that is it depends on its capacity

### 3. Ring Topology

Ring network topology is that in which all the terminals are arranged in a circular fashion and that all the data that is transmitted across the terminals is transferred in a circular pattern so that all the terminals receive it. In this kind of network topology, all the terminals have a data receiver to get all the signals from the predecessor computer and a data transmitter that ejects all the signals out to the next neighboring computer. The flow of the data depends on the speed of the transmission. Data flows in only one direction, clockwise or anti-clock wise. Each device in the ring topology acts as a repeater. It amplifies the signal and transmits it to the next device. Every node has a critical ring and every node has a signal repeater so that the data transmission remains strong. Figure 3 shows the ring topology.



**Figure 1.3 Ring Topology**

#### **Features:**

1. The transmission is unidirectional, but it can be made bidirectional by having 2 connections between each Network Node, it is called Dual Ring Topology.
2. In Dual Ring Topology, two ring networks are formed, and data flow is in opposite direction in them. Also, if one ring fails, the second ring can act as a backup, to keep the network up.
3. Data is transferred in a sequential manner that is bit by bit. Data transmitted, has

to pass through each node of the network, till the destination node.

### **Advantages:**

1. Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data.
2. Cheap to install and expand

### **Disadvantages:**

1. Troubleshooting is difficult in ring topology.
2. Adding or deleting the computers disturbs the network activity.
3. Failure of one computer disturbs the whole network.

## **4. Mesh Topology**

In mesh topology each device is connected to every other device i.e. it is a point-to-point connection. A device can send data to all the devices in the network. Mesh has  $n(n-1)/2$  physical channels to link  $n$  devices. Data sent by the other device can take any possible path to reach the destination. There are two techniques to transmit data over the Mesh topology, they are:

1. **Routing:** In routing, the nodes have a routing logic, as per the network requirements. Like routing logic to direct the data to reach the destination using the shortest distance. Or, routing logic which has information about the broken links, and it avoids that node etc. We can even have routing logic, to re-configure the failed nodes.
2. **Flooding:** In flooding, the same data is transmitted to all the network nodes; hence no routing logic is required. The network is robust, and it is very unlikely to lose the data. But it leads to unwanted load over the network.

### **Types of Mesh Topology:**

1. **Partial Mesh Topology:** In this topology some of the systems are connected in the same fashion as mesh topology but some devices are only connected to two or three devices.
2. **Full Mesh Topology:** Each and every nodes or devices are connected to each other.

### **Features:**

1. Fully connected.

2. Robust.
3. Not flexible.

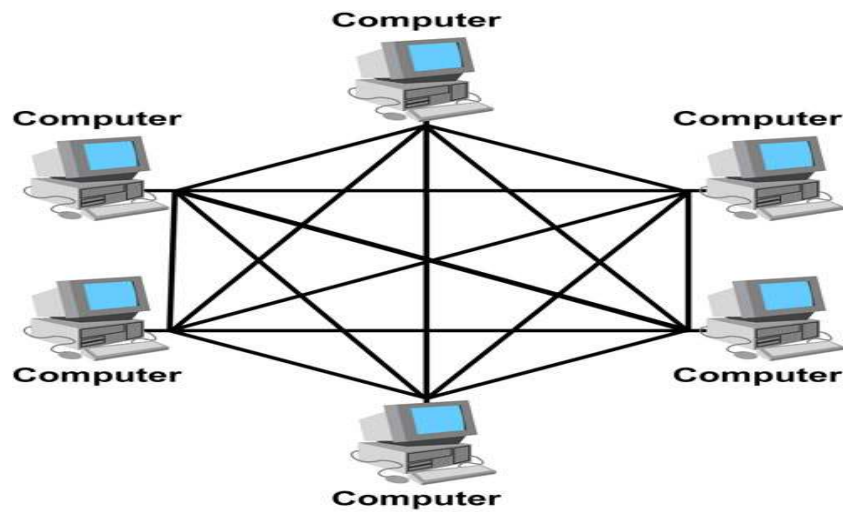


Figure 1.4 Mesh Topology

**Advantages:**

1. Each connection can carry its own data load.
2. It is robust.
3. Fault is diagnosed easily.
4. Provides security and privacy.

**Disadvantages:**

1. Installation and configuration is difficult.
2. Cabling cost is more.
3. Bulk wiring is required.

**5. Tree Topology**

It has a root node and all other nodes are connected to it forming a hierarchy. It is also called hierarchical topology. It should at least have three levels to the hierarchy.

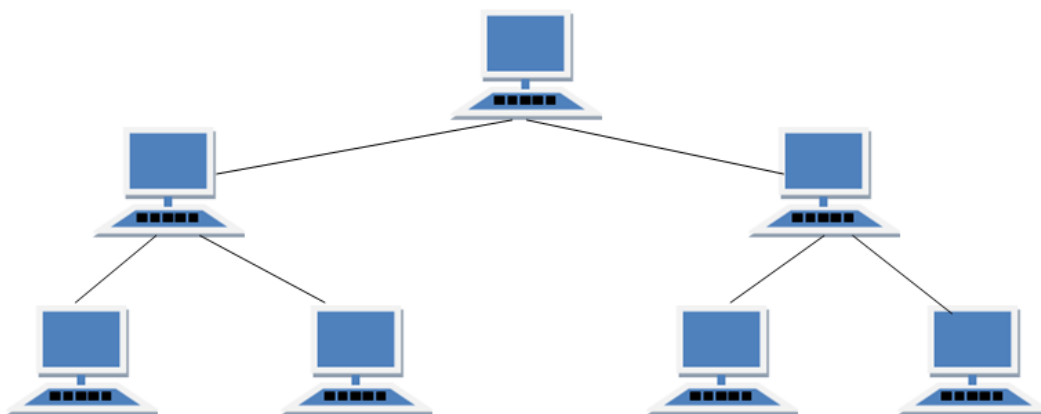


Figure 1.5 Tree Topology

**Features:**

1. Ideal if workstations are located in groups.
2. Used in Wide Area Network.

**Advantages:**

1. Extension of bus and star topologies.
2. Expansion of nodes is possible and easy.
3. Easily managed and maintained.
4. Error detection is easily done.

**Disadvantages:**

1. Heavily cabled.
2. Costly.
3. If more nodes are added maintenance is difficult.
4. Central hub fails, network fails.

## 6. Hybrid Topology

Hybrid topology is a combination of different network topologies. It is also known as a Special Topology. This topology is useful for corporate offices to link their internal LANs together while adding external networks through Wide Area Networks (WANs). For example if in an office in one department ring topology is used and in another star topology is used, connecting these topologies will result in Hybrid Topology (ring topology and star topology).

**Features:**

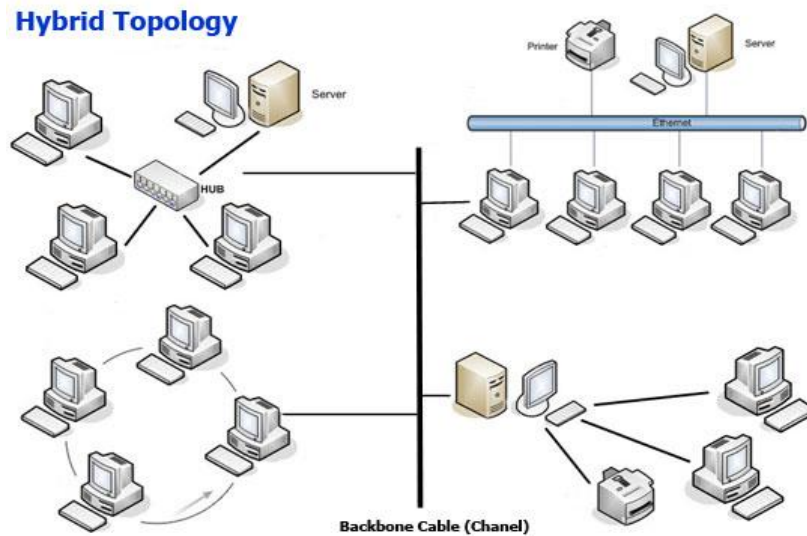
1. It is a combination of two or topologies
2. Inherits the advantages and disadvantages of the topologies included

**Advantages:**

1. Reliable as Error detecting and trouble shooting is easy.
2. Effective.
3. Scalable as size can be increased easily.
4. Flexible.

**Disadvantages:**

1. Complex in design.
2. Costly.



**Figure 1.6 Hybrid Topology**

## **Conclusion:**

---

---

---

## **Questions**

1. What are the advantages of a star topology over the bus topology?

---

---

---

2. Why is termination important on a physical bus topology?

---

---

---

3. How do hosts on a physical ring topology communication?

---

---

---

4. How does the logical topology differ from the physical topology? Why can a single physical topology support multiple logical topologies?

---

---

---

5. How does the logical topology differ from the physical topology? Why can a single physical topology support multiple logical topologies?

6. What is defined by the logical topology?

7. Compare Bus, Ring, Star and Mesh Topology.



## EXPERIMENT NO: 4

Name of the Student:- \_\_\_\_\_

Roll No. \_\_\_\_\_

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date  
& Marks

--

### Aim:

**Understand the operation of packet sniffer tools like wireshark**

### Theory:

#### What is packet sniffer?

- Packet sniffing may sound like the latest street drug craze but it's far from it. Packet sniffers or protocol analyzers are tools that are commonly used by network technicians to diagnose network-related problems. Packet sniffers can also be used by hackers for less than noble purposes such as spying on network user traffic and collecting passwords.
- Packet sniffers come in a couple of different forms. Some packet sniffers used by network technicians are single-purpose dedicated hardware solutions while other packet sniffers are software applications that run on standard consumer-grade computers, utilizing the network hardware provided on the host computer to perform packet capture and injection tasks.

#### How do Packet Sniffers Work?

Packet sniffers work by intercepting and logging network traffic that they can 'see' via the wired or wireless network interface that the packet sniffing software has access to on its host computer. On a wired network, what can be captured depends on the structure of the network. A packet sniffer might be able to see traffic on an entire network or only a certain segment of it, depending on how the network switches are configured, placed, etc. On wireless networks, packet sniffers can usually only capture one channel at a time unless the host computer has multiple wireless interfaces that allow for multichannel capture. Once the raw packet data is captured, the packet sniffing software must analyze it and present it in human-readable form so that the person using the packet sniffing software can make sense of it.

#### What Software Tools are Commonly Used in Packet Sniffing?

Just like everybody else, both network engineers and hackers love free stuff, which is why opensource and freeware sniffer software applications are often the tools of choice for packet sniffing tasks. One of the more popular open source offerings is: Wireshark (previously known as Ethereal).

#### Wireshark Network Protocol Analyzer

##### What Is Wireshark?:

Wireshark, a network protocol analyzer, otherwise known as a "packet sniffer", captures and decodes packets of information from a network. Wireshark can capture live network traffic or read data from a file and translate the data to be presented in a format the user can understand. Network analyzers such as Wireshark are invaluable tools for

administrators to diagnose and troubleshoot problems with, but are also used by intruders to obtain unauthorized information.

## What Does Wireshark Do?:

Wireshark can be used to capture and analyze network packets and discover a wide array of information such as:

- Troubleshooting network issues and locating bottlenecks
- Network intrusion detection
- Log network traffic for forensic analysis

Discovering a DoS (denial-of-service) attack

It can also be used by attackers for more nefarious purposes such as:

- Capturing usernames and passwords
- OS fingerprinting
- Capturing sensitive or proprietary information
- Network mapping

## How to Install Wireshark?

**It is successful with the set of commands below:**

```
sudo apt-get install wireshark
```

```
sudogroupaddwireshark
```

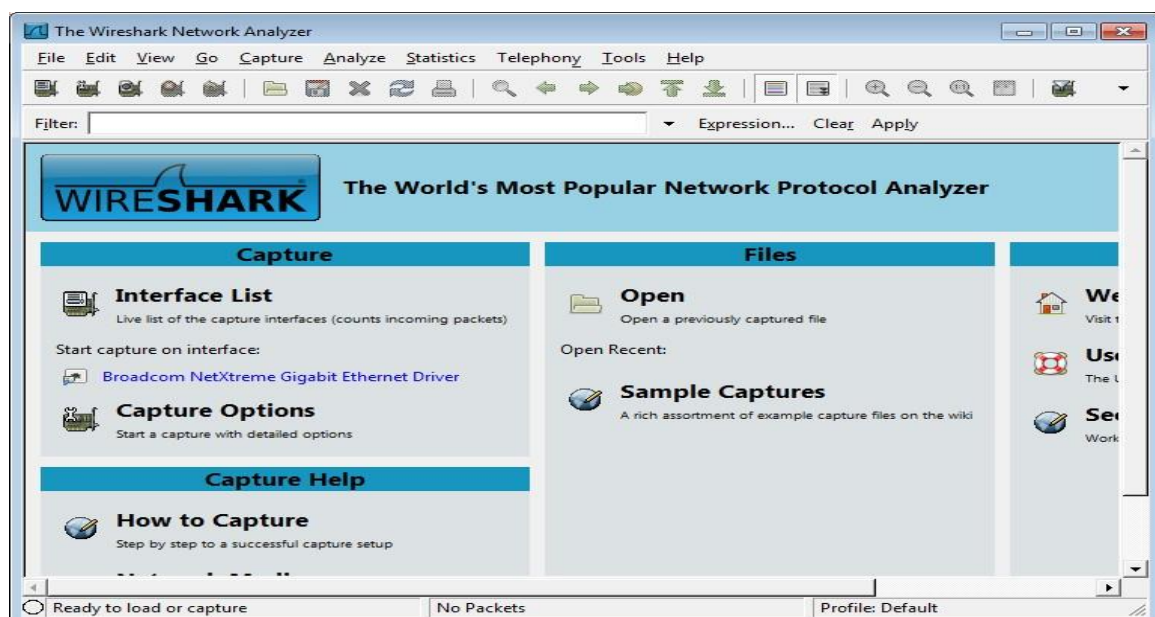
```
sudousermod -a -G wireshark YOUR_USER_NAME
```

```
sudochgrpwireshark /usr/bin/dumpcap
```

```
sudochmod 750 /usr/bin/dumpcap
```

```
sudosetcapcap_net_raw,cap_net_admin=eip /usr/bin/dumpcap
```

```
sudogetcap /usr/bin/dumpcap
```



The image shows a Wireshark packet capture window titled 'udp.cap - Wireshark'. The main pane displays a list of 11 captured packets. The selected packet (No. 10) is expanded in the bottom pane, showing the following details:

- Internet Protocol, Src: 10.144.246.184 (10.144.246.184), Dst: 216.93.191.240 (216.93.191.240)
- User Datagram Protocol, Src Port: 56627 (56627), Dst Port: openvpn (1194)
  - Source port: 56627 (56627)
  - Destination port: openvpn (1194)
  - Length: 109
  - Checksum: 0x9a15 [validation disabled]
- Data (101 bytes)

User Datagram Protocol (udp), 8 bytes

Packets: 105 Displayed: 105 Marked: 0

## Conclusion:

---

---

---

## Questions:

1. What is Packet Sniffer?

---

---

---

2. How to install Wireshark?

---

---

---

3. What is the use of Packet Sniffer?

---

---

---

4. What is Wireshark?

---

---

---

5. How to install tcpdump?

---

---

---

6. What is the use of tcpdump?

---

7. What is ethereal?

## Experiment No. 05

Name of the Student:-

Roll No. \_\_\_\_\_ Subject:- Computer Network

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date:



**Aim:** i) Installation & Configuration of NS2 in Linux Environment

ii) Implement DVR(Distance Vector Routing)

### Theory:

i) Network simulators are tools used to simulate discrete events in a network and which helps to predict the behaviours of a computer network. Generally the simulated networks have entities like links, switches, hubs, applications, etc. Once the simulation model is complete, it is executed to analyse the performance. Administrators can then customize the simulator to suit their needs. Network simulators typically come with support for the most popular protocols and networks in use today, such as WLAN,UDP,TCP,IP, WAN, etc. In communication and computer network research, network simulation is a technique whereby a software program models the behavior of a network either by calculating the interaction between the different network entities (routers, switches, nodes, access points, links etc.)

Simulating the network involves configuring the state elements like links, switches, hubs, terminals, etc. and also the events like packet drop rate, delivery status and so on. The most important output of the simulations are the trace files. Trace files log every packet, every event that occurred in the simulation and are used for analysis. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots. Most of the simulation is performed in discrete time intervals where events that are in the queue are processed one after the other in an order.

Since simulation is a complex task, we cannot guarantee that all the simulators can provide exact or accurate results for all the different type of information. Examples of network simulators are: ns, NetSim, etc.

ns2 is a name for series of discrete event network simulators like ns-1, ns-2 and ns-3. All of them are discrete-event network simulators, primarily used in research and teaching. ns2 is free software, publicly available under the GNU GPLv2 license for research, development,

and use.

This practical deals with the installation of "ns2" also called the "network simulator 2" in Ubuntu 14.04.

## Installation and Configuration

### Sept 1: Download and Extract ns2

Download the all in one package for ns2. The package downloaded will be named "ns-allinone-2.35.tar.gz". Copy it to the home folder. Then in a terminal use the following two commands to extract the contents of the package.:

```
cd
```

```
tar -xvzf ns-allinone-2.35.tar.gz
```

All the files will be extracted into a folder called "ns-allinone-2.35".

### Sept 2: Building the dependencies

Ns2 requires a few packages to be pre installed. It also requires the GCC- version 4.3 to work correctly. So install all of them by using the following command:

```
sudo apt-get install build-essential autoconf automake libxmu-dev
```

One of the dependencies mentioned is the compiler GCC-4.3, which is no longer available, and thus we have to install GCC-4.4 version. The version 4.4 is the oldest we can get. To do that, use the following command:

```
sudo apt-get install gcc-4.4
```

Once the installation is over , we have to make a change in the "ls.h" file. Use the following steps to make the changes:

Navigate to the folder "linkstate", use the following command. Here it is assumed that the ns folder extracted is in the home folder of your system.

```
cd ~/ns-allinone-2.35/ns-2.35/linkstate
```

Now open the file named "ls.h" and scroll to the 137th line. In that change the word "error" to "this->error". The image below shows the line 137 (highlighted in the image below) after making the changes to the ls.h file. To open the file use the following command:

```
gedit ls.h
```



```

root@akshay-UBPC: /home/akshay/ns-allinone-2.35/ns-2.35/linkstate
root@akshay-UBPC: /home/akshay/ns-allinone-2.35/ns-2.35# cd ns-2.35/
root@akshay-UBPC: /home/akshay/ns-allinone-2.35/ns-2.35# ls
adc          bitmap      COPYRIGHTS  gaf          Makefile    plm          satellite   validate
allinone     CHANGES.html  dccp        gen          Makefile.in puma        sctp        validate.out
aodv         classifier    delaybox    HOWTO-CONTRIBUTE  Makefile.vc pushback    sensor-nets VERSION
aomdv        common       diffserv    imcp         mcast       qs          src_rtg     webcache
apps         conf         diffusion    indep-utils  mdart       queue       tcp         test-all
asim         config.guess  diffusion3   install-sh   mobile      rap         test-all   xcp
autoconf.h   config.h      doc         INSTALL.WIN32  mpls       README      tools
autoconf.h.in config.log     dsdv        lib          nix         realeaudio  tmix
autoconf-win32.h config.status  dsr         LICENSES     ns.l        release_steps.txt TODO.html
BASE-VERSION config.sub     empweb      link         ns_tclsh.cc routealgo   tora
baytcp       configure     emulate     linkstate    packmine    routing     trace
bin          configure.in  FILES       mac          pgm         rtproto
root@akshay-UBPC: /home/akshay/ns-allinone-2.35/ns-2.35# cd linkstate/
root@akshay-UBPC: /home/akshay/ns-allinone-2.35/ns-2.35/linkstate# gedit ls.h

```

Save that file and close it.

```

// this next typedef of iterator seems extraneous but is required by gcc-2.96
typedef typename map<Key, T, less<Key> >::iterator iterator;
typedef pair<iterator, bool> pair_iterator_bool;
iterator insert(const Key & key, const T & item) {
    typename baseMap::value_type v(key, item);
    pair_iterator_bool ib = baseMap::insert(v);
    return ib.second ? ib.first : baseMap::end();
}

void eraseAll() { this->erase(baseMap::begin(), baseMap::end()); }
T* findPtr(Key key) {
    iterator it = baseMap::find(key);
    return (it == baseMap::end()) ? (T *)NULL : &((*it).second);
}

```

Now there is one more step that has to be done. We have to tell the ns which version of GCC will be used. To do so, go to your ns folder and type the following command:

*Sudo gedit ns-allinone-2.34/otcl-1.13/Makefile.in*

```

akshay@akshay-UBPC: ~/ns-allinone-2.35/otcl-1.14
akshay@akshay-UBPC:~$ cd ns-allinone-2.35/
akshay@akshay-UBPC:~/ns-allinone-2.35$ cd otcl-1.14/
akshay@akshay-UBPC:~/ns-allinone-2.35/otcl-1.14$ gedit Makefile.in

```

In the file, change Change CC= @CC@ to CC=gcc-4.4, as shown in the image below.

```
*Makefile.in x
#
# try ./configure first to fill in all the definitions corresponding
# to your system, but you always can edit the sections below manually.
#
CC= gcc-4.4
CFLAGS= @CFLAGS@
RANLIB= @RANLIB@
INSTALL= @INSTALL@

#
# how to compile, link, and name shared libraries
#
SHLIB_LD= @SHLIB_LD@
SHLIB_CFLAGS= @SHLIB_CFLAGS@
SHLIB_SUFFIX= @SHLIB_SUFFIX@
SHLD_FLAGS= @DL_LD_FLAGS@
DL_LIBS= @DL_LIBS@
```

### Step 3 Building the dependencies

Now we are ready to install ns2. To do so we first require root privileges and then we can run the install script. Use the following two commands:

```
sudo su cd ~/ns-allinone-2.35/./install
```

The following is a snap of these commands:

```
root@akshay-UBPC: /home/akshay/ns-allinone-2.35
akshay@akshay-UBPC:~$ cd ns-allinone-2.35/
akshay@akshay-UBPC:~/ns-allinone-2.35$ ls
cweb      install      ns-2.35      sgb          tk8.5.10
dei80211mr-1.1.4  INSTALL.WIN32  otcl-1.14    tcl8.5.10    xgraph-12.2
gt-itm     nam-1.15      README      tclcl-1.20   zlib-1.2.3
akshay@akshay-UBPC:~/ns-allinone-2.35$ sudo su
[sudo] password for akshay:
root@akshay-UBPC:/home/akshay/ns-allinone-2.35# ./install
```

The image below shows how it looks upon successful execution



```
Please put /home/akshay/ns-allinone-2.35/bin:/home/akshay/ns-allinone-2.35/tcl8.5.10/unix:/home/akshay/ns-allinone-2.35/tk8.5.10/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/akshay/ns-allinone-2.35/otcl-1.14, /home/akshay/ns-allinone-2.35/lib,
into your LD_LIBRARY_PATH environment variable.
If it complains about X libraries, add path to your X libraries
into LD_LIBRARY_PATH.
If you are using csh, you can set it like:
    setenv LD_LIBRARY_PATH <paths>
If you are using sh, you can set it like:
    export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/akshay/ns-allinone-2.35/tcl8.5.10/library into your TCL_LIBRARY environmental
variable. Otherwise ns/nam will complain during startup.

After these steps, you can now run the ns validation suite with
cd ns-2.35; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list archive
for related posts.

root@akshay-UBPC:/home/akshay/ns-allinone-2.35#
```

It took almost 6 minutes to build and install ns2 on my system. But before we run it, we need to add the build path to the environment path.

## Step 4 Installation

The final step is to tell the system, where the files for ns2 are installed or present. To do that, we have to set the environment path using the ".bashrc" file. In that file, we need to add a few lines at the bottom. The things to be added are given below. But for the path indicated below, many of those lines have "/home/akshay/ns-allinone-2.35/..." , but that is where I have my extracted folder. Make sure you replace them with your path. For example, if you have installed it in a folder "/home/abc", then replace "/home/akshay/ns-allinone-2.35/otcl-1.14" with "/home/abc/ns-allinone-2.35/otcl-1.14".

Do this for all the required lines.

```
sudo gedit ~/.bashrc
```

Lines to be added:

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/akshay/ns-allinone-2.35/otcl-1.14
NS2_LIB=/home/akshay/ns-allinone-2.35/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_
LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/home/akshay/ns-allinone-2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/home/akshay/ns-allinone-2.35/bin:/home/akshay/ns-allinone-
2.35/tcl8.5.10/unix:/home/akshay/ns-allinone-2.35/tk8.5.10/unix
#the above two lines beginning from xgraph and ending with unix should come on the same
line
```

```
NS=/home/akshay/ns-allinone-2.35/ns-2.35/  
NAM=/home/akshay/ns-allinone-2.35/nam-1.15/  
PATH=$PATH:$XGRAPH:$NS:$NAM
```

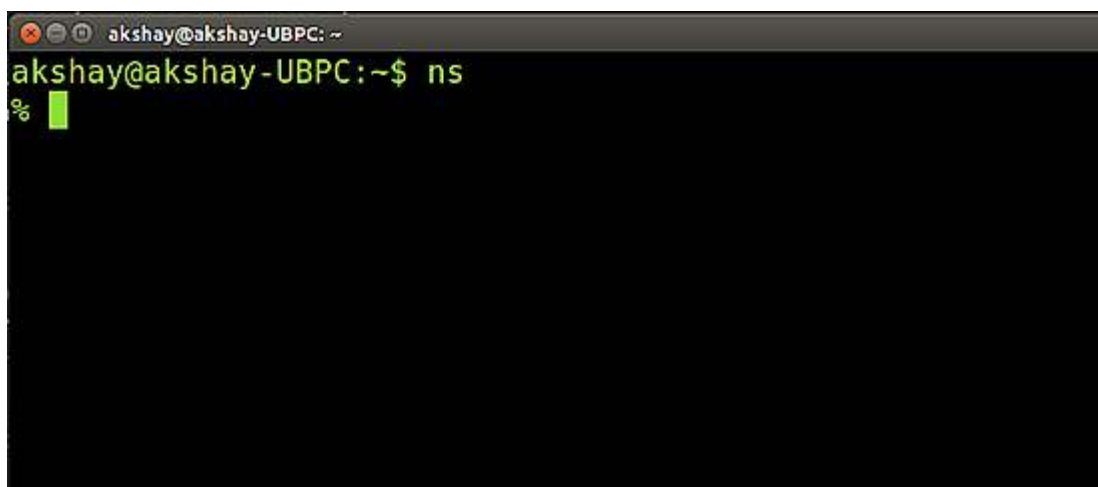
Once the changes have been made, save the file and restart the system.

### Step 5 Running ns2

Once the system has restarted, open a terminal and start ns2 by using the following command:

```
ns
```

If the installation is correct then the terminal looks like the image below



### Simulation workflow

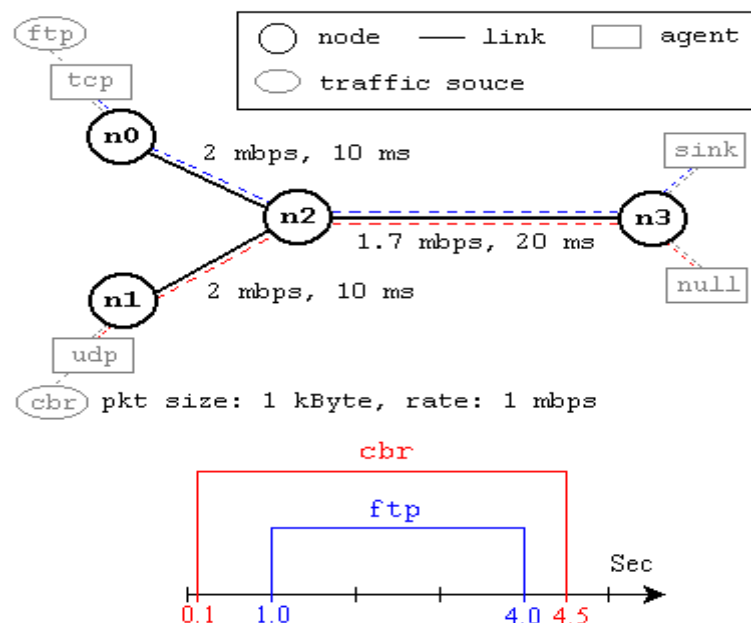
The general process of creating a simulation can be divided into several steps:

1. **Topology definition:** To ease the creation of basic facilities and define their interrelationships, ns-3 has a system of containers and helpers that facilitates this process.
2. **Model development:** Models are added to simulation (for example, UDP, IPv4, point-to-point devices and links, applications); most of the time this is done using helpers.
3. **Node and link configuration:** models set their default values (for example, the size of packets sent by an application or MTU of a point-to-point link); most of the time this is done using the attribute system.
4. **Execution:** Simulation facilities generate events, data requested by the user is logged.

5. **Performance analysis:** After the simulation is finished and data is available as a time-stamped event trace. This data can then be statistically analysed with tools like **R** to draw conclusions.
6. **Graphical Visualization:** Raw or processed data collected in a simulation can be graphed using tools like **Gnuplot**, **matplotlib** or **XGRAPH**.

Following simple network as shown in figure 4.1 is simulated in NS2

This network consists of 4 nodes (n0, n1, n2, n3) as shown in above figure. The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte. A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec



**Figure 4.1** A Simple Network Topology and Simulation Scenario

## Example of TCL Script

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
```

```
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
```

```
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
```

```
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
```

```
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
```

```
#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"
```

```
#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
```

```
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
```

```
#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"
```

```
#Run the simulation
$ns run
```

ii) Distance Vector Routing (DVR) The name distance vector is derived from the fact

that routes are advertised as vectors of (distance, direction), where distance is defined in terms of a metric and direction is defined in terms of the next-hop router. For example, "Destination A is a distance of 5 hops away, in the direction of next-hop router X." As that statement implies, each router learns routes from its neighboring routers' perspectives and then advertises the routes from its own perspective. Because each router depends on its neighbors for information, which the neighbors in turn may have learned from their neighbors, and so on, distance vector routing is sometimes facetiously referred to as "routing by rumor."

The common Characteristics are

**Periodic Updates :**Periodic updates means that at the end of a certain time period, updates will be transmitted.

**Neighbors :**In the context of routers, neighbors always mean routers sharing a common data link.

**Broadcast Updates:** When a router first becomes active on a network, how does it find other routers and how does it announce its own presence? Several methods are available.

**Full Routing Table Updates :**Most distance vector routing protocols take the very simple approach of telling their neighbors everything they know by broadcasting their entire route table, with some exceptions .

**Split Horizon:** A route pointing back to the router from which packets were received is called a reverse route. Split horizon is a technique for preventing reverse routes between two routers.

## Conclusion:

---

---

## Questions:

1. What is mean by Network Simulator?

---

---

---

2. List different Network Simulator.

---

---

---

3. Which are the pre-requisite software for NS2?

---

---

---

4. Explain the simulation workflow.

---

---

5. What is the difference between ns-1, ns-2 and ns-3?

---

---

---

6. List all the commands used in installation of ns2.

---

---

---

## EXPERIMENT NO: 6

Name of the Student :-

Roll No. \_\_\_\_\_

Subject:-

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date  
& Marks

**Aim:** i) Setting up multiple IP Addresses on a single LAN.  
ii) Using netstat and route commands.

### Theory:

i) The concept of creating or configuring multiple IP addresses on a single network interface is called **IP aliasing**. IP aliasing is very useful for setting up multiple virtual sites on **Apache** using one single network interface with different **IP addresses** on a single subnet network.

The main advantage of using this **IP aliasing** is, you don't need to have a physical adapter attached to each **IP**, but instead you can create multiple or many virtual interfaces to a single physical card.

### Step 1:

#### Ifconfig

```
rohanmh@fossee: ~$ ifconfig
eth0      Link encap:Ethernet  HWaddr e0:db:55:a2:da:7a
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 9c:2a:70:d5:94:4d
          inet addr:10.101.201.248  Bcast:10.101.255.255  Mask:255.255.0.0
          inet6 addr: fe80::9e2a:70ff:fed5:944d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:0 overruns:0 frame:809
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8151 (8.1 KB)  TX bytes:4640 (4.6 KB)
          Interrupt:19

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:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3500 (3.5 KB)  TX bytes:3500 (3.5 KB)

rohanmh@fossee:~$
```

From this image there is having eth1 interface available to me Now set ip address for interface eth0

Steps are as follows :



## Step 2

edit /etc/network/interfaces(Change ip address as per your requirement)

```
#sudo nano /etc/network/interfaces
```

```
auto lo eth0
iface lo inet loopback
iface eth0 inet static
address 10.101.201.230
netmask 255.255.0.0
gateway 10.101.201.230
```

## Step 3

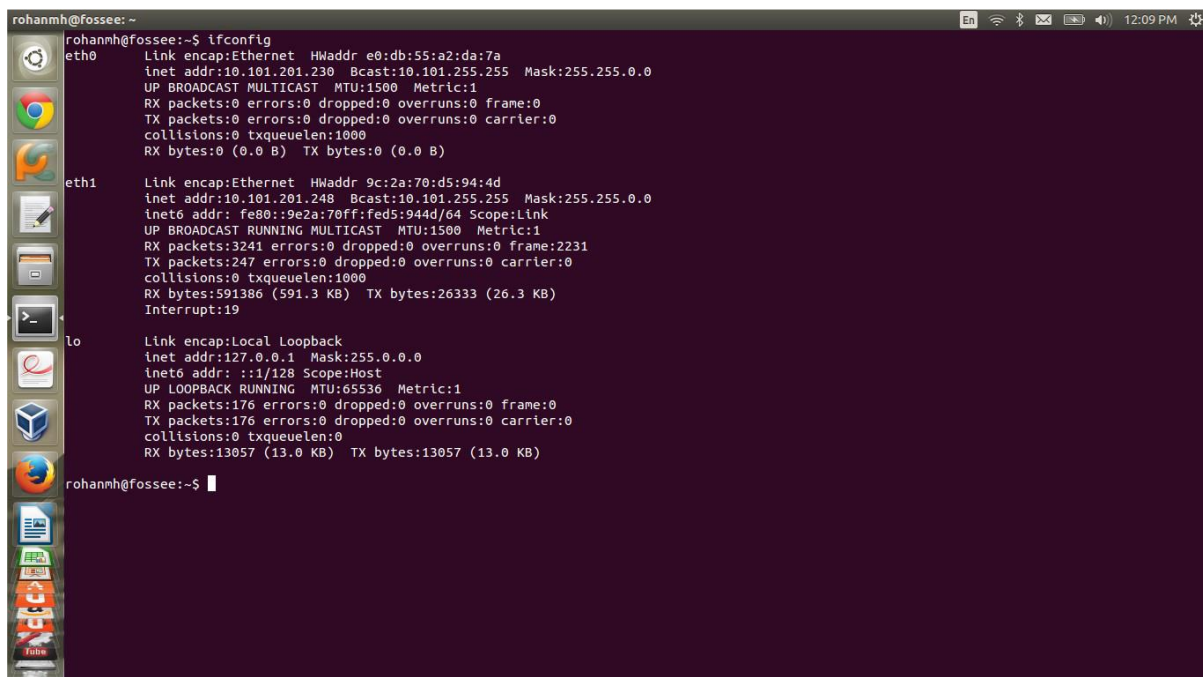
restart networking service or restart computer

```
#sudo service networking restart
```

## Step 4

check output using ifconfig cmd

```
#ifconfig
```



```
rohanmh@fossee:~$ ifconfig
eth0: Link encap:Ethernet  HWaddr e0:db:55:a2:da:7a
      inet addr:10.101.201.230  Bcast:10.101.255.255  Mask:255.255.0.0
      UP BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1: Link encap:Ethernet  HWaddr 9c:2a:70:d5:94:4d
      inet addr:10.101.201.240  Bcast:10.101.255.255  Mask:255.255.0.0
      inet6 addr: fe80::9e2a:70ff:fed5:944d/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:3241 errors:0 dropped:0 overruns:0 frame:2231
      TX packets:247 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:591386 (591.3 KB)  TX bytes:26333 (26.3 KB)
      Interrupt:19

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:176 errors:0 dropped:0 overruns:0 frame:0
      TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:13057 (13.0 KB)  TX bytes:13057 (13.0 KB)

rohanmh@fossee:~$
```

**And another way:**

Step1: Ifconfig

Step2 : sudo su

Step3: ifconfig eth0:0 172.17.13.40 netmask 255.255.128.0

Step4: ifconfig eth0:1 172.17.13.41 netmask 255.255.128.0

Step5: ifconfig

**How to View Your Current Routing Table**

The `netstat -nr` command will provide the contents of the routing table. Networks with a gateway of 0.0.0.0 are usually directly connected to the interface. No gateway is needed to reach your own directly connected interface, so a gateway address of 0.0.0.0 seems appropriate. The route with a destination address of 0.0.0.0 is your default gateway. In this example there are two gateways, the default and one to 255.255.255.255 which is usually added on DHCP servers. Server bigboy is a DHCP server in this case.

```
[root@bigboy tmp]# netstat -nr
```

```
administrator@administrator-H81M-DS2:~$ netstat -nr
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
0.0.0.0          172.17.1.1      0.0.0.0         UG      0  0        0 eth0
169.254.0.0      0.0.0.0         255.255.0.0     U       0  0        0 eth0
172.17.0.0       0.0.0.0         255.255.128.0   U       0  0        0 eth0
```

`netstat` - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

**netstat -a** command shows all connections from different protocols like tcp, udp and unix sockets. However this is not quite useful. Administrators often want to pick out specific connections based on protocols or port numbers for example.

**netstat -at** To list out only tcp connections use the `t` options.

**netstat -au** Similarly to list out only udp connections use the `u` option.

**Netstat -tnl** Any network daemon/service keeps an open port to listen for incoming connections. These too are like socket connections and are listed out by `netstat`. To view only listening ports use the `l` options. Now we can see only listening tcp ports/connections. If you want to see all listening ports, remove the `t` option. If you want to see only listening udp ports use the `u` option instead of `t`. Make sure to remove the 'a' option, otherwise all connections would get listed and not just the listening connections.

**Netstat -s** The `netstat` command can also print out network statistics like total number of packets received and transmitted by protocol type and so on. To list out statistics of all packet types. To print out statistics of only select protocols like TCP or UDP use the corresponding options like `t` and `u` along with the `s` option. Simple!

**Netstat -i** The `netstat` command can also print out the information about the network interfaces. The `i` option does the task.

**Netstat -g** The `g` option will display the multicast group information for IPv4 and IPv6 protocols.

The `-r` option specifies that you want the routing table. The `-n` option is similar to that of the `route` command.

**\$ sudo route -n**

```
administrator@administrator-H81M-DS2:~$ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
0.0.0.0          172.17.1.1      0.0.0.0         UG      0      0      0 eth0
169.254.0.0      0.0.0.0         255.255.0.0     U       1000    0      0 eth0
172.17.0.0       0.0.0.0         255.255.128.0   U        1      0      0 eth0
```

The `-n` option means that you want numerical IP addresses displayed, instead of the corresponding host names.

## Adding Temporary Static Routes

The route add command can be used to add new routes to your server that will last till the next reboot. It has the advantage of being universal to all versions of Linux and is well documented in the man pages. In our example the reference to the 10.0.0.0 network has to be preceded with a -net switch and the subnet mask and gateway values also have to be preceded by the netmask and gw switches respectively.

```
[root@bigboy tmp]# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

If you wanted to add a route to an individual server, then the "-host" switch would be used with no netmask value. (The route command automatically knows the mask should be 255.255.255.255). Here is an example for a route to host 10.0.0.1.

```
[root@bigboy tmp]# route add -host 10.0.0.1 gw 192.168.1.254 wlan0
```

A universal way of making this change persistent after a reboot would be to place this route add command in the file /etc/rc.d/rc.local, which is always run at the end of the booting process.

## Delete a Route

Here's how to delete the routes added in the previous section.

```
[root@bigboy tmp]# route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

## Change Your Default Gateway

Your server needs to have a single default gateway. DHCP servers will automatically assign a default gateway to DHCP configured NICs, but NICs with configured static IP addresses will need to have a manually configured default gateway.

## Temporary Default Gateway Assignment

You can change temporarily change your default gateway till the next reboot using a simple command. This example uses a newly installed wireless interface called wlan0, most PCs would be using the standard Ethernet interface eth0.

```
[root@bigboy tmp]# route add default gw 192.168.1.1
```

## Conclusion:

---

---

## Frequently Asked Questions:

1. What is IP aliasing?

---

---

---

2. What is the main advantage if using IP aliasing?

---

---

---

3. What is the use of sudo command in linux?

---

---

---

4. What is netmask?

---

---

---

5. What is subnet mask?

---

---

---

6. What is subnetting?

---

---

---

7. What is routing table and what information does it contain?

---

---

---

8. What is netstat command?

---

---

---

9. What is a default gateway?

---

---

---

## Experiment No. 07

Name of the Student:-

\_\_\_\_\_

Roll No. \_\_\_\_\_ Subject:- \_\_\_\_\_ Computer Network

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date:

--

**Aim:** A client-server application using socket programming in Java

### Theory:

#### Client-Server Model

The client-server model of computing is a distributed application that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests

#### Socket Programming

Sockets allow communication between two different processes on the same or different machines. Sockets allow applications to communicate using standard mechanisms built into network hardware and operating systems. A socket represents a single connection between exactly two pieces of software. More than two pieces of software can communicate in client/server or distributed systems (for example, many Web browsers can simultaneously communicate with a single Web server) but multiple sockets are required to do this. Sockets are bidirectional, meaning that either side of the connection is capable of both sending and receiving data. Libraries implementing sockets for Internet Protocol use TCP for streams, UDP for datagrams, and IP itself for raw sockets.

#### Primitive used in Socket:

Primitives	Meaning
<b>SOCKET</b>	Create a New Communication Endpoint.
<b>BIND</b>	Attach a Local Address to a SOCKET.
<b>LISTEN</b>	Shows the Willingness to Accept Connections.
<b>ACCEPT</b>	Block the Caller until a Connection Attempts Arrives.
<b>CONNECT</b>	Actively Attempt to Establish a Connection.
<b>SEND</b>	Send Some Data over Connection.
<b>RECEIVE</b>	Receive Some Data from the Connection.
<b>CLOSE</b>	Release the Connection.

- i. **SOCKET:** It creates a new end point. It allocates table space for the new end point, within the transport entity. It has to be used by both, the server as well as the client, to create server socket and client socket respectively.
- ii. **BIND:** Newly created server sockets do not have a network address. The BIND primitive assigns a network address to a server socket. The network address is used so that clients can connect to the server socket. The BIND primitive is not required for client sockets.
- iii. **LISTEN:** It is used to maintain a queue of the clients who want to connect to the server socket. The difference between this primitive and the LISTEN primitive in the previous section is that, the LISTEN in this case is not a blocking LISTEN (i.e. the server will not be blocked on executing this LISTEN).
- iv. **ACCEPT:** It blocks the server and makes it wait for an incoming connection.
- v. **CONNECT:** It blocks the client and tries to establish a connection with the server.
- vi. **SEND and RECEIVE:** After the connection is established between the client and the server, data can be exchanged using the SEND and RECEIVE primitives.
- vii. **CLOSE:** The connection is released.

## Socket Programming:

### I) Server side:

Server startup executes SOCKET, BIND— & LISTEN primitives.

LISTEN primitive allocate queue for multiple simultaneous clients.

Then it use ACCEPT to suspend server until request.

When client request arrives: ACCEPT returns.

Start new socket (thread or process) with same properties as original, this handles the request, server goes on waiting on original socket.

If new request arrives while spawning thread for this one, it is queued.

If queue full it is refused.

### Example for server

```
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);}
    }
}
```

### II) Client side:

It uses SOCKET primitives to create.

Then use CONNECT to initiate connection process.

When this returns the socket is open.

Both sides can now SEND, RECEIVE.

Connection not released until both sides do CLOSE.

Typically client does it, server acknowledges.

### Example for client

```
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);}
}
}
```

## **Conclusion:**

---

---

---

## **Questions**

---

1) What is Socket?

---

---

---

2) What does bind() method of ServerSocket offer?

---

---

---

3) What happens if ServerSocket is not able to listen on the specified port?

---

---

---

4) How does applet and servlet communicate?

---

---

---

5) Which of the below are common network protocols?

---

---

---

6) What is the java method for ping?



---

---

---

7) What happens if IP Address of host cannot be determined?

---

---

---

## Experiment No. 08

Name of the Student:-

Roll No. \_\_\_\_\_ Subject:- Computer Network

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date:

**Aim:** Study and implementation of CRC algorithm

### **Theory:**

When a codeword is transmitted, one or more number of transmitted bits will be reversed (0 to 1 or vice versa) due to transmission impairments. Thus error will be introduced. It is possible for the receiver to detect the error if the received codeword (corrupted) is not one of valid codeword. Hence to detect error at receiver, the valid codeword's should be separated by a distance of more than 1. Otherwise the incorrect received codeword will also become some other valid codeword and the error detection will be impossible. The number of error that can be detected depends on the distance between any two valid codeword's.

Some of the most important error detection methods are as follow:

1. Parity checking
2. Checksum error detection
3. Cyclic Redundancy Code (CRC)

A Cyclic Redundancy Check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short check value attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption. CRCs can be used for error correction.

CRCs are so called because the check (data verification) value is a redundancy (it expands the message without adding information) and the algorithm is based on cyclic codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyze mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function.

A CRC is derived using a more complex algorithm than the simple CHECKSUM, involving MODULO ARITHMETIC (hence the 'cyclic' name) and treating each input word as a set of coefficients for a polynomial. It is not based on binary addition Rather it is based on binary division. At the sender side, the data unit to be transmitted IS divided by a predetermined divisor (binary number) in order to obtain the remainder. This remainder is called CRC.

The CRC has one bit less than the divisor. It means that if CRC is of  $n$  bits, divisor is of  $n+1$  bit.

The sender appends this CRC to the end of data unit such that the resulting data unit becomes exactly divisible by predetermined divisor i.e. remainder becomes zero. At the destination, the incoming data unit i.e. data + CRC is divided by the same number (predetermined binary divisor).

If the remainder after division is zero then there is no error in the data unit & receiver accepts it.

If remainder after division is not zero, it indicates that the data unit has been damaged in transit and therefore it is rejected. This technique is more powerful than the parity check and checksum error detection. CRC is based on binary division. A sequence of redundant bits called CRC or CRC remainder is appended at the end of a data unit such as byte.

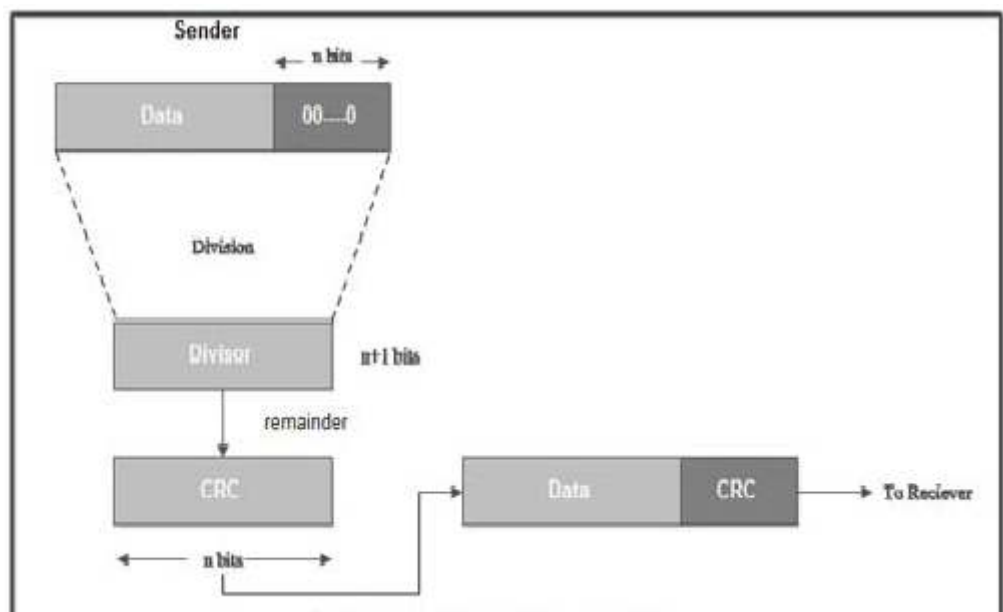
### *Requirements of CRC :*

A CRC will be valid if and only if it satisfies the following requirements:

- It should have exactly one less bit than divisor.
- Appending the CRC to the end of the data unit should result in the bit sequence which is exactly divisible by the divisor.

The various steps followed in the CRC method are

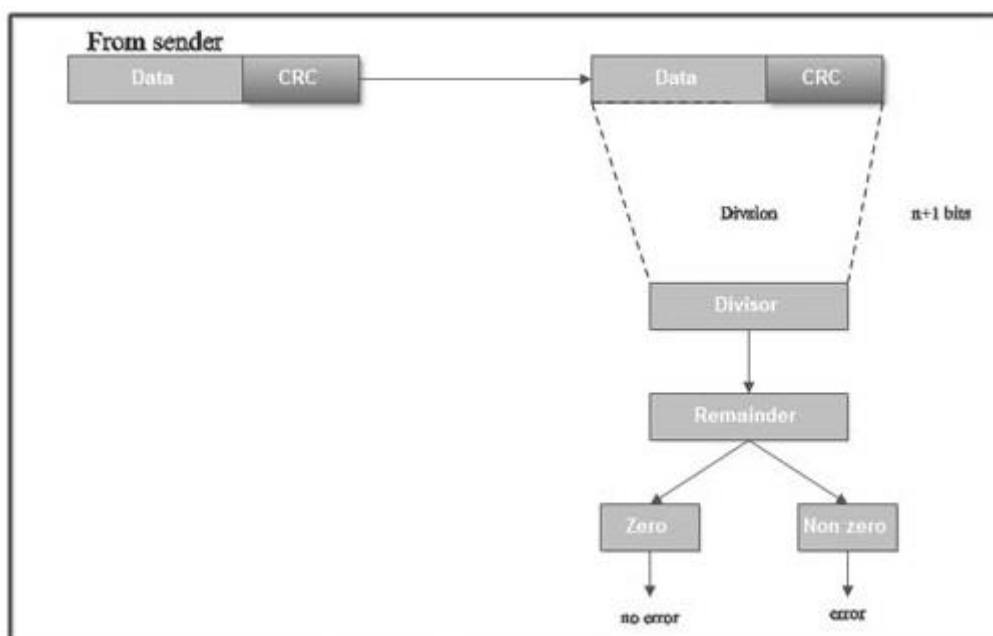
1. A string of  $n$  as is appended to the data unit. The length of predetermined divisor is  $n+1$ .
2. The newly formed data unit i.e. original data + string of  $n$  as are divided by the divisor using binary division and remainder is obtained. This remainder is called CRC.



**Figure 5.1 Sender side process of CRC generator**

3. String of  $n$  zeros appended to data unit is replaced by the CRC remainder (which is also of  $n$  bit).
4. The data unit + CRC is then transmitted to receiver.
5. The receiver on receiving it divides data unit + CRC by the same divisor & checks the remainder.
6. If the remainder of division is zero, receiver assumes that there is no error in data and it accepts it.
7. If remainder is non-zero then there is an error in data and receiver rejects it.

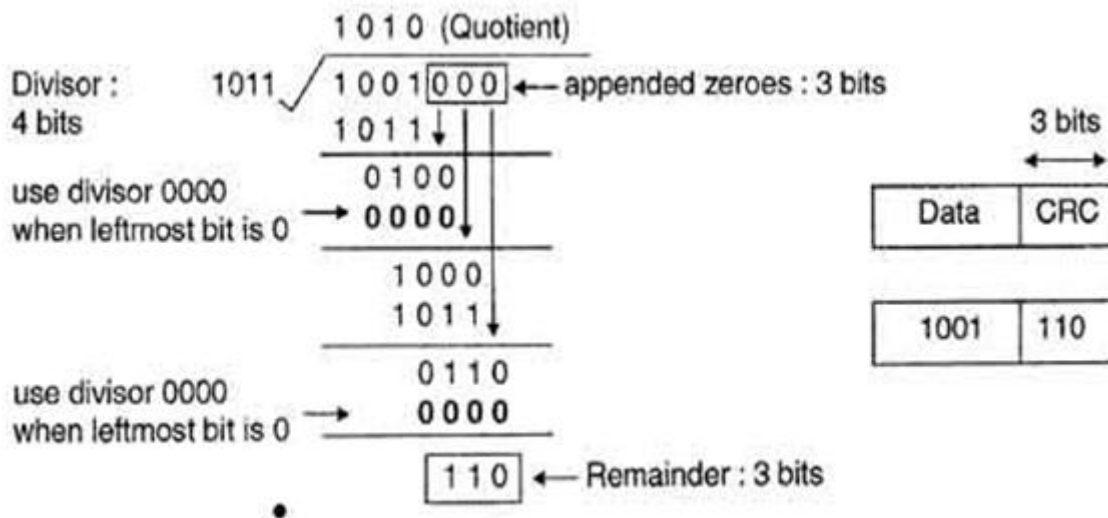
Example: if data to be transmitted is 1001 and predetermined divisor is 1011. The procedure given below is used:



**Figure 5.2 Ckecker at reciver side process of CRC generator**

String of 3 zeroes is appended to 1011 as divisor is of 4 bits. Now newly formed data is 1011000.

1. Data unit 1011000 is divided by 1011.



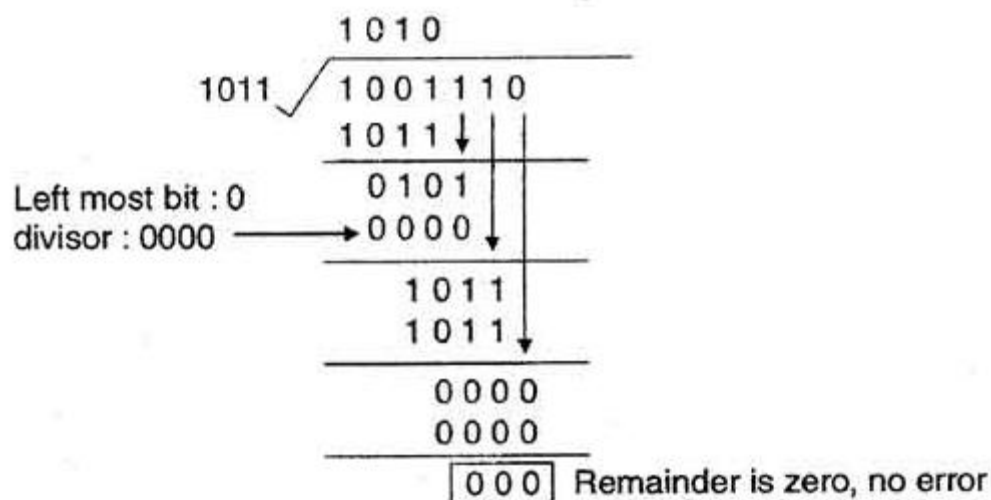
**Figure 5.3 CRC generated (Binary division)**

During this process of division, whenever the leftmost bit of dividend or remainder is 0, we use a string of 0s of same length as divisor. Thus in this case divisor 1011 is replaced by 0000.

At the receiver side, data received is 1001110.

This data is again divided by a divisor 1011.

The remainder obtained is 000; it means there is no error.



**Figure 5.4 CRC decoded (Binary division)**

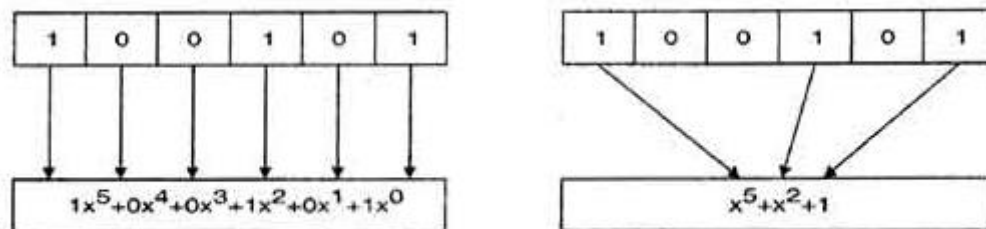
- CRC can detect all the burst errors that affect an odd number of bits.
- The probability of error detection and the types of detectable errors depends on the choice of divisor.

- Thus two major requirement of CRC are:
  - (a) CRC should have exactly one bit less than divisor.
  - (b) Appending the CRC to the end of the data unit should result in the bit sequence which is exactly divisible by the divisor.

## Polynomial codes

A pattern of 0s and 1s can be represented as a polynomial with coefficient of 0 and 1. Here, the power of each term shows the position of the bit and the coefficient shows the values of the bit.

For example, if binary pattern is 100101, its corresponding polynomial representation is  $x^5 + x^2 + 1$ . Figure shows the polynomial where all the terms with zero coefficient are removed and  $x^j$  is replaced by  $x$  and  $x^0$  by 1.



**Figure 5.4 Binary pattern and its polynomial representation**

The benefits of using polynomial codes are that it produces short codes. For example here a 6-bit pattern is replaced by 3 terms.

In polynomial codes, the degree is 1 less than the number of bits in the binary pattern. The degree of polynomial is the highest power in polynomial. For example as shown in fig degree of polynomial  $x^5 + x^2 + 1$  are 5. The bit pattern in this case is 6.

```
C:\Users\Omkar\Desktop>javac crc_gen.java
C:\Users\Omkar\Desktop>java -cp . crc_gen
Enter number of data bits :
4
Enter data bits :
1
0
0
1
Enter number of bits in divisor :
4
Enter Divisor bits :
1
0
0
1
Dividend (after appending 0's) are : 1001000
CRC code :
1001110
Enter CRC code of 7 bits :
1
0
0
1
1
1
0
No Error
THANK YOU.... :>
C:\Users\Omkar\Desktop>

C:\Users\Omkar\Desktop>javac crc_gen.java
C:\Users\Omkar\Desktop>java -cp . crc_gen
Enter number of data bits :
4
Enter data bits :
1
0
0
1
Enter number of bits in divisor :
4
Enter Divisor bits :
1
0
0
1
Dividend (after appending 0's) are : 1001000
CRC code :
1001110
Enter CRC code of 7 bits :
1
0
0
1
1
1
0
Error
THANK YOU.... :>
C:\Users\Omkar\Desktop>
```

**Conclusion:**

---

---

---

## Questions

1. What is an Error? Define Error Control.

---

---

---

2. What are three types of redundancy checks used in data communication?

---

---

---

3. Define a) Code word b) Code rate c) Code efficiency

---

---

---

4. Discuss the concept of redundancy in error detection.

---

---

---

5. Explain CRC method of Error Detection?

---

---

---

6. Obtain the 4-bit CRC code word for the data bit sequence 10011011100 (leftmost bit is the least significant) using the generator 1101.

---

---

---



## EXPERIMENT NO: 9

Name of the Student:-

Roll No. \_\_\_\_\_ Subject:-

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date  
& Marks

--

**Aim:** Perform File Transfer and Access using FTP

- i. Set up anonymous access of FTP server.
- ii. Enable individual logins and add FTP users with Read- only access.
- iii. Transfer Files.

### Theory:

The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server.[1] FTP users may authenticate themselves using a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different.

The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as Web page editors.

\*To check vsftpd packages installed or not in the system  
#dpkg -s vsftpd

**\*To install ftp server in ubuntu we use the application VSFTPD**

#sudo apt-get update #sudo apt-get install vsftpd
--

### i. Set up anonymous access of FTP server.

Vsftpd is having configuration file vsftpd.conf

Open terminal using Ctrl+Alt+T

Open the configuration file using nano editor

```
#sudo nano /etc/vsftpd.conf
```

Search for anonymous \_enable keyword and make changes shown as follows  
anonymous\_enable=Yes

To exit from nano editor press (Ctrl+x) and then Enter

To restart vsftpd service

```
#sudo service vsftpd restart
```

In vsftpd anonymous user is having default location for home directory as /srv/ftp

To check, open terminal and goto /srv/ftp location and create a file

Open terminal using Ctrl+Alt+T

```
#cd /srv/ftp  
#sudo touch abc
```

Open browser and enter ftp://localhost

## ii. Enable individual logins and add FTP users with Read- only access.

Open terminal using Ctrl+Alt+T

Open the configuration file using nano editor

```
#sudo nano /etc/vsftpd.conf
```

**anonymous\_enable=No**

**# Uncomment this to allow local users to log in.**

**local\_enable=YES**

**# Uncomment this to enable any form of FTP write command.**

**write\_enable=No**

To exit from nano editor press (Ctrl+x) and then Enter

To restart vsftpd service

```
#sudo service vsftpd start
```

**To check Open browser and enter ftp: //localhost**

**Enrter ur system users username and password (this will browse all the files under the home directory of that user)**

## iii. Transfer Files.

Open terminal using Ctrl+Alt+T  
Open the configuration file using nano editor

```
#sudo nano /etc/vsftpd.conf
anonymous_enable=No
# Uncomment this to allow local users to log in.
local_enable=YES
# Uncomment this to enable any form of FTP write command.
write_enable=Yes
```

To exit from nano editor press (Ctrl+x) and then Enter

### **Conclusion:**

---

---

---

### **Frequently Asked Questions:**

1. What is file transfer protocol?

---

---

---

2. How much data can be send in a Single FTP session

---

---

---

3. Can an IP Packet carry FTP data?

---

---

---

4. What is localhost?

---

---

---

5. Explain security concerns of FTP?

---

---

---

## EXPERIMENT NO: 10

Name of the Student:-

Roll No. \_\_\_\_\_ Subject:-

Date of Practical Performed:- \_\_\_\_\_ Staff Signature with Date  
& Marks

--

**Aim:** Perform Remote Login using Telnet server.

- i) To install and configure TELNET server.

### Theory:

Telnet is a protocol used on the Internet or local area networks to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection. User data is interspersed in-band with Telnet control information in an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP).

Telnet is a user command and an underlying TCP/IP protocol for accessing remote computers. Through Telnet, an administrator or another user can access someone else's computer remotely. On the Web, HTTP and FTP protocols allow you to request specific files from remote computers, but not to actually be logged on as a user of that computer. With Telnet, you log on as a regular user with whatever privileges you may have been granted to the specific application and data on that computer.

### To install and configure TELNET server.

1. Install telnet use this command in terminal (Applications/Accessories/Terminal):

```
#sudo apt-get install xinetd telnetd
```

2. Edit /etc/inetd.conf using your favorite file editor with root permission, add this line:

```
telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
```

3. Edit /etc/xinetd.conf, make its content look like following:

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
# Please note that you need a log_type line to be able to use log_on_success
# and log_on_failure. The default is the following :
```

```
# log_type = SYSLOG daemon info
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
```

4. You can change telnet port number by edit /etc/services with this line:

```
telnet
23/tcp
```

5. If you're not satisfied with default configuration. Edit etc/xinetd.d/telnet, add following:

```
# default: on
# description: The telnet server serves telnet sessions; it uses
# unencrypted username/password pairs for authentication.
service telnet
{
    disable = no
    flags = REUSE
    socket_type = stream
    wait = no
    user = root
    server = /usr/sbin/in.telnetd
    log_on_failure += USERID
}
add these lines as you like:
only_from = 192.168.120.0/24 #Only users in 192.168.120.0 can access to
only_from = .bob.com #allow access from bob.com
no_access = 192.168.120.{101,105} #not allow access from the two IP.
access_times = 8:00-9:00 20:00-21:00 #allow access in the two times
```

6. Use this command to start telnet server:

```
sudo /etc/init.d/xinetd restart
for remote login use following syntax
#telnet user@host / (Ip Address of Host m/c)
```

## **Conclusion:**

---

---

---

## **Frequently Asked Questions:**

1. What is telnet?

---

---

---

2. What is the use of telnet server?

---

---

---

3. What is the use of port number?

---

---

---

4. Explain: Remote Login

---

---

---