

National Institute of Technology, Raipur

Department of Computer Science and Engineering



Lab Report

Computer Network Lab

CS104401CS

Autumn 2021

Submitted by:	J. Varun Iyer
Faculty name:	Ms. Rakhi Seth
Semester:	IV
Roll no.:	19115037

Index

S. no.	Date of experiment	Name of experiment	Pg. no.	Date of submission	Remarks
1.		Introduction to Local Area Network with its cables, connectors and topologies.	4	21/04/21	
2.		Installation of Switch. Hub their cascading and network mapping.	6	21/04/21	
3.		Installation of UTP, Co-axial cable, Cross cable, parallel cable NIC and LAN card.	7	21/04/21	
4.		Case Study of Ethernet (10 base 5,10 base 2,10 base T)	11	21/04/21	
5.		Installation and working of Net meeting and Remote Desktop.	13	21/04/21	
6.		Installation and working with Telnet (Terminal Network).	16	21/04/21	
7.		Installation and working with FTP (File Transfer Protocol).	18	21/04/21	
8.		Installation and Computers via serial or Parallel ports and enable the computers to share disk and printer port.	21	21/04/21	
9.		Installation of NS-2/3 Network Simulator: Basics of Network Simulation	23	21/04/21	
10.		Simulating a Local Area Network and LAN topologies.	25	21/04/21	
11.		Implementation of various MAC protocols.	28	21/04/21	
12.		Measuring Network Performance: Network Performance Evaluation, Performance Evaluation Metrics.	31	21/04/21	
13.		Performance Evaluation of routing Protocol.	32	21/04/21	
14.		Parameter Affecting the Performance of Network, Performance Evaluation Technique, Network Performance Evaluation using NS-2/3	34	21/04/21	
15.		Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.	35	21/04/21	

S. no.	Date of experiment	Name of experiment	Pg. no.	Date of submission	Remarks
16.		Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.	37	21/04/21	
17.		Write a program for error detecting code using CRC (16- bits)	40	21/04/21	
18.		Write a program to find the shortest path between vertices using bellman-ford algorithm.	43	21/04/21	
19.		Write a program for simple RSA algorithm to encrypt and decrypt the data.	46	21/04/21	
20.		Write a program for congestion control using leaky bucket algorithm.	48	21/04/21	

Experiment-1

Aim: Introduction to Local Area Network with its cables, connectors and topologies.

Theory:

LAN(Local Area Network)- Local Area Network is a group of computers connected to each other in a small area such as building, office. LAN is used for connecting two or more personal computers through a communication medium such as twisted pair, coaxial cable, etc. It is less costly as it is built with inexpensive hardware such as hubs, network adapters, and ethernet cables. The data is transferred at an extremely faster rate in Local Area Network. Local Area Network provides higher security.

LAN cables:

1. Twisted-pair cables: The twisted-pair cable was primarily developed for computer networks. This cable is also known as **Ethernet cable**. Almost all modern LAN computer networks use this cable. This cable consists of color-coded pairs of insulated copper wires. Every two wires are twisted around each other to form pair. Usually, there are four pairs.
2. Fiber optic cable: This cable consists of core, cladding, buffer, and jacket. The core is made from the thin strands of glass or plastic that can carry data over the long distance. The core is wrapped in the cladding; the cladding is wrapped in the buffer, and the buffer is wrapped in the jacket.

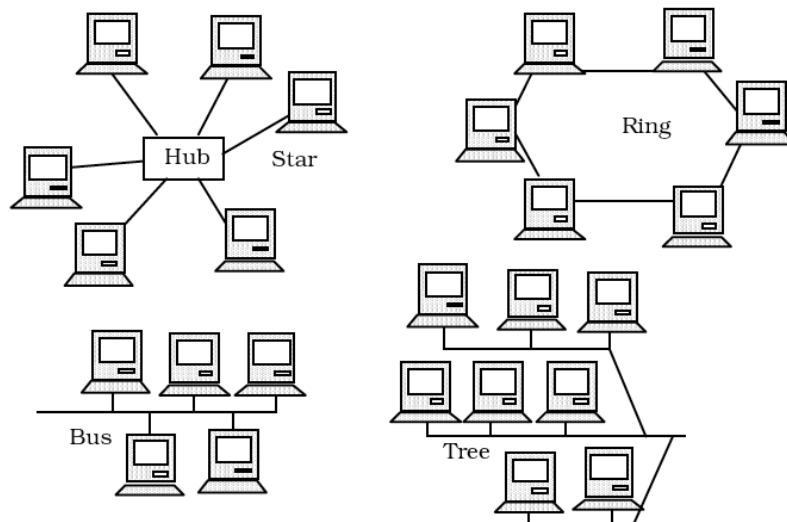
LAN connectors: Network cable connectors such as RJ-45, RJ-11, USB, MT-RJ, Coaxial BNC, LC Local Connector, MT-RJ, USB BNC and AUI are used to connect network cables and form a LAN.

1. USB- Universal Serial Bus, or USB, is a computer standard designed to eliminate the guesswork in connecting peripherals to a PC. It is expected to replace serial and parallel ports. A single USB port can be used to connect up to 127 peripheral devices, such as mice, modems, keyboards, digital cameras, printers, scanners, MP3 players and many more.
2. Registered Jack-11: Standard telephone cable connectors, **RJ-11** has 4 wires (and RJ-12 has 6 wires). **RJ-11** is a four- or six-wire connector primarily used to connect telephone equipment.
3. F-Type: The **F connector** is a type of RF connector commonly used for cable and universally for satellite television. They are also used for the cable TV connection in DOCSIS cable modems, usually with RG-6 tri-shield cable. The F connector is inexpensive, yet has good performance up to 1 GHz.
4. Fiber LC (Local Connector): These connectors are used for single-mode and multimode fiber-optic cables. FC connectors offer extremely precise positioning of the fiber-optic cable with respect to the transmitter's optical source emitter and the receiver's optical detector. FC connectors feature a position locatable notch and a threaded receptacle.

5. **MT-RJ (Mechanical Transfer Registered Jack):** **MT-RJ** connectors are used with single-mode and multimode fiber-optic cables. The **MT-RJ** connectors are constructed with a plastic housing and provide for accurate alignment via their metal guide pins and plastic ferrules.

Four LAN topologies exist:

1. **Star (Hub-and-Spoke) Topology:** All stations are attached by cable to a central point, usually a wiring hub or other device operating in a similar function. Several different cable types can be used for this point-to-point link, such as shielded twisted-pair (STP), unshielded twisted-pair (UTP), and fiber-optic cabling. Wireless media can also be used for communications links.
2. **Ring Topology:** All stations in a ring topology are considered repeaters and are enclosed in a loop. Unlike the star topology, a ring topology has no end points. The repeater in this case is a function of the LAN-attached station's network interface card (NIC).
3. **Bus topology:** Bus is a simple design that utilizes a single length of cable, also known as the medium, with directly attached LAN stations which all stations share. Ethernet (IEEE 802.3) best represents this topology. Ethernet has the ability to utilize many different cable schemes.
4. **Tree Topology:** The tree topology is a logical extension of the bus topology and could be described as multiple interconnected bus networks. The physical (cable) plant is known as a branching tree with all stations attached to it. The tree begins at the root, the pinnacle point, and expands to the network endpoints. An example of a tree topology network is a bridged or switched network running the spanning tree algorithm, usually found with Ethernet (IEEE 802.3) networks.



Experiment-2

Aim: Installation of Switch, Hub- their cascading and network mapping.

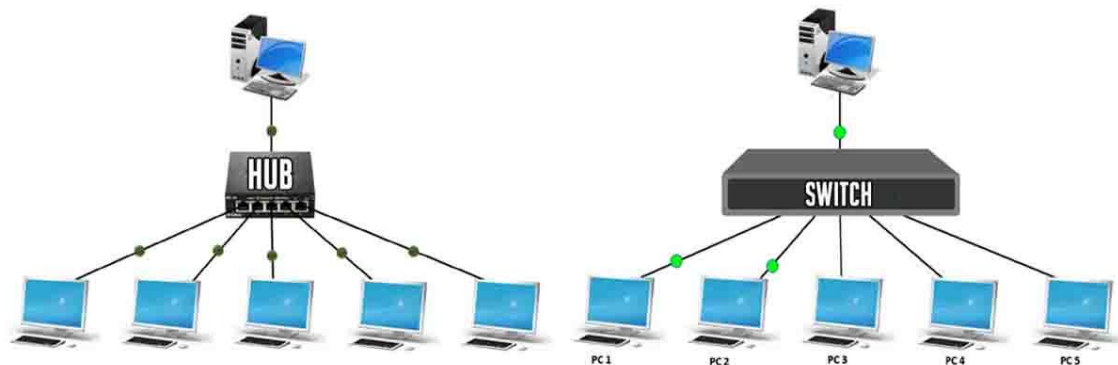
Theory:

Hubs: A hub is a physical layer networking device which is used to connect multiple devices in a network. They are generally used to connect computers in a LAN. A hub has many ports in it. A computer which intends to be connected to the network is plugged in to one of these ports. When a data frame arrives at a port, it is broadcast to every other port, without considering whether it is destined for a particular destination or not.

Switches: A switch is a data link layer networking device which connects devices in a network and uses packet switching to send and receive data over the network. Like a hub, a switch also has many ports, to which computers are plugged in. However, when a data frame arrives at any port of a network switch, it examines the destination address and sends the frame to the corresponding device(s). Thus, it supports both unicast and multicast communications.

Installation of Switch:

- Connect devices via LAN cable with the hub/switch.
- Assign IP addresses(IP Config).
- Check Connectivity(Ping ip_address).



DIFFERENCE BETWEEN HUB AND SWITCH

Experiment-3

Aim: Installation of UTP, Co-axial cable, Cross cable, parallel cable NIC and LAN card.

Theory:

UTP

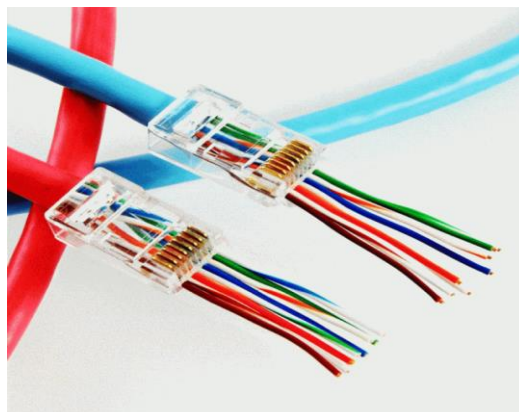
- UTP stands for Unshielded Twisted Pair cable. It is basically a 100 ohm copper cable that consists of 2 to 1800 unshielded twisted pairs surrounded by an outer jacket.
- It is called as unshielded as it has no metallic shield.
- It is small in diameter (because of the lack of shield) but is also unprotected against electrical interference.
- The twisted cables improve its immunity to electrical noise and EMI.

Categories of UTP:

Category	Speed	Use
1	1 Mbps	Voice only
2	4 Mbps	LocalTalk and Telephone
3	16 Mbps	10BaseT Ethernet
4	20 Mbps	Token Ring
5	100 Mbps (2 pair)	100BaseT Ethernet
	1000 Mbps (4 pair)	Gigabit Ethernet
5e	1,000 Mbps	Gigabit ethernet
6	10,000 Mbps	Gigabit Ethernet

UTP cable connectors: RJ-45

- RJ-45 (Registered Jack-45) is an eight wire connector used to connect computers to a LAN and mostly with Ethernet cables.
- It is a standard connector which features eight pins to which the wire strands of cable interface electrically.
- RJ-11 closely resemble it and is slightly narrower, used for telephone cables.



Coaxial Cable:

They are commonly called as coax, are copper cables with metal shielding that provide immunity against noise and a greater bandwidth. It can transmit signals over a large distance at a higher speed if compared to the twisted pair cable.

Structure of coaxial cable:

It has a central core of copper which transmits the signals. It is covered by an insulating material. This insulator is again woven into a outer metal conductor that acts as a shield for noise, which is then coated with a insulating cover.

Categories of coaxial cables:

The cables are categorixed into three categoriesas per radio Government (RG) readings:

- 1) RG-59; It has impedance of 75 W and used in cable TV.
- 2) RG-58; It has impedance of 50 W and used in thin Ethernet cables.
- 3) RG-11: It has impedance of 50 W and used in thick ethernet cables.

Applications of coaxial cables:

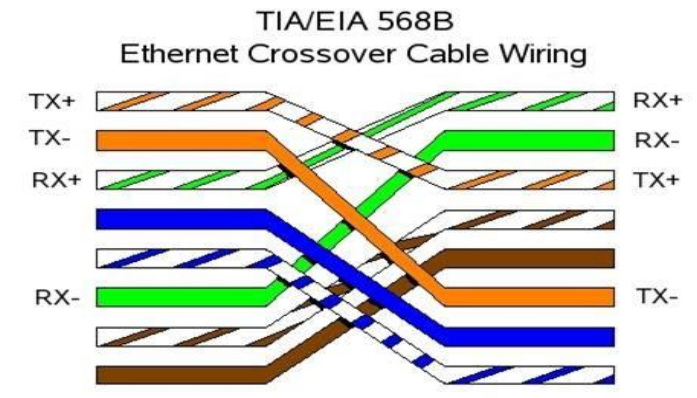
- They are used in analog telephone networks and can carry upto 10,000 voice signals.
- They can be used in digital telephone networks with a speed upto 600 Mbps.
- These cables can be used in traditional Ethernet LANs and MANs.
- They can be used in cable TV networks.



Ethernet cross-over cables:

- It is used for connecting two Ethernet network devices directly without a switch or router in between by connecting the transmitting pins of one side by receiving pins of other side.
- They can send and receive data by enabling complex data transfers,
- They have a crisscross pair of wires; which reverses the incoming and outgoing signals.
- It can be used for connecting:
 - Two computers
 - Two hubs

- A hub to a switch
- A cable modem and a router
- Two router interfaces



NIC:

NIC stands for Network Interface Card is a hardware component that is a prerequisite for the computer to be connected to the internet. It is basically a circuit board installed in a computer that provides dedicated network connection to the computer. It is also called as Network Interface controller/Network Adapter/LAN Adapter/LAN Card.

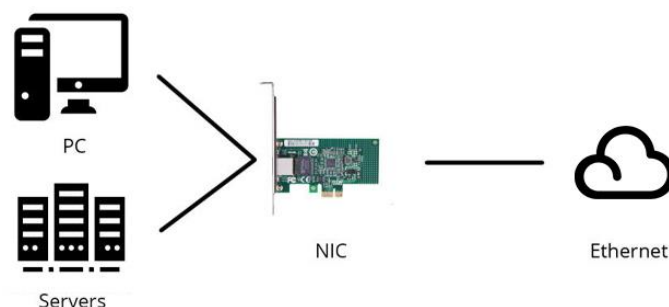
It is of two types:

Internal network cards: They are inserted in a predefined slot of motherboard and requires network cables.

External network cards: If there is no internal NIC in desktops or laptops, then it is used and can be wither wireless or USB type.

Purpose of NIC:

- It allows both wired and wireless communication.
- It is in both physical and data link layer.
- It allows communications between computer connected through LANs as well as communications over large-scale network through Internet Protocol (IP).



Procedure:

The following steps discuss the procedure to make your own Ethernet cable:

Materials Required:

- Unshielded Twisted Pair (UTP) Patch cable

- Modular connector (8P8C plug, aka RJ45)
- Crimping tool
- Cable tester (optimal, but recommended)

Steps:

- 1) Strip the jacket of cable up to about 1.5 inches down from the end.
- 2) Spread the four pairs of twisted wire apart, for category 5e pull string can be used.
- 3) Untwist the wire pairs and neatly align them in the T568B orientation.
- 4) Cut the wires now (as straight as possible) about 0.5 inches above the end of the jacket.
- 5) Insert the wires all the way into the modular connector carefully and make sure that each wire passes through the appropriate guides inside the connector.
- 6) Push the connector inside the crimping tool and squeeze it all the way down.
- 7) Repeat the steps 1-6 for the other end of the cable too.
- 8) Use a cable tester for each pin to make sure that you have terminated each end of the cable successfully.

Conclusion:

In this experiment we made our own ethernet cable and also got an idea about the UTP, Coaxial cables, Cross cable and NIC or the LAN card.

Experiment-4

Aim: Case Study of Ethernet (10 base 5,10 base 2,10 base T).

Theory:

Ethernet is a family of frame-based computer networking technologies for local area networks (LANs). The name comes from the physical concept of the ether. It defines a number of wiring and signaling standards for the physical layer, through means of network access at the Media Access Control (MAC)/Data Link Layer, and a common addressing format. Ethernet is standardized as IEEE 802.3. The combination of the twisted pair versions of Ethernet for connecting end systems to the network, along with the fiber optic versions for site backbones, is the most widespread wired LAN technology.

10 base 5- 10 base 5(also known as thicknet) is the original "full spec" variant of Ethernet cable, using special cable similar to RG-8/U coaxial cable. This is a stiff, 0.375-inch (approx. 9.5 mm) diameter cable with an impedance of 50 ohms, a solid center conductor, a foam insulating filler, a shielding braid, and an outer jacket. The outer sheath is often yellow-to-orange/brown foam fluorinated ethylene propylene (for fire resistance) so it frequently is just called "yellow cable", "orange hose", or sometimes humorously "frozen yellow garden hose". 10BASE5 cable is designed to allow transceivers to be added while existing connections are live. This is achieved using a *vampire tap* - a device which (with sufficient practice) clamps onto the cable, forcing a spike to pierce through the outer shielding to contact the inner conductor while other spikes bite into the outer conductor. This is often built into the transceiver and a more flexible multi-wire cable carries the connection between the transceiver and the node. Transceivers can also be connected by using N connectors at the end of a cable segment.



10 base 2- 10 base 2 (also known as cheapernet, thin ethernet, thinnet or thinwire) is a variant of Ethernet that uses thin coaxial cable (RG-58 or similar, as opposed to the thicker RG-8 cable used in 10BASE5 networks), terminated with BNC connectors. For many years this was the dominant 10 Mbit/s Ethernet standard, but due to the immense demand for high speed networking, the low cost of Category 5 Ethernet cable, and the popularity of 802.11 wireless networks, both 10BASE2 and 10BASE5 have become almost obsolete. In a 10BASE2 network, each segment of cable is connected to the transceiver (which is usually built into the network adaptor) using a BNC T-connector, with one segment connected to each arm of the T. At the physical end of the network a 50 Ohm terminator is required. This is most commonly connected directly to the T-connector on a workstation though it does not technically have to be.



10 base-2



10 base-T

10 base-T- There are several standards for Ethernet over twisted pair or copper-based computer networking physical connectivity methods. The currently most widely used of these are 10BASE-T, 100BASE-TX, and 1000BASE-T (Gigabit Ethernet), running at 10 Mbit/s, 100 Mbit/s, and 1000 Mbit/s (1 Gbit/s) respectively. These three standards all use the same connectors. Higher speed implementations nearly always support the lower speeds as well, so that in most cases different generations of equipment can be freely mixed. They use 8 position modular connectors, usually (but incorrectly) called RJ45 in the context of Ethernet over twisted pair. The cables usually used are four-pair Category 5 or above twisted pair cable. Each of the three standards support both full duplex and half-duplex communication. According to the standards, they all operate over distances of 'up to 100 meters'.

Experiment-5

Aim: Installation and working of NetMeeting and Remote Desktop.

NetMeeting:

- Microsoft Windows NetMeeting is used for real time PC-to-PC screen sharing and remote control of Windows PC's across the network.
- It can be used in parallel with the main video conferencing system so that any site connected to it can share the screen or application of other sites.
- It was included with Microsoft Windows versions 95 OSR2 to Windows XP.

Installation of NetMeeting:

- 1) Download the NetMeeting software from Microsoft free of cost.
- 2) Install it, while installing it asks whether we would want to log on to a directory server when NetMeeting starts.
- 3) If we select this option others will be able to see our name (which might result in unsolicited callers).
- 4) Once the program is installed, launch it and finish the configuration.

Activating remote desktop sharing:

- 1) Launch NewMeeting.
- 2) Choose **Tools-> Enable Sharing** and follow the instructions.
- 3) Reboot the computer.
- 4) After rebooting, launch NewMeeting.
- 5) Choose **Tools->Remote Desktop Sharing** and follow instructions on screen.
- 6) Close NetMeeting.

Running a remote session:

- 1) Launch NewMeeting.
- 2) Click on **Place Call** button.
- 3) On the top bar types the IP address or name of the remote computer.
- 4) If **Require Security for this call** check box isn't selected, select it to turn on security.

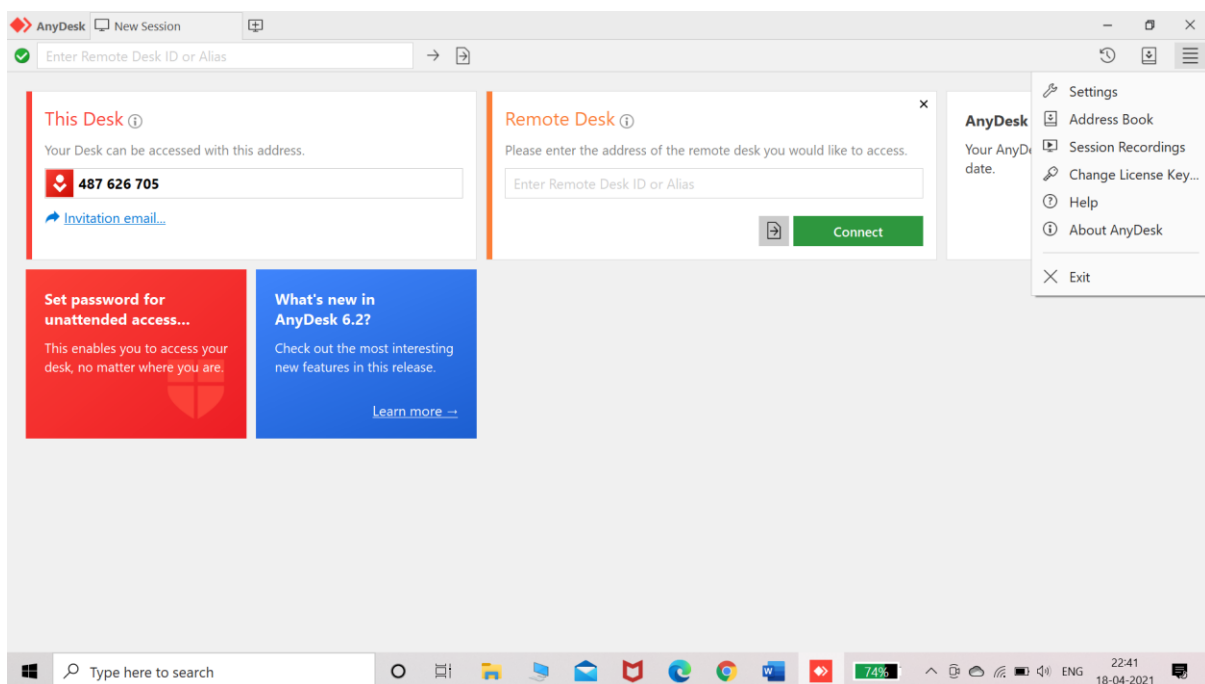


Anydesk:

It is a remote desktop software that can be run like a normal program. It ensures secure remote desktop connections for IT professionals.

Features:

- Supports audio and file transfer.
- Connections can be configured to favor quality or speed.
- Works on Linux, MacOS, Windows.
- Feature of recording the remote session.
- Auto updates for new versions.



TeamViewer:

It is a remote desktop software allowing to connect workstations remotely. It enhances the remote control performance with GPU for hardware accelerated image processing.

Features:

- Allow you to pass the control from one person to other.
- It supports Augmented reality.
- Multiple parties can be included in connection.
- Files can be dragged and dropped from one computer to another.
- Easy and secure.

ConnectionExtrasHelpFeedback

Insert partner IDConnect

Free license (non-commercial use only) - Sanjay Tiwari

Sign In
Don't have an account?

Remote Control

Remote Management

Meeting

Computers & Contacts

Chat

Augmented RealityNEW

More SolutionsNEW

TeamViewer

Allow Remote Control

Your ID

985 202 218

Password

u3f2ttus

Unattended Access

☐ Start TeamViewer with Windows

☐ Grant easy access

Control Remote Computer

Partner ID

☒ Remote control

☐ File transfer

Connect

Ready to connect (secure connection)

Experiment-6

Aim: Installation and working with Telnet (Terminal Network).

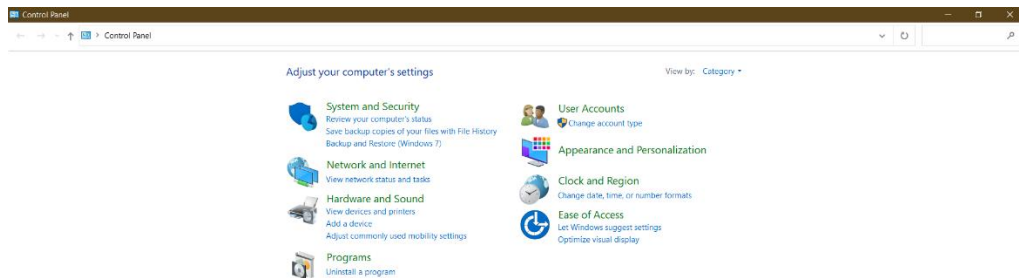
Telnet:

- It is a popular client-server program used to provide a general client-server environment where the user can access any application program of remote computer.
- It stands for Terminal Network.
- It provides connection to remote computer in a way that local terminal appears to be at remote side.

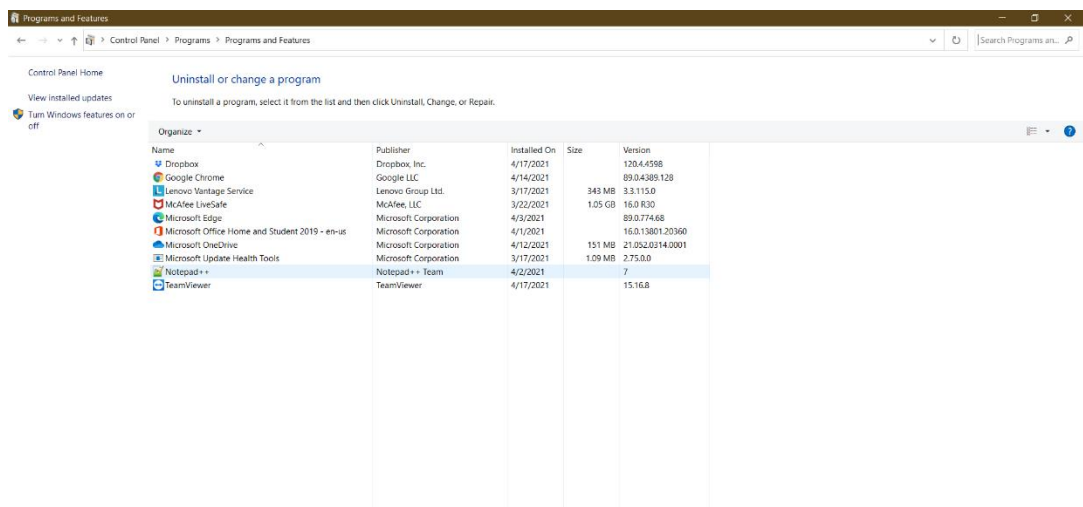
Installation of Telnet on Windows:

Unlike Linux Telnet is not preinstalled in on Windows, it can be installed by following steps:

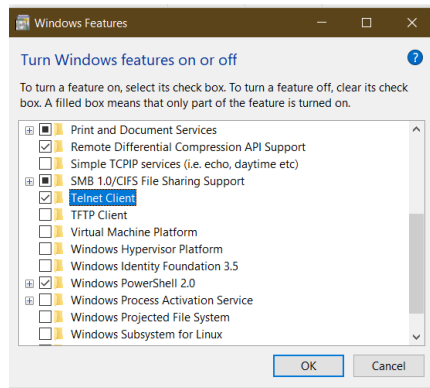
- 1) Click on **Start**.
- 2) Select **Control Panel**.



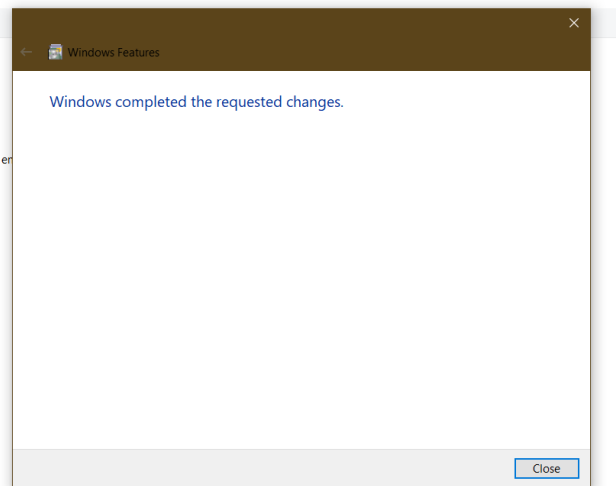
- 3) Choose **Program and Features**.



- 4) Click **Turn Windows features on or off**.
- 5) Select the **Telnet Client** option.



- 6) Click OK.
- 7) A dialogue box appears to configure the installation. The Telnet commands are then available.



Experiment-7

Aim: Installation and working with FTP (File Transfer Protocol).

FTP:

File Transfer protocol server can be set up on Windows to upload and download files from virtually anywhere to the computer without the limitations of the cloud storage. It creates a private cloud without any file or size restrictions and can be used for mutiple accounts. Thus, it proves to be very useful personally and proffessionally.

Installation of FTP server components:

1. The required components of FTP can be added manually.
2. Open *Control Panel*.
3. Click on *Programs*.
4. Under *Programs and features* click the *Turn Windows features on or off*.
5. When the dialog box opens, expand the *Internet Information Services* feature and expand *FTP server* option.
6. Check the *FTP Extensibility* and *FTP Service* option.
7. Check the *Web Management Tools* option with the default selections, but making sure that the *IIS Management Console* option is checked.
8. Click the OK button.
9. Click the Close button.

Configure FTP server site on Windows 10

After installing the required components, configure an FTP server on the computer, which involves creating a new FTP site, setting up firewall rules, and allowing external connections.

Setting up an FTP site

- 1) Open *Control Panel*.
- 2) Click on *System and Security*.
- 3) Click on *Administrative Tools*.
- 4) Double-click *the Internet Information Services (IIS) Manager* shortcut.
- 5) On the *Connections* pane, right-click *Sites*, and select the *Add FTP Site* option.
- 6) In the FTP site name, type a short name for your server.
- 7) In the *Content Directory section*, under *Physical path* click the button on the right to locate the folder you want to use to store your FTP files.
- 8) Click the Next button.
- 9) Use the default *Binding settings* selections.
- 10) Check the *Start FTP site automatically* option.
- 11) In the "SSL" section, check the *No SSL* option
- 12) Click the Next button.
- 13) In the *Authentication* section, check the *Basic* option.

- 14) In the *Authorization* section, use the drop-down menu, and select *Specified users* option.
- 15) Type the email address of your Windows 10 account or local account name to allow yourself access to the FTP server.
- 16) Check the *Read and Write* options.
- 17) Click on the *Finish* button.

FTP is now operational on the computer.

Configuring Firewall Rules:

The services should be manually allowed as the built-in firewall on Windows-10 will block the FTP server.

- 1) Open *Windows Defender Security Center*.
- 2) Click on *Firewall & network protection*.
- 3) Click the *Allow an app through firewall* option.
- 4) Click the *Change settings* button.
- 5) Check the *FTP Server* option, as well as the options to *allow Private and Public access*.

Allowing External Connections:

For reaching external servers, TCP/IP port number 21 has to set up.

- 1) Open *Settings*.
- 2) Click on *Network & Internet*.
- 3) Click on *Status*.
- 4) Click the *Change connections properties* option.
- 5) Make a note of the IPv4 DNS server address, which is the address of your router. Usually, it's private address in the 192.168.x.x range. For instance, 192.168.1.1 or 192.168.2.1.
- 6) Open your default web browser.
- 7) On the address bar enter the router's IP address and press Enter.
- 8) Sign-in with your router credentials.
- 9) Open the *Port Forwarding* page.
- 10) Add a new rule to forward incoming connections to the FTP server from the internet by including this information:
 - Service name: Type a descriptive name for the port forwarding rule.
 - Port range: 21.
 - Local IP: This is the FTP server IP address that the router will forward incoming connections.
 - Local port: 21.
 - Protocol: TCP
- 11) Click the add button.
- 12) Click the Apply button to save the changes.

After completing the steps, any incoming connection on port 21 will be forwarded to the FTP server to establish a networking session

Setting up static IP address:

- 1) Open *Control Panel*.
- 2) Click on *Network and Internet*.
- 3) Click on *Network and Sharing Center*.
- 4) In the left pane, click the *Change adapter settings* option.
- 5) Right-click the network adapter, and select the *Properties* option.
- 6) Select the *Internet Protocol Version 4 (TCP/IPv4)* option.
- 7) Click the *Properties* button.
- 8) Select the *Use the following IP address* option.
- 9) Specify the IP settings:
 - IP address: Specify a static network address for the computer.
 - Subnet mask: In a home network, the address usually is 255.255.255.0.
 - Default gateway: This is usually the IP address of the router.
 - Preferred DNS server: Typically, this is also the IP address of your router.

Experiment- 8

Aim: Installation and Computers via serial or Parallel ports and enable the computers to share disk and printer port.

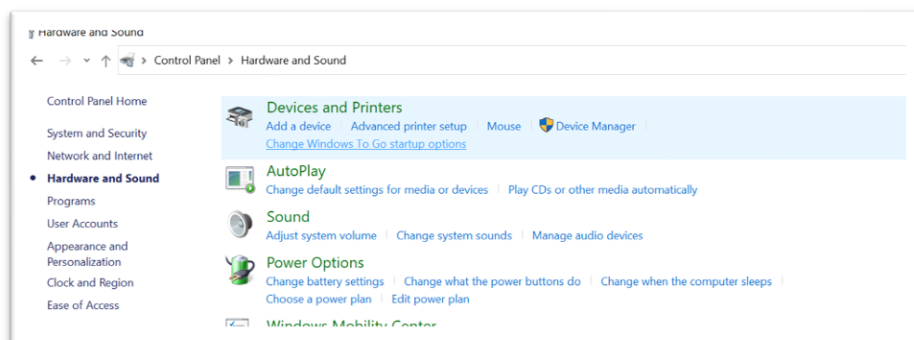
Theory: A port is a point at which an external device (peripheral) attaches to the computer system. Ports allow data to be sent/retrieved from the external device.

The most common type of ports are:

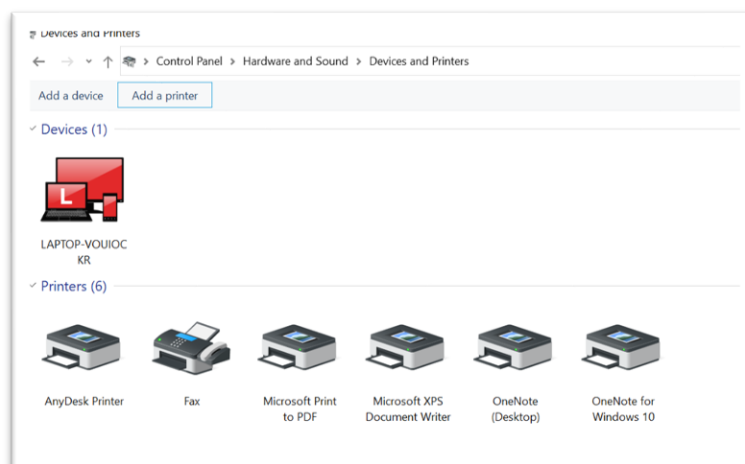
- a. Serial port - Serial ports transmit data sequentially one bit at a time. Serial ports are usually 9-pin or 25-pin male connectors. They are also known as COM (communication) ports or RS323C ports.
- b. Parallel port- Parallel ports can send or receive 8 bits or 1byte at a time. Parallel ports come in form of 25-pin female pins and are used to connect printer, scanner, external hard disk drive, etc.

Adding a Network Printer to Your Windows Computer-

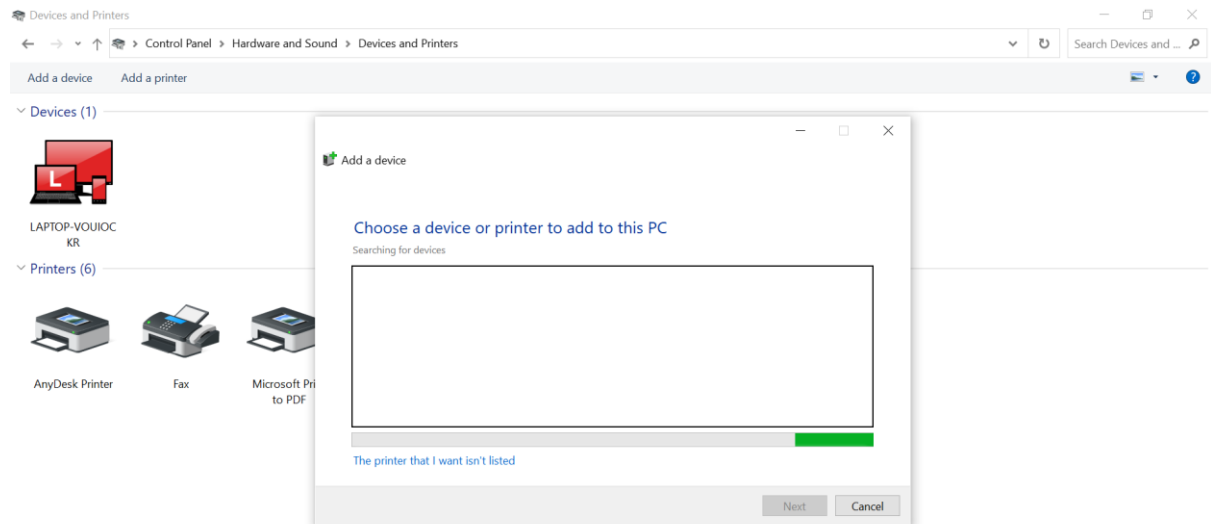
1. Click on the Start button, and then select Devices and Printers.



2. In the Devices and Printers window, click on Add a printer.



3. In the Add Printer window, click on the option Add a local printer.



4. Select Create a new port, and then select Standard TCP/IP Port from the drop-down menu. Click Next when you're done.

5. Enter the IP address of your printer. . Select the option Query the printer and automatically select the driver to use. Click Next when you're done.

Note: If the printer is not automatically detected, please select the Manufacturer and Driver.6. Select the option Use the driver that is currently installed (recommended).

7. Enter a descriptive name for the printer in the text box next to Printer name. (In the example below, "Harris 2nd Floor Break Room" is entered as the printer name.)

9. Select the option Do not share this printer and then click Next.

10. If the printer was added successfully, you should see the following confirmation message. Click Print a test page to confirm that you can print to the printer.

11. Click Close in the following window.

12. Click Finish.

13. If you want, you can view the new printer listing by clicking the Start button and then clicking on Devices and Printers.

Experiment- 9

Aim: Installation of NS-2/3 Network Simulator: Basics of Network Simulation

NS2: NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Features of NS2:

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support
7. Tclcl: C++ and otcl linkage
8. Discrete event scheduler

“Ns” Components:

- Ns, the simulator itself
- Nam, the network animator
- Visualize ns (or other) output
- Nam editor: GUI interface to generate ns scripts
- Pre-processing: Traffic and topology generators (use Tcl to write)
- Post-processing:

Simple trace analysis, often in Awk, Perl, or Tcl

You can also use grep (under linux), or C/java

Installation of NS2:

1. Update and install prerequisites packages using these commands:

```
$sudo apt-get update
```

```
$sudo apt-get dist-upgrade
```

```
$sudo apt-get update
```

```
$sudo apt-get gcc
```

```
$sudo apt-get install build-essential autoconf automake
```

```
$sudo apt-get install tcl8.5-dev tk8.5-dev
```

```
$sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev
```

2. Download ns-allinone-2.35.tar.gz.
3. Select the file and click the right mouse button and choose option extract here.
4. After extracting type on the command prompt.

\$cd ns-allinone-2.35

5. Run Command ./install see in the image. Step-6 Set the path of the NS-2 for global access. Use command

\$sudo gedit ~/.bashrc

It will open a file .bashrc

Set the path

```
# LD_LIBRARY_PATH
OTCL_LIB=/Yourpath/ns-allinone-2.35/otcl-1.14
NS2_LIB=/Yourpath/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:$USR
_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/Yourpath/ns-allinone-2.35/tcl8.5.10/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/Yourpath/ns-allinone-2.35/bin:/Yourpath/ns-allinone-
2.35/tcl8.5.10/unix:/Yourpath/ns-allinone-2.35/tk8.5.10/unix
NS=/Yourpath/ns-allinone-2.35/ns-2.35/
NAM=/Yourpath/ns-allinone-2.35/nam-1.15/
PATH=$PATH:$XGRAPH:$NS:$NAM
```


Experiment- 10

Aim: Simulating LAN and LAN topologies

Instructions for execution:

- To analyze the given problem you have to write a Tcl script and simulate with ns2
- Begin by specifying the trace files and the nam files to be created
- Define a finish procedure
- Determine and create the nodes that will be used to create the topology. Here in our experiment we are selecting 6 nodes namely 0, 1, 2, 3, 4, 5
- Create the links to connect the nodes
- Set up the LAN by specifying the nodes, and assign values for bandwidth, delay, queue type and channel to it
- Optionally you can position and orient the nodes and links to view a nice video output with Nam
- Set up the TCP and/or UDP connection(s) and the FTP/CBR (or any other application) that will run over it
- Schedule the different events like simulation start and stop, data transmission start and stop
- Call the finish procedure and mention the time at what time your simulation will end
- Execute the script with ns

Code:

```
#lan.tcl
#Lan simulation
set ns [new Simulator]
#define color for data flows
$ns color 1 Blue
$ns color 2 Red
#open tracefiles
set tracefile1 [open out.tr w]
set winfile [open winfile w]
$ns trace-all $tracefile1
#open nam file
set namfile [open out.nam w]
$ns namtrace-all $namfile
#define the finish procedure
proc finish {} {
    global ns tracefile1 namfile
    $ns flush-trace
```

```

close $tracefile1
close $namfile
exec nam out.nam &
exit 0
}
#create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box
#create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd
Channel]
#Give node position
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
#set queue size of link(n2-n3) to 20
$ns queue-limit $n2 $n3 20
#setup TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set packet_size_ 552
#set ftp over tcp connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
#setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null

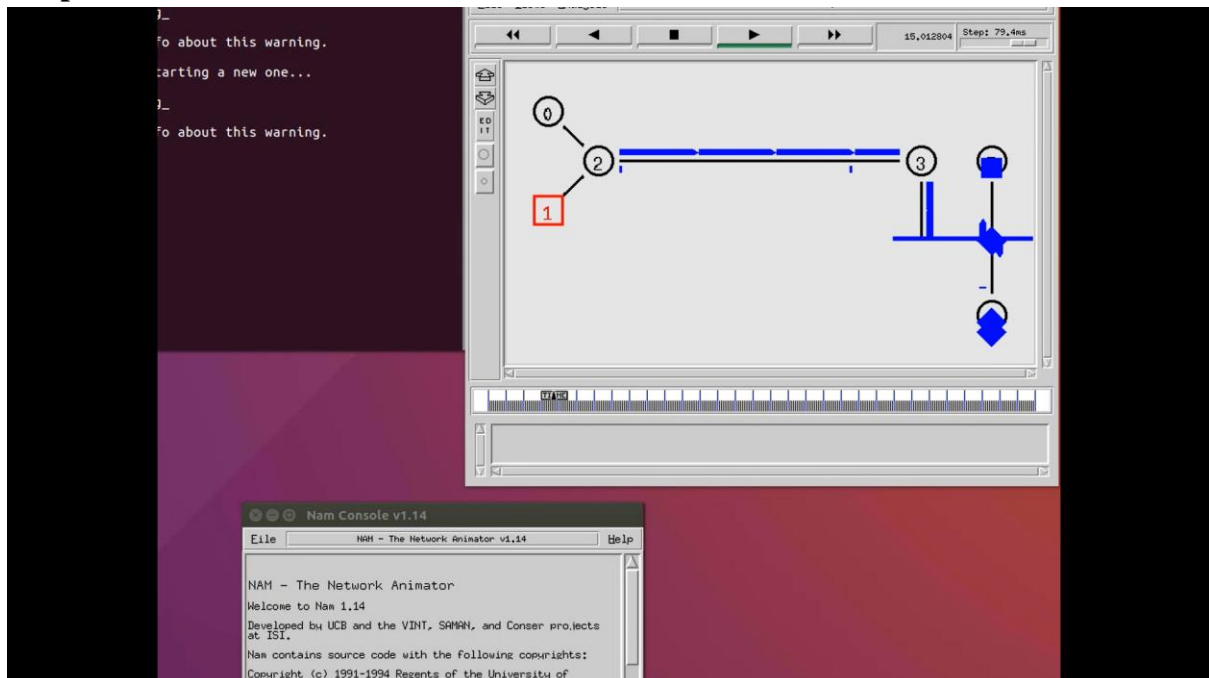
```

```

$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_ 0.01Mb
$cbr set random_ false
#scheduling the events
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 125.5 "$cbr stop"
proc plotWindow {tcpSource file} {
    global ns
    set time 0.1
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $file "$now $cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpSource $file"
} $ns at 0.1 "plotWindow $tcp $winfile"
$ns at 125.0 "finish"
$ns run

```

Output:



Experiment- 11

Aim: Implementation of various MAC protocols.

Instructions:

- Create a simulator object
- Define different colors for different data flows
- Open a nam trace file and define finish procedure then
- close the trace file, and execute nam on trace file.
- Create six nodes that forms a network numbered from 0 to 5
- Create duplex links between the nodes and add Orientation to the nodes for setting a LAN topology
- Setup TCP Connection between n(0) and n(4)
- Apply FTP Traffic over TCP
- Setup UDP Connection between n(1) and n(5)
- Apply CBR Traffic over UDP.
- Apply CSMA/CA and CSMA/CD mechanisms and study their performance
- Schedule events and run the program.

Code:

```
#csma.tcl
set ns [new Simulator]
#Define different colors for data flows (for nam)
$ns color 1 Blue
$ns color 2 red
#Open the Trace files
set file1 [open out.tr w]
set winfile [open Winfile w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
#Define a 'finish' procedure
proc finish {} {
    global ns file1 file2
    $ns flush-trace
    close $file1
    close $file2
}
```

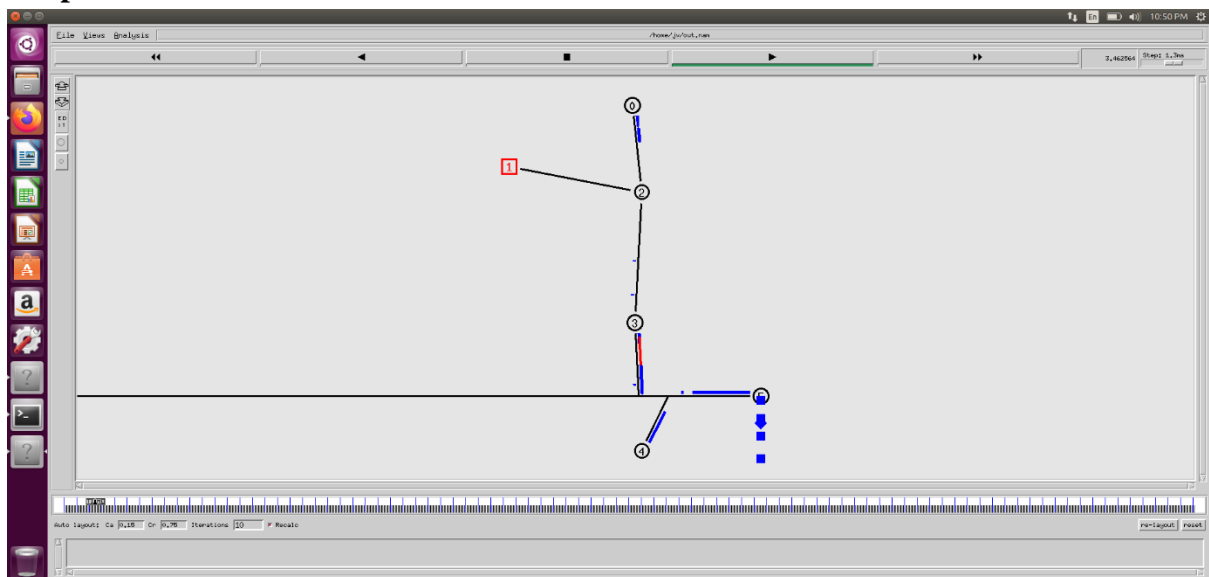
```
exec nam out.nam &
exit 0
}
#create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box
#create link between nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/Droptail MAC/Csma/Channel]
#setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#setup 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 $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#setup a cbr over udp connexion
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
```

```

$cbrr set rate_ 0.01mb
$cbrr set random_ false
$ns at 0.1 "$cbrr start"
$ns at 1.0 "$ftp start"
$ns at 123.0 "$ftp stop"
$ns at 124.5 "$cbrr stop"
#next procedure gets two arguments: the name of tcp source node will be called here
tcp
#and the name of output file
proc plotWindow {tcpSource file} {
  global ns
  set time 0.1
  set now [$ns now]
  set cwnd [$tcpSource set cwnd_]
  set wnd [$tcpSource set window_]
  puts $file "$now $cwnd"
  $ns at [expr $now+$time] "plotWindow $tcpSource $file"}
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 5 "$ns trace-annotate \"packet drop\""
#PPP
$ns at 125.0 "finish"
$ns run

```

Output:



Experiment- 12

Aim: Measuring Network Performance: Network Performance Evaluation, Performance Evaluation Metrics.

Concept:

Network performance Evaluation- Network performance is defined by the overall quality of service provided by a network. Network performance Evaluation encompasses numerous parameters and metrics that must be analysed collectively to assess a given network. Set of processes and tools that can be used to quantitatively and qualitatively assess network performance and provide actionable data to remediate any network performance issues. We measure it to assure user satisfaction & network quality and fix issues if any.

Performance Evaluation Metrics-

<i>Category</i>	<i>Metric</i>	<i>Unit</i>
Productivity	Throughput	Bytes per second
Responsiveness	Delay, Jitter	Seconds
Utilization	Channel Utilization	Percentage of time busy
Loss	Packet drops, Retransmission count	Number
Buffer Space	Queue size, overflow and underflow rate	Bytes

Following are some of the performance measurement metrics:

- **Latency:** It can take a long time for a packet to be delivered across intervening networks. In reliable protocols where a receiver acknowledges delivery of each chunk of data, it is possible to measure this as round-trip time. •**Packet loss:** In some cases, intermediate devices in a network will lose packets. This may be due to errors, to overloading of the intermediate network, or to intentional discarding of traffic in order to enforce a particular service level.

$$\text{Latency} = \text{Propagation Time} + \text{Transmission Time} + \text{Queuing Time} + \text{Processing Delay}$$
$$\text{Propagation time} = \text{Distance} / \text{Propagation speed}$$

- **Retransmission:** When packets are lost in a reliable network, they are retransmitted. This incurs two delays: First, the delay from re-sending the data; and second, the delay resulting from waiting until the data is received in the correct order before forwarding it up the protocol stack.
- **Throughput:** The amount of traffic a network can carry is measured as throughput, usually in terms such as kilobits per second. Throughput is analogous to the number of lanes on a highway, whereas latency is analogous to its speed limit.

Experiment-13

Aim: Performance Evaluation of routing Protocol.

Routing Protocol:

This protocol is a set of rules used by the routers to communicate between the source and destination. Each protocol has its own algorithm to choose path.

There are two kinds of routing:

- 1) *Unicast Routing*: It is a type of information transfer and used when there is a participation of single sender and a single recipient.
- 2) *Multicast Routing*: It used in case of multiple senders and recipients.

Performance Analysis of routing Protocol:

The analysis of routing Protocol can be assessed by factors such as:

- 1) *Packet delivery Ratio*: It is measured as the ratio of number of packets delivered in total to the number of packets sent from the source node to destination node.
- 2) *Energy Consumptions*: Energy consumption is measured as the total energy used to perform the complete process.
- 3) *Throughput*: The number of packets transferred per unit time.
- 4) *Average Delay*: It is the delay per packet transfer from source node to the destination node.

All of the above factors and some others affect the performance of routing protocol and can be used to assess its performance.

Program Code:

The following program shows how an XGraph can be used to plot the bandwidth of two nodes connected through the duplex wired link (An XGraph program draws a graph on an x-display given data read from either data file or standard input):

```
#Create a simulator object
set ns [new Simulator]
#Open the output trace file
set f0 [open out0.tr w]
#Create 2 nodes
set n0 [$ns node]
set n1 [$ns node]
#Connect the nodes using duplex link
$ns duplex-link $n0 $n1 1Mb 100ms DropTail
#Define a 'finish' procedure
proc finish {} {
    global f0
    #Close the output files
    close $f0
```

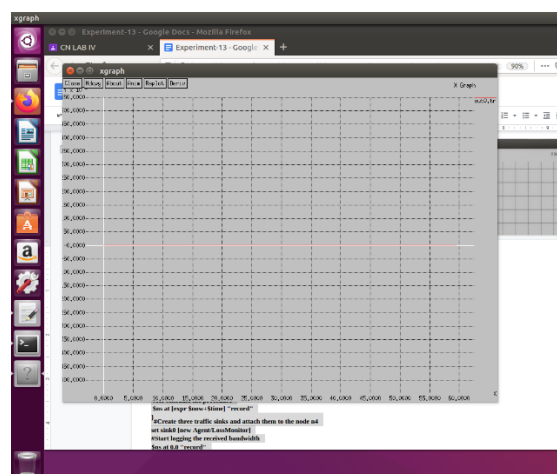


```

#Call xgraph to display the results
exec xgraph out0.tr -geometry 800x400 &
exit 0
}
#Define a procedure which periodically records the bandwidth received by the
proc record { } {
global sink0 f0
#Get an instance of the simulator
set ns [Simulator instance]
#Set the time after which the procedure should be called again
set time 0.5
#How many bytes have been received by the traffic sinks?
set bw0 [$sink0 set bytes_]
#Get the current time
set now [$ns now]
#Calculate the bandwidth (in MBit/s) and write it to the files
puts $f0 "$now [expr $bw0/$time*8/1000000]"
#Reset the bytes_ values on the traffic sinks
$sink0 set bytes_ 0
#Re-schedule the procedure
$ns at [expr $now+$time] "record"
}
#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
#Start logging the received bandwidth
$ns at 0.0 "record"
$ns at 60.0 "finish"
#Run the simulation
$ns run

```

Output:



Experiment- 14

Aim: Parameter Affecting the Performance of Network, Performance Evaluation Technique, Network Performance Evaluation using NS-2/3.

Concept:

Parameters Affecting the Performance of Networks:

- Different parameters can together or independently determine how well a network would perform. A few such are mentioned below:
- Bandwidth: It is the maximum data transfer rate which a link allows. It is expressed in bits per second(bps).
- Propagation Delay: It is the amount of time required for a packet to travel from one node to another. If the propagation delay is high then throughput will be low i.e they are inversely proportional to each other.
- Queue type and queue size: The queue of a node is implemented as a part of a link whose input is that node to handle the overflow at the queue. But if the buffer capacity of the output queue is exceeded then the last packet arrived is dropped. We do set the buffer capacity by using queue size.

Network Performance Evaluation in NS2:

Let's discuss how to evaluate performance of a network by simulating it with ns2.

- Design, develop and generate a network topology to be used within the simulation. This could be a wired network; in which case the topology remains fixed. However, for a wireless network with mobile nodes the topology would change with time, or randomly.
- Once the topology has been generated, traffic source(sender) and destination(receiver) are fixed. Assign suitable traffic sources to the source nodes like tcp, udp and ftp, and traffic sinks like sink, null to the destination nodes.
- Some of the parameters that can be used for comparative study of performance of the network are: link bandwidth, propagation delay, node queue type

Experiment- 15

Aim: Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

Code:

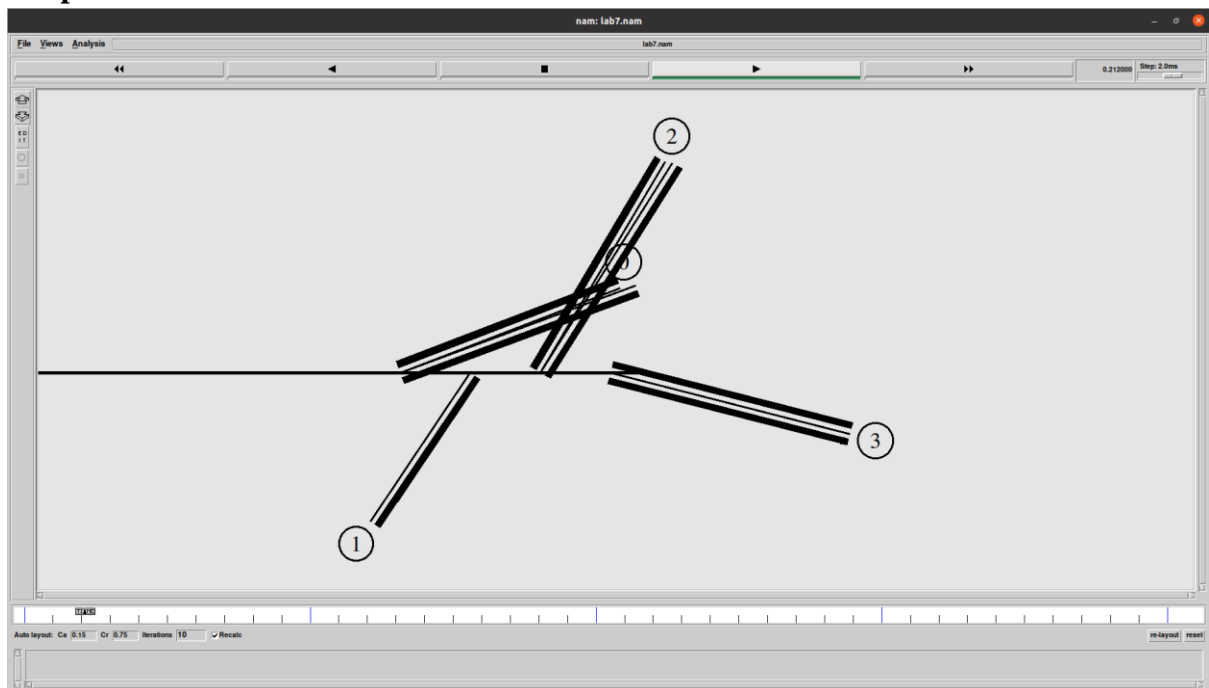
```
set ns [new Simulator]
set tf [open lab7.tr w]
$ns trace-all $tf
set nf [open lab7.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns make-lan "$n0 $n1 $n2 $n3" 10mb 10ms LL Queue/DropTail Mac/802_3
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp0 $sink3
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp2 $sink1
#####To trace the congestion window#####
set file1 [open file1.tr w]
$tcp0 attach $file1
$tcp0 trace cwnd_
$tcp0 set maxcwnd_ 1 0
set file2 [open file2.tr w]
$tcp2 attach $file2
$tcp2 trace cwnd_
proc finish { } {
    global nf tf ns
    $ns flush-trace
    exec nam
    lab7.nam &close $nf
    close $tf
    exit 0
}
$ns at 0.1 "$ftp0 start"
```

```
$ns at 1.5 "$ftp0 stop"
$ns at 2 "$ftp0 start"
$ns at 3 "$ftp0 stop"
$ns at 0.2 "$ftp2 start"
$ns at 2 "$ftp2 stop"
$ns at 2.5 "$ftp2 start"
$ns at 4 "$ftp2 stop"
$ns at 5.0 "finish"
$ns run
```

awk script

```
BEGIN {
}
{
if($6=="cwnd_") /* don't leave space after writing cwnd_ */ printf("%f\t%f\t\n",$1,$7); /*
you must put \n in printf */
END {
}
```

Output:



Experiment- 16

Aim: Implement simple ESS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.

Code:

```
set ns [new Simulator]
set tf [open exp4.tr w]
    $ns trace-all $tf

set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open exp4.nam w]

$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
    -llType LL \
    -macType Mac/802_11 \
    -ifqType Queue/DropTail \
    -ifqLen 50 \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -propType Propagation/TwoRayGround \
    -antType Antenna/OmniAntenna \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON
create-god 3

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"

#The below code is used to give the initial node positions.
$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
```

```
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0
```

```
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
```

```
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
```

```
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
```

```
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
```

```
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
```

```
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
```

```
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
#The below code is used to provide the node movements.
$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
```

```
proc finish {} {
    global ns nf tf
    $ns flush-trace
    exec nam exp4.nam &
    close $tf
    exit 0
}
$ns at 250 "finish"
$ns run
```

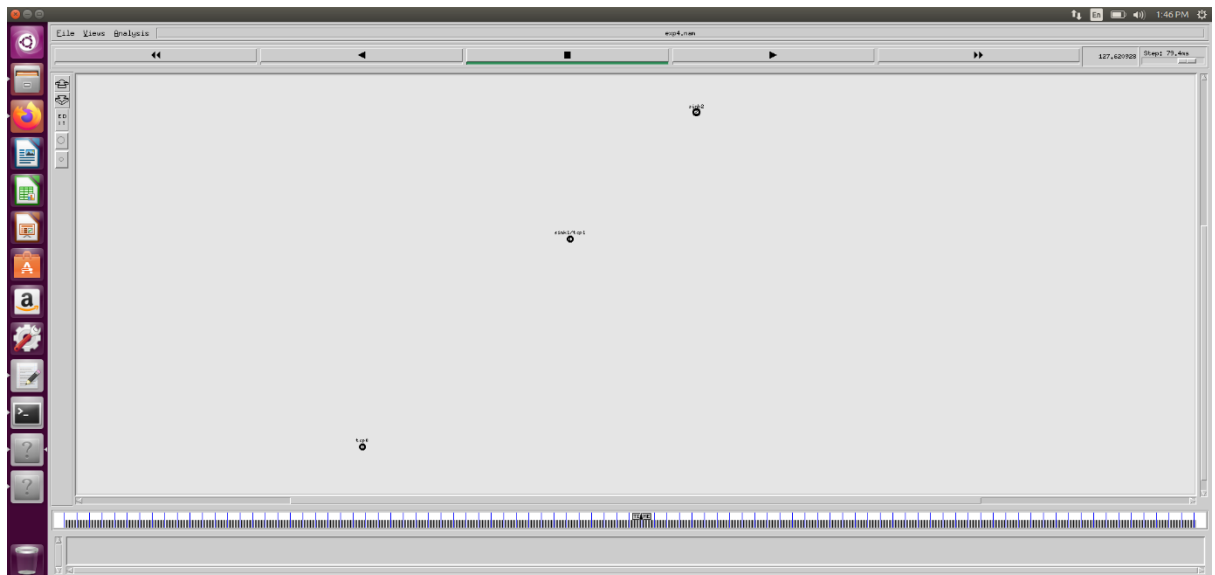
awk script

```
BEGIN{
    #include<stdio.h>
    count1=0
```

```

        count2=0
        pack1=0
        pack2=0
        time1=0
        time2=0
    }
    {
        if($1=="r"&&$3=="_1_"&&$4=="AGT")
        {
            count1++
            pack1=pack1+$8
            time1=$2
        }
        if($1=="r"&&$3=="_2_"&&$4=="AGT")
        {
            count2++
            pack2=pack2+$8
        }
    }
}
END{
    time2=$2
    printf("The Throughput from n0 to n1: %fMbps\n",
    ((count1*pack1*8)/(time1*1000000)))
    printf("The Throughput from n1 to n2: %f
    Mbps\n", ((count2* pack2 * 8) /(time2*1000000)))
}

```



Experiment- 17

Aim: Write a program for error detecting code using CR C-C CITT (16- bits)

Code:

```
package hello;
import java.io.*;
class BellmanFord{
    public static void main(String args[]) throws IOException
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int[ ] data;
        int[ ]div;
        int[ ]divisor;
        int[ ]rem;
        int[ ] crc;
        int data_bits, divisor_bits, tot_length;
        System.out.println("Enter number of data bits : ");
        data_bits=Integer.parseInt(br.readLine());
        data=new int[data_bits];
        System.out.println("Enter data bits : ");
        for(int i=0; i<data_bits; i++)
            data[i]=Integer.parseInt(br.readLine());
        System.out.println("Enter number of bits in divisor : ");
        divisor_bits=Integer.parseInt(br.readLine());
        divisor=new int[divisor_bits];
        System.out.println("Enter Divisor bits : ");
        for(int i=0; i<divisor_bits; i++)
            divisor[i]=Integer.parseInt(br.readLine());
        tot_length=data_bits+divisor_bits;
        div=new int[tot_length];
        rem=new int[tot_length];
        crc=new int[tot_length];
        /*----- CRC GENERATION -----*/
        for(int i=0;i<data.length;i++)
            div[i]=data[i];
        System.out.print("Dividend (after appending 0's) are : ");
        for(int i=0; i< div.length; i++)
            System.out.print(div[i]);
        System.out.println();
        for(int j=0; j<div.length; j++){
            rem[j] = div[j];
```



```

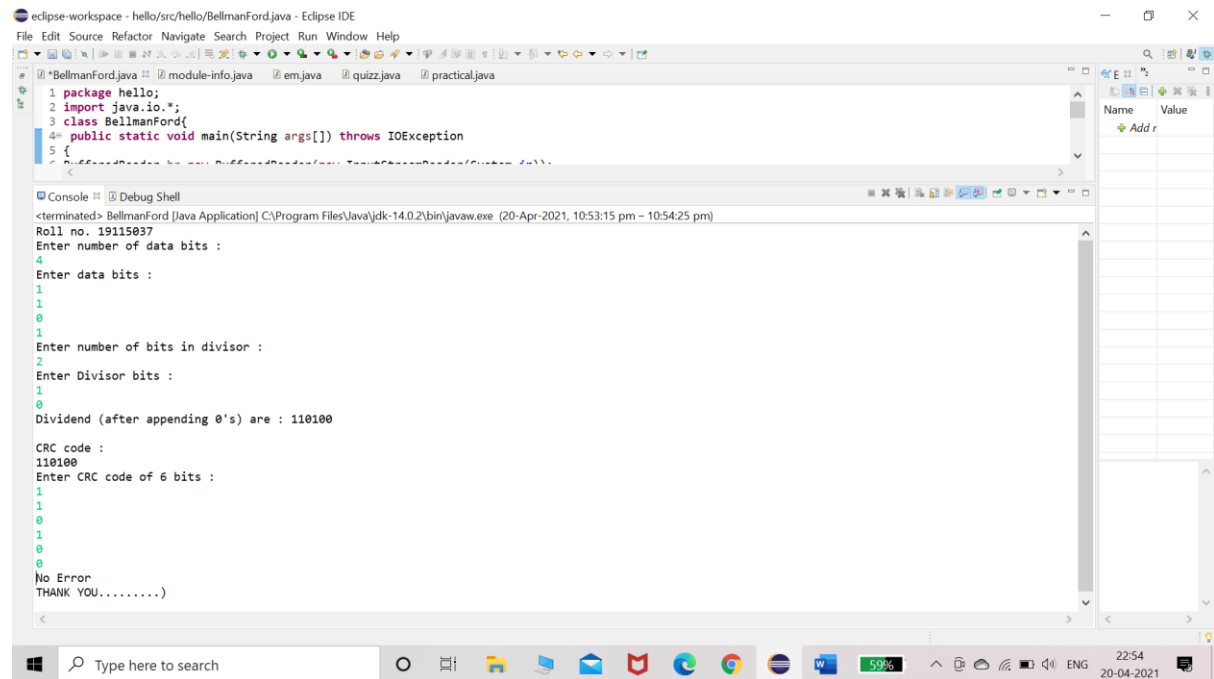
    }
    rem=divide(div, divisor, rem);
    for(int i=0;i<div.length;i++)
    //append dividend and remainder
    {
        crc[i]=(div[i]^rem[i]);
    }
    System.out.println();
    System.out.println("CRC code : ");
    for(int i=0;i<crc.length;i++)
        System.out.print(crc[i]);
    /*-----ERROR DETECTION -----*/
    System.out.println();
    System.out.println("Enter CRC code of "+ tot_length +" bits : ");
    for(int i=0; i<crc.length; i++)
        crc[i]=Integer.parseInt(br.readLine());
    for(int j=0; j<crc.length; j++){
        rem[j] = crc[j];
    }
    rem=divide(crc, divisor, rem);
    for(int i=0; i< rem.length; i++)
    {
        if(rem[i]!=0)
        {
            System.out.println("Error");
            break;
        }
        if(i==rem.length-1)
            System.out.println("No Error");
    }
    System.out.println("THANK YOU.....");
}

static int[] divide(int div[],int divisor[], int rem[])
{
    int cur=0;
    while(true)
    {
        for(int i=0;i<divisor.length;i++)
            rem[cur+i]=(rem[cur+i]^divisor[i]);
        while(rem[cur]==0 && cur != rem.length-1)
            cur++;
        if((rem.length-cur)<divisor.length)
            break;
    }
}

```

```
        return rem;
    }
}
```

Output:



The screenshot shows the Eclipse IDE interface. The editor displays a Java file named `BellmanFord.java` with the following code:

```
1 package hello;
2 import java.io.*;
3 class BellmanFord{
4     public static void main(String args[]) throws IOException
5     {
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7         // Roll no. 19115037
8         // Enter number of data bits :
9         int dataBits = Integer.parseInt(br.readLine());
10        // Enter data bits :
11        String data = "";
12        for (int i = 0; i < dataBits; i++)
13        {
14            char ch = br.readLine().charAt(0);
15            data += ch;
16        }
17        // Enter number of bits in divisor :
18        int divisorBits = Integer.parseInt(br.readLine());
19        // Enter Divisor bits :
20        String divisor = "";
21        for (int i = 0; i < divisorBits; i++)
22        {
23            char ch = br.readLine().charAt(0);
24            divisor += ch;
25        }
26        // Dividend (after appending 0's) are : 110100
27        String dividend = data + "000000";
28        // CRC code :
29        String crcCode = "";
30        for (int i = 0; i < divisorBits; i++)
31        {
32            char ch = dividend.charAt(i);
33            crcCode += ch;
34        }
35        // Enter CRC code of 6 bits :
36        String crcCode6 = "";
37        for (int i = 0; i < 6; i++)
38        {
39            char ch = crcCode.charAt(i);
40            crcCode6 += ch;
41        }
42        // No Error
43        System.out.println("THANK YOU.....");
44    }
45 }
```

The console output shows the program's execution:

```
<terminated> BellmanFord [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (20-Apr-2021, 10:53:15 pm - 10:54:25 pm)
Roll no. 19115037
Enter number of data bits :
4
Enter data bits :
1
1
0
1
Enter number of bits in divisor :
2
Enter Divisor bits :
1
0
Dividend (after appending 0's) are : 110100
CRC code :
110100
Enter CRC code of 6 bits :
1
1
0
1
0
0
No Error
THANK YOU.....)
```

Experiment- 18

Aim: Write a program to find the shortest path between vertices using bellman-ford algorithm.

Code:

```
package hello;
import java.util.Scanner;
public class BellmanFord
{
    private int D[];
    private int num_ver;
    public static final int MAX_VALUE = 999;
    public BellmanFord(int num_ver)
    {
        this.num_ver = num_ver;
        D = new int[num_ver + 1];
    }
    public void BellmanFordEvaluation(int source, int A[][])
    {
        for (int node = 1; node <= num_ver; node++)
        {
            D[node] = MAX_VALUE;
        }
        D[source] = 0;
        for (int node = 1; node <= num_ver - 1; node++)
        {
            for (int sn = 1; sn <= num_ver; sn++)
            {
                for (int dn = 1; dn <= num_ver; dn++)
                {
                    if (A[sn][dn] != MAX_VALUE)
                    {
                        if (D[dn] > D[sn] + A[sn][dn])
                        D[dn] = D[sn] + A[sn][dn];
                    }
                }
            }
        }
        for (int sn = 1; sn <= num_ver; sn++)
        {
            for (int dn = 1; dn <= num_ver; dn++)
            {
```

```

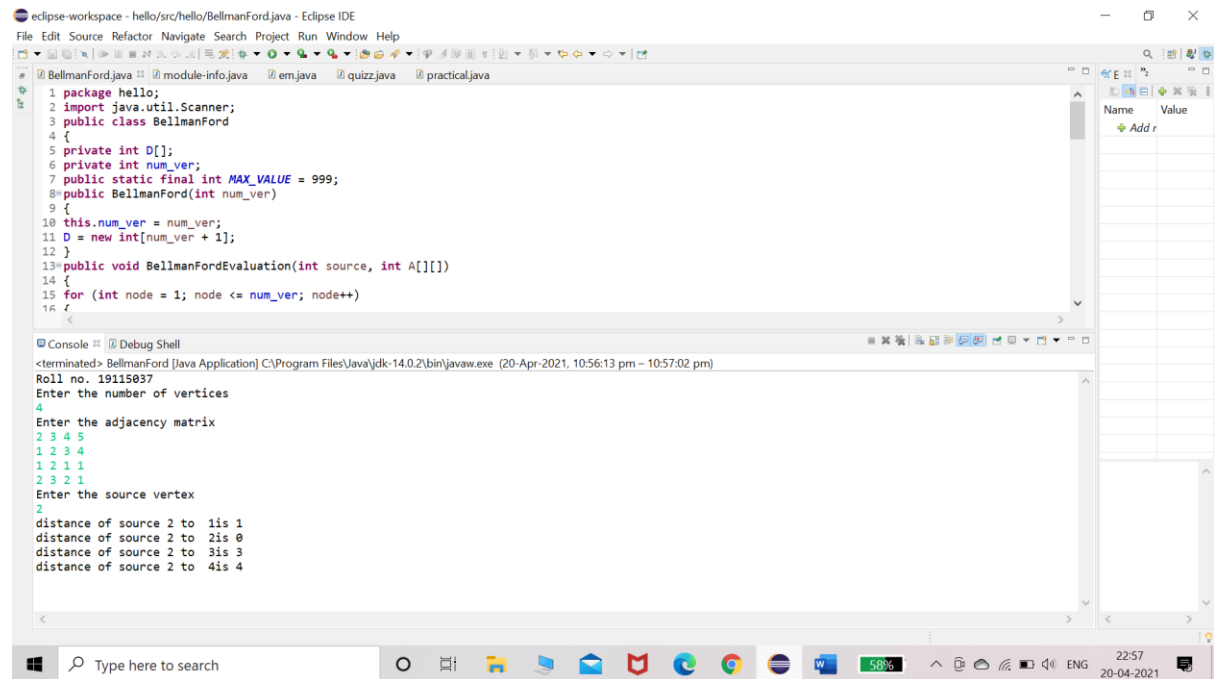
        if (A[sn][dn] != MAX_VALUE)
        {
            if (D[dn] > D[sn] + A[sn][dn])
                System.out.println("The Graph contains negative egde cycle");
        }
    }
}

for (int vertex = 1; vertex <= num_ver; vertex++)
{
    System.out.println("distance of source " + source + " to " + vertex + "is " + D[vertex]);
}

public static void main(String[ ] args)
{
    int num_ver= 0;int source;
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter the number of vertices");
    num_ver = scanner.nextInt();
    int A[][] = new int[num_ver + 1][num_ver + 1];
    System.out.println("Enter the adjacency matrix");
    for (int sn = 1; sn <= num_ver; sn++)
    {
        for (int dn = 1; dn <= num_ver; dn++)
        {
            A[sn][dn] = scanner.nextInt();
            if (sn == dn)
            {
                A[sn][dn] = 0;
                continue;
            }
            if (A[sn][dn] == 0)
            {
                A[sn][dn] = MAX_VALUE;
            }
        }
    }
    System.out.println("Enter the source vertex");
    source = scanner.nextInt();
    BellmanFord b = new BellmanFord (num_ver);
    b.BellmanFordEvaluation(source, A);
    scanner.close();
}
}

```

Output:



The screenshot displays the Eclipse IDE interface with the following components:

- Editor:** Shows the code for `BellmanFord.java`. The code implements a Bellman-Ford algorithm to find the shortest paths from a source vertex to all other vertices in a weighted undirected graph.
- Console:** Shows the output of the program execution. The output is as follows:

```
<terminated> BellmanFord [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (20-Apr-2021, 10:56:13 pm - 10:57:02 pm)
Roll no. 19115837
Enter the number of vertices
4
Enter the adjacency matrix
2 3 4 5
1 2 3 4
1 2 1 1
2 3 2 1
Enter the source vertex
2
distance of source 2 to 1 is 1
distance of source 2 to 2 is 0
distance of source 2 to 3 is 3
distance of source 2 to 4 is 4
```
- Task List:** Shows a table with columns "Name" and "Value".
- System Tray:** Shows the system clock as 22:57 on 20-04-2021.

Experiment- 19

Aim: Write a program for simple RSA algorithm to encrypt and decrypt the data.

Code:

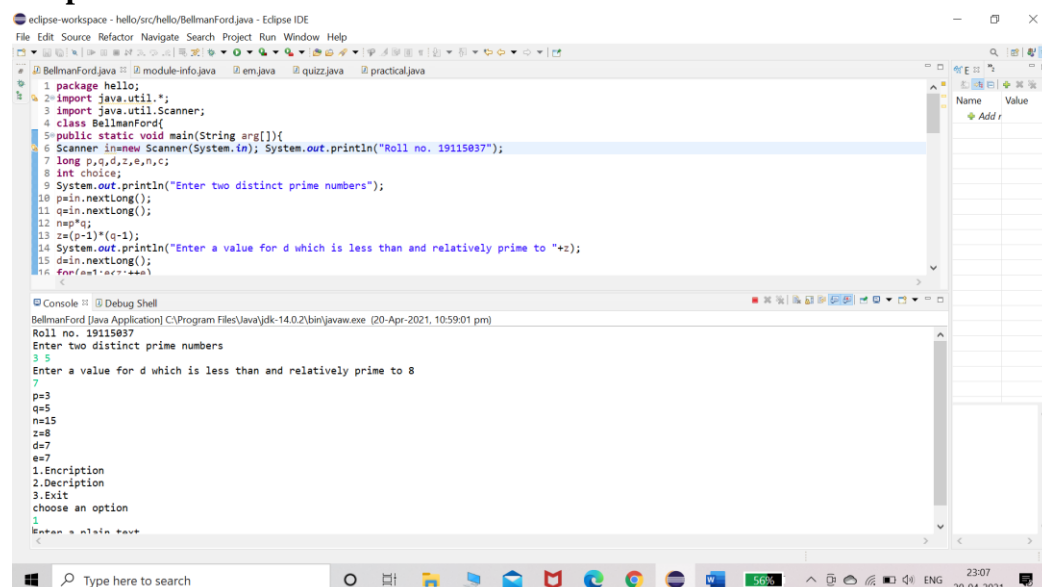
```
package hello;
import java.util.*;
import java.util.Scanner;
class h{
public static void main(String arg[]){
Scanner in=new Scanner(System.in);
long p,q,d,z,e,n,c;
int choice;
System.out.println("Enter two distinct prime numbers");
p=in.nextLong();
q=in.nextLong();
n=p*q;
z=(p-1)*(q-1);
System.out.println("Enter a value for d which is less than and relatively prime to "+z);
d=in.nextLong();
for(e=1;e<z;++e)
{
if(((e*d)%z)==1)
break;
}
System.out.println("p="+p+"\nq="+q+"\nn="+n+"\nz="+z+"\nd="+d+"\ne="+e);
do{
System.out.println("1.Encryption \n2.Decription\n3.Exit");
System.out.println("choose an option");
choice = in.nextInt();
switch(choice){
case 1:System.out.println("Enter a plain text");
String s = in.next();
System.out.println("Plain Text" + "\t" + "Cipher Text");
for(int i = 0; i < s.length(); i++){
long pl = (int) s.charAt(i);
c = modexp(pl,e,n);
System.out.println(s.charAt(i)+"\t\t" + c);
}
break;
case 2: System.out.println("Enter a cipher text (0 to stop input)");
long [] ci = new long[50];
```

```

    int j=0;
    do{
    ci[j] = in.nextLong();
    }while(ci[j++] != 0);
    System.out.println("Cipher Text" + "\t" + "Plain Text");
    for(int i = 0; i < j-1; i++){
    long pl = modexp(ci[i],d,n);
    System.out.println(ci[i]+" \t\t" + (char) pl);
    }
    break;
case 3: System.out.println("Program Terminated");
System.exit(0);
}
}while(choice != 3);
}
static long modexp(long a,long x,long n)
{long r=1;
while(x>0)
{
if(x%2==1){
r=(r*a)%n;
}
a=(a*a)%n;
x/=2;
}
return(r);
}
}

```

Output:



The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the source code of `BellmanFord.java`. The code includes package declarations, imports, and a `main` method that prompts the user for a roll number, two prime numbers, a value `d`, and a choice between encryption, decryption, or exit.
- Console:** Shows the output of the program. It displays the roll number `19115837`, the two prime numbers `p=3` and `q=5`, the calculated values `n=15`, `z=8`, `d=7`, and `ea=7`. It then lists the menu options: `1. Encryption`, `2. Decryption`, `3. Exit`, and prompts the user to choose an option.
- Variables View:** Located on the right, it shows a table with columns `Name` and `Value`. It contains one entry: `r` with a value of `1`.

Experiment- 20

Aim: Write a program for congestion control using leaky bucket algorithm.

Code:

```
package hello;
import java.io.*;
import java.util.*;
class Queue
{
int q[],f=0,r=0,size;
void insert(int n)
{
Scanner in = new Scanner(System.in);
q=new int[10];
for(int i=0;i<n;i++)
{
System.out.print("\nEnter " + i + " element: ");
int ele=in.nextInt();
if(r+1>10)
{
System.out.println("\nQueue is full \nLost Packet: "+ele);
break;
}
else
{
r++;
q[i]=ele;
}
}
}
void delete()
{
Scanner in = new Scanner(System.in);
Thread t=new Thread();
if(r==0)
System.out.print("\nQueue empty ");
else
{
for(int i=f;i<r;i++)
{
try
```



```

{
    t.sleep(1000);
}
catch(Exception e){}
System.out.print("\nLeaked Packet: "+q[i]); f++;
}
}
System.out.println();
}
}
class practical extends Thread
{
    public static void main(String ar[]) throws Exception {
        Queue q=new Queue();
        Scanner src =new Scanner(System.in);
        System.out.println("\nEnter the packets to be sent:");int size=src.nextInt();
        q.insert(size);
        q.delete();
    }
}

```

Output:

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the source code of the program, including the `practical.java` file. The code is as follows:


```

23     q[i]=ele;
24 }
25 }
26 }
27 void delete()
28 {
29     Scanner in = new Scanner(System.in);
30     Thread t=new Thread();
31     if(r==0)
32         System.out.print("\nQueue empty ");
33     else
34     {
35         for(int i=f;i<r;i++)
36         {
37             try
38             {
39                 t.sleep(1000);
40             }
41             catch(Exception e){}
42             System.out.print("\nLeaked Packet: "+q[i]); f++;
43         }
44         System.out.println();
45     }
46 }
47 }
48 class BellmanFord extends Thread
49 {
50     public static void main(String ar[]) throws Exception {
51         Queue q=new Queue(); System.out.println("Roll no. 19115037");
52         Scanner src =new Scanner(System.in);
53         System.out.println("\nEnter the packets to be sent:");int size=src.nextInt();
54         q.insert(size);
55         q.delete();
56     }
57 }
58

```
- Console:** Shows the output of the program:


```

<terminated> BellmanFord [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (20-Apr-2021, 11:14:30)
Roll no. 19115037
Enter the packets to be sent:
3
Enter 0 element: 3
Enter 1 element: 4
Enter 2 element: 5
Leaked Packet: 3
Leaked Packet: 4
Leaked Packet: 5

```
- Taskbar:** Shows the Windows taskbar with the search bar and various application icons.